First-Order μ-Calculus over Generic Transition Systems and Applications to the Situation Calculus

Diego Calvanese[a], Giuseppe De Giacomo[b], Marco Montali[a], Fabio Patrizi[a]

[a] *Free University of Bozen-Bolzano, Italy*
lastname@inf.unibz.it
[b] *Sapienza Università di Roma*
degiacomo@dis.uniroma1.it

Abstract

In this paper, we consider $\mu\mathcal{L}$, $\mu\mathcal{L}_a$, and $\mu\mathcal{L}_p$, three variants of the first-order μ-calculus introduced in the literature for verification of data-aware processes. In the first one, quantification on objects ranges over the whole domain; in the second one, quantification ranges over objects in the active domain of the current state only; in the latter one, objects assigned to variables are further required to persist across transitions. Each of these three logics has a distinct corresponding notion of bisimulation. In spite of the differences among these logics, we show that the three notions of bisimulation collapse for dynamic systems that are *generic*, which include all those systems specified in a state-based fashion using a logical formalism, such as, e.g., the situation calculus. As a result, for such dynamic systems, $\mu\mathcal{L}$, $\mu\mathcal{L}_a$, and $\mu\mathcal{L}_p$ have the same expressive power. In addition, we show that, when the dynamic system stores only a *bounded* number of objects in each state, a finite abstraction can be constructed which is faithful for $\mu\mathcal{L}$ (the most general variant). This yields decidability of verification. In particular, by applying such results, we show decidability of $\mu\mathcal{L}$ verification over *bounded situation calculus action theories*. These decidability results contrast with the undecidability of verification for first-order LTL over state-bounded generic transition systems, and notably imply that first-order LTL cannot be captured by $\mu\mathcal{L}$.

Keywords: Reasoning about actions, verification, situation calculus, first-order μ-calculus, infinite transition systems, state-bounded transition systems.

1. Introduction

In this paper, we study first-order μ-calculus as a verification language for transition systems whose states are first-order (relational) models over a fixed infinite object domain. In particular, we consider *generic transition systems*, i.e., roughly speaking, transition systems whose successor states depend only on the logical properties of the current one [1, 28]. Essentially, all dynamic systems in literature specified through some sort of logical formalism give rise to

Preprint submitted to Information & Computation *January 20, 2017*

generic transition systems, including data-aware processes studied in Databases [10, 26, 3], and action theories studied in Artificial Intelligence, e.g., expressed in the situation calculus [31, 36].

Recently, many important results have been devised regarding sound, complete, and terminating verification for dynamic systems with a first-order relational state description [17, 8, 20, 3, 5, 40, 9, 21, 22]. These results are concerned with verification logics that are variants of those studied in the area of model checking of finite-state transition systems, like LTL, CTL, or modal μ-calculus, which subsumes the previous one in the propositional setting [16, 6]. Obviously, to be used in the context of formalisms with a first-order state description, such logics need to be extended with the ability of querying the state in first-order logic. However, in most proposals, e.g., [20, 23], such ability is limited to the use of first-order sentences (closed formulas), without the possibility of quantifying over object across different states. *Quantification across (states)* refers to the possibility of using variables quantified in the current state also in future states. Without quantification across, these first-order temporal logics remain quite similar to their propositional counterpart (though with infinitely many propositions corresponding to first-order sentences, instead of the usual finite ones). In particular, notions like bisimulation and bisimulation invariance essentially correspond to those known for the propositional case. Only very few papers have studied verification logics with quantification across [8, 3, 9, 21].

In this paper, we study in depth first-order μ-calculus with quantification across. In particular, we consider three basic μ-calculus variants proposed in literature, namely $\mu\mathcal{L}$, $\mu\mathcal{L}_a$ and $\mu\mathcal{L}_p$, which are characterized by different restrictions on how quantification across is controlled. The logic $\mu\mathcal{L}$ poses no restriction on object quantification. This logic has been used in several contexts [24, 34, 3] and is the most natural extension of modal μ-calculus to express properties of transition systems with first-order states. The logic $\mu\mathcal{L}_a$ is a restriction of $\mu\mathcal{L}$ where quantification is required to range over objects in the *active domain* of the current state only, i.e., objects present in the extension of some predicate in the current state. In the context of artifact-centric dynamic systems [18], the logic $\mu\mathcal{L}_a$ was studied in [3], while its CTL fragment has been investigated in [8, 9]. The logic $\mu\mathcal{L}_p$ is a restriction of $\mu\mathcal{L}_a$ which further requires that the objects assigned to the quantified variables persist across the states traversed while checking the formula. The logic $\mu\mathcal{L}_p$ was also studied in [3], and then in the context of situation calculus action theories [20, 21].

As shown in [3], these three logics can be characterized by three distinct notions of bisimulation over transition systems: standard *bisimulation* (though extended to deal with first-order states); history preserving bisimulation (or *a-bisimulation*) for $\mu\mathcal{L}_a$; and persistence preserving bisimulation (or *p-bisimulation*) for $\mu\mathcal{L}_p$. Specifically, $\mu\mathcal{L}$ is invariant with respect to bisimulation, $\mu\mathcal{L}_a$ is invariant with respect to a-bisimulation, and $\mu\mathcal{L}_p$ is invariant with respect to p-bisimulation, where *bisimulation invariance* means that two bisimilar states satisfy the exactly same formulas.

Decidability results for verification have also been devised. A crucial notion to obtain decidability is that of *state-boundedness* [8, 20, 3, 4]. In particular, [20] shows that verification of first-order μ-calculus without quantification across over *bounded action theories* in the situation calculus is decidable. Such theories have an infinite object domain, but the number of object tuples that belong to fluents in each situation remains bounded. Nonetheless, an agent may deal with an infinite number of objects over the course of an infinite execution. In [21], these results are extended to deal with quantification across, showing that models of bounded situation calculus action theories can be faithfully abstracted into p-bisimilar finite-state transition systems. This yields decidability of verification for $\mu\mathcal{L}_p$. Remarkably, such an abstraction is independent from the formula to verify.

Instead, [21] and [3] show, respectively for the situation calculus and for artifact-centric dynamic systems, that in the $\mu\mathcal{L}_a$ case (and hence also in $\mu\mathcal{L}$) no faithful finite abstraction can exist that is independent from the formula to check. Interestingly, [8, 9] prove that, for *state-bounded* transition systems, a faithful abstraction depending on the number of variables in the formula exists for the CTL fragment of $\mu\mathcal{L}_a$. Only recently it has been shown that this decidability result extends to $\mu\mathcal{L}_a$ [13], while it remained open, until now, whether it extends to $\mu\mathcal{L}$ as well.

Here we investigate thoroughly $\mu\mathcal{L}$, $\mu\mathcal{L}_a$ and $\mu\mathcal{L}_p$ and the bisimulation notions associated to them. We establish quite surprising results with respect to the expressive power of the three logics, and we establish decidability of verification for $\mu\mathcal{L}$ against state-bounded transition systems in general, and in particular against bounded situation calculus action theories.

Specifically, we present the following results:

- For generic transition systems, such as those generated by situation calculus theories, the notions of p-bisimilarity, a-bisimilarity and bisimilarity collapse.
- For generic transition systems with the additional condition that the active domain of each state is finite, though not necessarily bounded, $\mu\mathcal{L}$, $\mu\mathcal{L}_a$ and $\mu\mathcal{L}_p$ have exactly the same expressive power, in the sense that if a $\mu\mathcal{L}$ formula distinguishes two states, then there exists a $\mu\mathcal{L}_p$ formula, and thus a $\mu\mathcal{L}_a$ formula, that does so as well (and obviously vice-versa).
- As a consequence of the equivalence between p-bisimilarity and bisimilarity, we have that if two generic transition systems (with infinite object domains) are p-bisimilar, they satisfy the same $\mu\mathcal{L}$ formulas. We strengthen this result by showing that, if one of the transition systems has a finite object domain that is *large enough*, then it preserves all $\mu\mathcal{L}$ formulas that use at most a predefined number of variables.
- We further show that, for state-bounded generic transition systems, and for a given set of variables, it is always possible to define a faithful finite-state abstraction that preserves all $\mu\mathcal{L}$ formulas with variables belonging to that set. This in particular applies to models of bounded situation calculus action theories.
- Finally, we show that given a bounded situation calculus action theory

(including those with incomplete information), and a set of variables, we can effectively construct a new situation calculus action theory with finite domain that preserves $\mu\mathcal{L}$ formulas whose variables belong to that set. In this way, we obtain decidability of verification of $\mu\mathcal{L}$ formulas over bounded situation calculus action theories.

These results have a strong impact also for the following reason. In [3] it is shown that verification of first-order LTL with quantification across ranging over the active domain is undecidable even for state-bounded generic transition systems. Then, using the folk assumption that μ-calculus captures LTL also in the first-order case, e.g., [34], it is concluded that $\mu\mathcal{L}_a$ verification is undecidable for state-bounded transition systems. Here, we show that this is not true, and that $\mu\mathcal{L}$ verification is indeed decidable over state-bounded transition systems while first-order LTL verification is not. This has the notable consequence that *first-order μ-calculus cannot capture first-order LTL, in general.* In other words, once we allow for quantification across, the ability of LTL of talking about single traces cannot be mimicked anymore by μ-calculus. To the best of our knowledge this is the first formal proof of this important fact.

2. Generic Transition Systems

We consider *transition systems* with first-order (relational)[1] states, i.e., such that each state is associated with a full first-order interpretation over a fixed alphabet of predicates, including equality interpreted as identity, and a fixed object domain. Let $\mathcal{F}$ be a finite set of predicates, also called *fluents*, C a finite set of constants, and Δ an infinite object domain. We denote by $Int_{\Delta}^{\mathcal{F},C}$ the set of all possible interpretations of predicates in $\mathcal{F}$ and constants in C over the object domain Δ. A *transition system* (TS) (over predicates $\mathcal{F}$, constants C, and object domain Δ) is a tuple $T = \langle \Delta, Q, q_0, \rightarrow, \mathcal{I} \rangle$, where:

- Q is the *set of states*;
- $q_0 \in Q$ is the *initial state*;
- $\rightarrow \subseteq Q \times Q$ is the *transition relation*; and
- $\mathcal{I} : Q \mapsto Int_{\Delta}^{\mathcal{F},C}$ is the *labeling function* associating to each state q an interpretation $\mathcal{I}(q) = \langle \Delta, \cdot^{\mathcal{I}(q)} \rangle$ such that the constants in C are interpreted in the same way in all the states over which $\mathcal{I}$ is defined.

We denote by $adom(\mathcal{I}(q))$ the *active domain* of $\mathcal{I}(q)$, i.e., the set of objects occurring in the extension of some predicate in $\mathcal{I}(q)$ or interpreting some constant in C. Also we denote by $\tilde{\mathcal{I}}(q)$ the restriction of $\mathcal{I}(q)$ to its active domain, i.e., $\tilde{\mathcal{I}}(q) = \langle adom(\mathcal{I}(q)), \cdot^{\mathcal{I}(q)} \rangle$.

[1] In this paper, we focus on the relational view of first-order logic without considering function symbols.

Among the various TSs we are interested in those that are *generic*. Genericity is a standard notion in Databases [1] formalizing the fact that answers to queries depend only on the mutual relationships of the objects in the database (which can be seen as a first-order interpretation). Such notion has been adapted to capture when the states of a dynamic system are generated through a first-order specification, such as in [9] (there called *uniformity*), or in [3], or in [20, 21].

To introduce genericity formally, we first recall the standard notions of *isomorphism* and *isomorphic interpretations* [28]. Two first-order interpretations $\mathcal{I}_1 = \langle \Delta_1, \cdot^{\mathcal{I}_1} \rangle$ and $\mathcal{I}_2 = \langle \Delta_2, \cdot^{\mathcal{I}_2} \rangle$, over the same predicates $\mathcal{F}$ and constants C, are said to be *isomorphic*, written $\mathcal{I}_1 \sim \mathcal{I}_2$, if there exists a bijection (called *isomorphism*) $h : \Delta_1 \mapsto \Delta_2$ such that: *(i)* for every $F \in \mathcal{F}$, and every $\vec{x} \in F^{\mathcal{I}_1}$ if and only if $h(\vec{x}) \in F^{\mathcal{I}_2}$; *(ii)* for every $c \in C$, $c^{\mathcal{I}_2} = h(c^{\mathcal{I}_1})$. Intuitively, for two interpretations to be isomorphic, it is required that one can be obtained from the other by renaming the individuals in the interpretation domain. Notice that, necessarily, the interpretation domains of isomorphic interpretations have the same cardinality. When needed, to make it explicit that h is an isomorphism between $\mathcal{I}_1$ and $\mathcal{I}_2$, we write $\mathcal{I}_1 \sim_h \mathcal{I}_2$.

Definition 1 (Generic Transition System)**.** A TS $T = \langle \Delta, Q, q_0, \rightarrow, \mathcal{I} \rangle$ is said to be *generic* if: for every $q_1, q_1', q_2 \in Q$ and every bijection $h : \Delta \mapsto \Delta$, if $\mathcal{I}(q_1) \sim_h \mathcal{I}(q_2)$ and $q_1 \rightarrow q_1'$, then there exists $q_2' \in Q$ such that $q_2 \rightarrow q_2'$ and $\mathcal{I}(q_1') \sim_h \mathcal{I}(q_2')$.

Intuitively, genericity requires that if two states are isomorphic they induce the "same" transitions (modulo isomorphism). This property is always true if the next states are built by a logical specification involving only the current state and the next one, as long as we do not use predefined domains and relations (such as order) with special properties that are specified *extra-logically* (e.g., we allow for natural numbers without formalizing them in the logic itself). In particular it holds for situation calculus specifications (and indeed virtually all first-order based formalism for reasoning about actions used in AI) [36].

Next we introduce *state-bounded* TSs. These are TSs whose states can contain only boundedly many objects in the active domain. This restriction, together with genericity, is at the base of a series of decidability results for verification of various temporal logics against TSs with first-order states [9, 3, 20, 21].

Definition 2 (State-Bounded Transition System)**.** A TS $T = \langle \Delta, Q, q_0, \rightarrow, \mathcal{I} \rangle$ is said to be *(state-)bounded by b* if we have that $|adom(\mathcal{I}(q))| \leq b$ for every $q \in Q$. Moreover, T is *state bounded* if it is state-bounded by b for some b.

That is, we say that T is state-bounded if there is a bound on the number of objects that can be accumulated in the same state. Notice that this does not disallow the possibility of accumulating infinitely many objects along an infinite run (or the entire TS for the matter).

3. First-Order Variants of μ-Calculus

To specify temporal properties, we adopt modal μ-calculus [27, 37, 11], one of the most powerful temporal logics for which model checking has been investigated. It is well-known that in the propositional setting μ-calculus is able to capture both linear time logics, such as LTL, and branching time logics such as CTL and CTL* [16, 6]. The main feature of modal μ-calculus is the ability of expressing directly least and greatest fixpoints of (predicate-transformer) operators formed using formulae relating the current state to the next one. By using such fixpoint constructs one can easily express sophisticated temporal properties defined by induction or co-induction. In the following we consider three first-order variants of modal μ-calculus that have been considered in literature.

3.1. The Logic $\mu\mathcal{L}$

The first logic we consider is characterized by unrestricted quantification over objects, and was studied, e.g., in [24, 3]. The syntax of $\mu\mathcal{L}$ is

$$\Phi \ ::= \ \varphi \ \mid \ \neg\Phi \ \mid \ \Phi_1 \wedge \Phi_2 \ \mid \ \exists x.\Phi \ \mid \ \langle - \rangle\Phi \ \mid \ Z \ \mid \ \mu Z.\Phi,$$

where: φ is a first-order formula expressed using predicates in $\mathcal{F}$ and constants in C; the modal operator $\langle - \rangle\Phi$ denotes the existence of a transition from the current state to a next state where Φ holds; and $\mu Z.\Phi$ denotes the *least fixpoint* of the formula Φ seen as a predicate transformer with respect to Z. We use the standard abbreviations for $\supset$ and $\forall$. We also use $\nu Z.\Phi$ as an abbreviation for $\neg\mu Z.\neg\Phi[Z/\neg Z]^2$, to denote the *greatest fixpoint* of Φ. Note that in $\mu\mathcal{L}$ quantification across ranges over arbitrary objects in the object domain. As usual in μ-calculus, formulas of the form $\mu Z.\Phi$ (and $\nu Z.\Phi$) must obey to the *syntactic monotonicity* of Φ with respect to Z, which states that every occurrence of the variable Z in Φ must be within the scope of an even number of negation symbols. This ensures that the semantics of $\mu Z.\Phi$ and $\nu Z.\Phi$ is well defined.

Example 1. The $\mu\mathcal{L}$ formula

$$\forall x.\mathit{Stud}(x) \supset \mu Y.((\exists y.\mathit{Grad}(x, y)) \vee \langle - \rangle Y)$$

states that for each student x in the current state, there exists an evolution that eventually leads to the graduation of x (with some final mark y). ■

To interpret $\mu\mathcal{L}$ formulas over a TS $T = \langle \Delta, Q, q_0, \rightarrow, \mathcal{I} \rangle$, we use valuations (v, V) formed by an individual variable valuation v and a predicate variable valuation V parametrized by v, i.e., which maps each predicate variable Z to a subset $V(v, Z)$ of Q. We define the *extension function* $(\cdot)^T_{(v,V)}$, which maps $\mu\mathcal{L}$ formulas to subsets of Q, as shown in Figure 1.

Given a $\mu\mathcal{L}$ formula Φ, we say that a TS T *satisfies* Φ *at state* q *under* v *and* V, written $T, q, (v, V) \models \Phi$, if $q \in (\Phi)^T_{(v,V)}$. When Φ is closed on predicate

[2] $\Phi[Z/\neg Z]$ denotes the result of syntactically substituting Z with $\neg Z$ in Φ.

$$\begin{array}{rl}
(\varphi)^T_{(v,V)} & = \{q \mid q \in Q \text{ and } \mathcal{I}(q), v \models \varphi\} \\
(\neg\Phi)^T_{(v,V)} & = Q \setminus (\Phi)^T_{(v,V)} \\
(\Phi_1 \wedge \Phi_2)^T_{(v,V)} & = (\Phi_1)^T_{(v,V)} \cap (\Phi_2)^T_{(v,V)} \\
(\exists x.\Phi)^T_{(v,V)} & = \{q \mid \exists d \in \Delta.q \in (\Phi)^T_{(v,V)[x/d]}\} \\
(\langle - \rangle\Phi)^T_{(v,V)} & = \{q \mid \exists q'.q \rightarrow q' \text{ and } q' \in (\Phi)^T_{(v,V)}\} \\
(Z)^T_{(v,V)} & = V(Z) \\
(\mu Z.\Phi)^T_{(v,V)} & = \bigcap\{\mathcal{E} \subseteq Q \mid (\Phi)^T_{(v,V)[Z/\mathcal{E}]} \subseteq \mathcal{E}\}
\end{array}$$

$(v, V)[x/d]$ stands for (v', V) where v' is as v except that $v'(x) = d$. Similarly $(v, V)[Z/\mathcal{E}]$ stands for (v, V') where V' is as V except that $V'(v, Z) = \mathcal{E}$.

Figure 1: Semantics of $\mu\mathcal{L}$

variables, we omit V, as irrelevant, and write $T, q, v \models \Phi$. If Φ is closed on both individual and predicate variables we simply write $T, q \models \Phi$. For closed formulas, we say that T *satisfies* Φ, written $T \models \Phi$, if $T, q_0 \models \Phi$.

We can naturally extend the classical notion of *bisimulation* [32] to deal with TSs with first-order states. Let $T_1 = \langle \Delta_1, Q_1, q_{10}, \rightarrow_1, \mathcal{I}_1\rangle$ and $T_2 = \langle \Delta_2, Q_2, q_{20}, \rightarrow_2, \mathcal{I}_2\rangle$ be two TSs over predicates in $\mathcal{F}$ and constants in C. A relation $R \subseteq Q_1 \times Q_2$ is a *bisimulation* between T_1 and T_2, if there exists a bijection $b : \Delta_1 \mapsto \Delta_2$ such that $\langle q_1, q_2\rangle \in R$ implies that:

1. $\mathcal{I}_1(q_1) \sim_b \mathcal{I}_2(q_2)$;
2. for each $q'_1 \in Q_1$, if $q_1 \rightarrow_1 q'_1$ then there exists $q'_2 \in Q_2$ such that $q_2 \rightarrow_2 q'_2$ and $\langle q'_1, q'_2\rangle \in R$;
3. for each $q'_2 \in Q_2$, if $q_2 \rightarrow_2 q'_2$ then there exists $q'_1 \in Q_1$ such that $q_1 \rightarrow_1 q'_1$ and $\langle q'_1, q'_2\rangle \in R$;

We say that a state $q_1 \in Q_1$ is *bisimilar* to $q_2 \in Q_2$, written $q_1 \approx q_2$, if there exists a bisimulation R between T_1 and T_2 such that $\langle q_1, b, q_2\rangle \in R$. When needed, we also write $q_1 \approx_b q_2$, to explicitly name b. Finally, T_1 is said to be *bisimilar* to T_2, written $T_1 \approx T_2$, if $q_{10} \approx q_{20}$. It is immediate to see that bisimilarity between states and TSs, i.e., the (overloaded) relation $\approx$, is an equivalence relation.

Using the notion of bisimilarity, one can prove a suitable version of the classical *bisimulation invariance result* for the μ-calculus [11], which states that bisimilar TSs satisfy exactly the same μ-calculus formulas.

Theorem 3. *Consider two TSs $T_1 = \langle \Delta_1, Q_1, q_{10}, \rightarrow_1, \mathcal{I}_1\rangle$ and $T_2 = \langle \Delta_2, Q_2, q_{20}, \rightarrow_2, \mathcal{I}_2\rangle$ with Δ_1 and Δ_2 infinite. If $T_1 \approx T_2$, then for every $\mu\mathcal{L}$ closed formula Φ, $T_1 \models \Phi$ if and only if $T_2 \models \Phi$.*

Proof. Straightforward extension of the proof for modal μ-calculus in the propositional setting [11]. □

The converse of this theorem does not hold in general, but we show later that it holds under some general conditions.

3.2. The Logic $\mu\mathcal{L}_a$

The logic $\mu\mathcal{L}_a$ is characterized by the assumption that quantification over objects is restricted to those objects that are present in the current active domain, and was studied in [3] and in [9][3]. The syntax of $\mu\mathcal{L}_a$ is

$$\Phi \ ::= \ \varphi \ \mid \ \neg\Phi \ \mid \ \Phi_1 \wedge \Phi_2 \ \mid \ \exists x.\textsc{live}(x) \wedge \Phi \ \mid \ \langle - \rangle\Phi \ \mid \ Z \ \mid \ \mu Z.\Phi,$$

Note that in $\mu\mathcal{L}_a$ quantification across is forced to range over objects in the current active domain, through the special predicate $\textsc{live}(\cdot)$, which denotes membership to the active domain, and can be seen as an abbreviation for the disjunction $\bigvee P(\ldots, x, \ldots)$ over all predicates P and all positions of x in P. That is, individuals over which quantification ranges must belong to the active domain of the current state of the TS.

Example 2. The $\mu\mathcal{L}$ formula

$$\forall x.\mathit{Stud}(x) \supset \mu Y.((\exists y.\mathit{Grad}(x,y)) \vee \langle - \rangle Y)$$

is in fact a $\mu\mathcal{L}_a$ formula since $\forall x.\mathit{Stud}(x) \supset \mu Y.((\exists y.\mathit{Grad}(x,y)) \vee \langle - \rangle Y)$ is equivalent to $\forall x.\textsc{live}(x) \wedge \mathit{Stud}(x) \supset \mu Y.((\exists y.\textsc{live}(y) \wedge \mathit{Grad}(x,y)) \vee \langle - \rangle Y)$. ■

Next, we introduce the notion of *history-preserving bisimulation*, which captures $\mu\mathcal{L}_a$. Given a bijection $h : Q \mapsto Q'$, we denote with $\textsc{dom}(h)$ the domain of h, i.e., the set of elements in Q for which h is defined, and with $\textsc{img}(h)$ the image of h, i.e., the set of elements q' in Q' such that $q' = h(q)$ for some $q \in Q$. A bijection h' *extends* h if $\textsc{dom}(h) \subseteq \textsc{dom}(h')$ and $h'(x) = h(x)$ for all $x \in \textsc{dom}(h)$ (or equivalently $\textsc{img}(h) \subseteq \textsc{img}(h')$ and $h'^{-1}(y) = h^{-1}(y)$ for all $y \in \textsc{img}(h)$).

A history-preserving bisimulation relation can be defined as follows. Let $T_1 = \langle \Delta_1, Q_1, q_{10}, \rightarrow_1, \mathcal{I}_1 \rangle$ and $T_2 = \langle \Delta_2, Q_2, q_{20}, \rightarrow_2, \mathcal{I}_2 \rangle$ be two TSs (over the predicates in $\mathcal{F}$ and constants in C), and let H be the set of all possible bijections $h : D_1 \mapsto D_2$, for $D_1 \subseteq \Delta_1$ and $D_2 \subseteq \Delta_2$. A relation $R \subseteq Q_1 \times H \times Q_2$ is a *history-preserving bisimulation* (or *a-bisimulation*) between T_1 and T_2, if $\langle q_1, h, q_2 \rangle \in R$ implies that:

1. $\tilde{\mathcal{I}}_1(q_1) \sim_h \tilde{\mathcal{I}}_2(q_2)$, i.e., h is an isomorphism between the restrictions of $\mathcal{I}_1(q_1)$ and $\mathcal{I}_2(q_2)$ to their active domains;

2. for each $q_1' \in Q_1$, if $q_1 \rightarrow_1 q_1'$ then there exists $q_2' \in Q_2$ such that:

   (a) $q_2 \rightarrow_2 q_2'$, and

   (b) there exists a bijection $h' : \textsc{dom}(h) \cup \mathit{adom}(\mathcal{I}_1(q_1')) \mapsto \textsc{img}(h) \cup \mathit{adom}(\mathcal{I}_2(q_2'))$ that is an extension of h and such that $\langle q_1', h', q_2' \rangle \in R$;

[3] Actually, [9] considers the CTL fragment of $\mu\mathcal{L}_a$.

3. for each $q_2' \in Q_2$, if $q_2 \rightarrow_2 q_2'$ then there exists $q_1' \in Q_1$ such that:
    (a) $q_1 \rightarrow_1 q_1'$, and
    (b) there exists a bijection $h' : \textsc{dom}(h) \cup \mathit{adom}(\mathcal{I}_1(q_1')) \mapsto \textsc{img}(h) \cup \mathit{adom}(\mathcal{I}_2(q_2'))$ that is an extension of h and such that $\langle q_1', h', q_2' \rangle \in R$.

In other words, we say that two states (possibly of two different TSs) are history-preserving bisimilar if there is an isomorphism between them that can be *extended* in successor states, while preserving bisimulation. This means that, starting from the initial states of the two TSs, the identity of the objects seen along each history is preserved when moving to successor states.

We say that a state $q_1 \in Q_1$ is *history-preserving bisimilar* (or *a-bisimilar*) to $q_2 \in Q_2$, written $q_1 \approx^a q_2$, if there exists an a-bisimulation R between T_1 and T_2 such that $\langle q_1, h, q_2 \rangle \in R$, for some h; when needed, we also write $q_1 \approx^a_h q_2$, to explicitly name h. Finally, T_1 is said to be *a-bisimilar* to T_2, written $T_1 \approx^a T_2$, if $q_{10} \approx^a q_{20}$. It is immediate to see that bisimilarity between states and TSs, i.e., the (overloaded) relation $\approx^a$, is an equivalence relation.

Using the notion of a-bisimilarity, one can prove a suitable version of the classical *bisimulation invariance result*, see e.g., [11].

Theorem 4 ([3]). *Consider two TSs $T_1 = \langle \Delta_1, Q_1, q_{10}, \rightarrow_1, \mathcal{I}_1 \rangle$ and $T_2 = \langle \Delta_2, Q_2, q_{20}, \rightarrow_2, \mathcal{I}_2 \rangle$ with Δ_1 and Δ_2 infinite. If $T_1 \approx^a T_2$, then for every $\mu\mathcal{L}_a$ closed formula Φ, $T_1 \models \Phi$ if and only if $T_2 \models \Phi$.*

The converse of this theorem does not hold in general, but, as before, we show later that it holds under some general conditions.

3.3. The Logic $\mu\mathcal{L}_p$

Next, we consider a restriction of $\mu\mathcal{L}_a$, called $\mu\mathcal{L}_p$, studied in [3, 21]. The syntax of $\mu\mathcal{L}_p$ is

$$\begin{array}{rl}\Phi ::= & \varphi \mid \neg\Phi \mid \Phi_1 \wedge \Phi_2 \mid \exists x.\textsc{live}(x) \wedge \Phi \mid \\ & \textsc{live}(\vec{x}) \wedge \langle - \rangle \Phi \mid \textsc{live}(\vec{x}) \wedge [-]\Phi \mid Z \mid \mu Z.\Phi\end{array}$$

Note that in $\mu\mathcal{L}_p$ quantification across ranges over objects in the current active domain that *persist in the extension of some fluents across situations.* This is obtained by forcing through $\textsc{live}(\vec{x}) \wedge \langle - \rangle \Phi$ and $\textsc{live}(\vec{x}) \wedge [-]\Phi$ that the variables occurring free in Φ[4] are assigned to objects that are in the active domain of the current state.

Example 3. The following $\mu\mathcal{L}_p$ formula:

$$\forall x.\mathit{Stud}(x) \supset \mu Y.((\exists y.\mathit{Grad}(x,y)) \vee \textsc{live}(x) \wedge \langle - \rangle Y)$$

states that for each student x in the current state, there exists an evolution, where x *remains in the active domain*, which eventually leads to the graduation of x (with some final mark y). ■

[4] With the proviso that second order variables are substituted by their corresponding fixpoint formula.

The bisimulation relation that captures $\mu\mathcal{L}_p$ is defined as follows. Let $T_1 = \langle \Delta_1, Q_1, q_{10}, \rightarrow_1, \mathcal{I}_1 \rangle$ and $T_2 = \langle \Delta_2, Q_2, q_{20}, \rightarrow_2, \mathcal{I}_2 \rangle$ be two TSs (over the predicates in $\mathcal{F}$ and constants in C) and let H be the set of all possible bijections $h : D_1 \mapsto D_2$, for $D_1 \subseteq \Delta_1$ and $D_2 \subseteq \Delta_2$. A relation $R \subseteq Q_1 \times H \times Q_2$ is a *persistence-preserving bisimulation* (or *p-bisimulation*) between T_1 and T_2, if $\langle q_1, h, q_2 \rangle \in R$ implies that:

1. $\tilde{\mathcal{I}}_1(q_1) \sim_h \tilde{\mathcal{I}}_2(q_2)$;
2. for each $q'_1 \in Q_1$, if $q_1 \rightarrow_1 q'_1$ then there exists $q'_2 \in Q_2$ such that:
   (a) $q_2 \rightarrow_2 q'_2$, and
   (b) there exists a bijection $h' : \mathit{adom}(\mathcal{I}_1(q_1)) \cup \mathit{adom}(\mathcal{I}_1(q'_1)) \mapsto \mathit{adom}(\mathcal{I}_2(q_2)) \cup \mathit{adom}(\mathcal{I}_2(q'_2))$ such that its restriction $h'|_{\mathit{adom}(\mathcal{I}_1(q_1))}$ coincides with $h|_{\mathit{adom}(\mathcal{I}_1(q_1))}$ and $\langle q'_1, h'|_{\mathit{adom}(\mathcal{I}_1(q'_1))}, q'_2 \rangle \in R$;
3. for each $q'_2 \in Q_2$, if $q_2 \rightarrow_2 q'_2$ then there exists $q'_1 \in Q_1$ such that:
   (a) $q_1 \rightarrow_1 q'_1$, and
   (b) there exists a bijection $h' : \mathit{adom}(\mathcal{I}_1(q_1)) \cup \mathit{adom}(\mathcal{I}_1(q'_1)) \mapsto \mathit{adom}(\mathcal{I}_2(q_2)) \cup \mathit{adom}(\mathcal{I}_2(q'_2))$ such that its restriction $h'|_{\mathit{adom}(\mathcal{I}_1(q_1))}$ coincides with $h|_{\mathit{adom}(\mathcal{I}_1(q_1))}$ and $\langle q'_1, h'|_{\mathit{adom}(\mathcal{I}_1(q'_1))}, q'_2 \rangle \in R$.

In other words, we say that two states (possibly of two different TSs) are persistence-preserving bisimilar if there is an isomorphism between them that can be *maintained* in the successor state for all objects that are in the intersection of the active domains of the current and the successor state itself. This means that the identity of objects is preserved only as long as they persist in the active domain.

We say that a state $q_1 \in Q_1$ is *persistence-preserving bisimilar* (or *p-bisimilar*) to $q_2 \in Q_2$, written $q_1 \approx^p q_2$, if there exists a p-bisimulation R between T_1 and T_2 such that $\langle q_1, h, q_2 \rangle \in R$, for some h; when needed, we also write $q_1 \approx^p_h q_2$, to explicitly name h. Finally, a TS T_1 is said to be *p-bisimilar* to T_2, written $T_1 \approx^p T_2$, if $q_{10} \approx^p q_{20}$. Again, it is immediate to see that p-bisimilarity is an equivalence relation.

Theorem 5 ([3])**.** *Consider two TSs $T_1 = \langle \Delta_1, Q_1, q_{10}, \rightarrow_1, \mathcal{I}_1 \rangle$ and $T_2 = \langle \Delta_2, Q_2, q_{20}, \rightarrow_2, \mathcal{I}_2 \rangle$ with Δ_1 and Δ_2 infinite. If $T_1 \approx^p T_2$, then for every closed $\mu\mathcal{L}_p$ formula Φ, $T_1 \models \Phi$ if and only if $T_2 \models \Phi$.*

Again, the converse of this theorem does not hold in general, but, we show later that it holds under some general conditions.

For a state-bounded generic TS we can construct a finite faithful abstraction with respect to closed $\mu\mathcal{L}_p$ formulas.

Theorem 6 ([3, 20, 21])**.** *Given a state-bounded generic TS T, there exists a finite state TS T^f that is p-bisimilar to T.*

By the two theorems above, we have that for every state-bounded generic TS T there exists a finite TS T^f which is a faithful abstraction, i.e., for every closed $\mu\mathcal{L}_p$ formula Φ, $T \models \Phi$ if and only if $T^f \models \Phi$. Hence we can use T^f to model check properties of interest over T.

Unfortunately, it is easy to see that in the case of $\mu\mathcal{L}$ or $\mu\mathcal{L}_a$, there exists no finite-state faithful abstraction of T that is *independent from the formula* to verify. Indeed, assume that we have a TS where every transition replaces an object in the active domain with a fresh object. Then, for every bound n on the number of objects in a candidate finite abstraction, we can write a (fixpoint-free) formula saying that there exists a finite run with more than n distinct objects:

$$\begin{array}{l}\exists x_1.\text{LIVE}(x_1) \wedge \langle - \rangle(\exists x_2.\text{LIVE}(x_2) \wedge x_2 \neq x_1 \wedge \\ \quad \langle - \rangle(\exists x_3 \text{LIVE}(x_3) \wedge x_3 \neq x_1 \wedge x_3 \neq x_2 \wedge \\ \quad\ \cdots \\ \quad \langle - \rangle(\exists x_{n+1} \text{LIVE}(x_{n+1}) \wedge x_{n+1} \neq x_1 \wedge \cdots \wedge x_{n+1} \neq x_n)))\end{array}$$

This formula is obviously true in the original TS, but it is false in any finite abstraction using at most n objects [3, 20, 21].

4. Expressiveness over Generic Transition Systems

We start by proving the key result of this section: *on generic TSs, p-bisimilarity implies bisimilarity.*[5]

Theorem 7. *Consider two generic TSs, $T_1 = \langle \Delta_1, Q_1, q_{10}, \rightarrow_1, \mathcal{I}_1 \rangle$ and $T_2 = \langle \Delta_2, Q_2, q_{20}, \rightarrow_2, \mathcal{I}_2 \rangle$, with $|\Delta_1| = |\Delta_2|$ infinite. If there exists a p-bisimulation P between T_1 and T_2, then there exists a bisimulation B between T_1 and T_2.*

Proof. For an n-ary relation $R \subseteq D_1 \times \cdots \times D_n$, define the *projection of R on the i-th component* as: $\pi_i.R \doteq \{d_i \in D_i \mid \exists \langle d_1, \ldots, d_i, \ldots, d_n \rangle \in R\}$.

Since $T_1 \approx^p T_2$, we have that $q_{10} \approx^p_{h_0} q_{20}$, for some suitable isomorphism h_0. Let $\ell : \Delta_1 \mapsto \Delta_2$ be a bijection extending h_0 to the whole Δ_1. Such an ℓ exists because h_0 is a bijection and $|\Delta_1| = |\Delta_2|$. We define the relation $B \subseteq Q_1 \times Q_2$ such that $\langle q_1, q_2 \rangle \in B$ iff: $q_1 \in \pi_1.P$, $q_2 \in \pi_3.P$, $\mathcal{I}(q_1) \sim_\ell \mathcal{I}(q_2)$. We show that B is a bisimulation between T_1 and T_2. To this end, consider the definition of (plain) bisimulation (p. 7) and let $\langle q_1, q_2 \rangle \in B$. Requirement 1 is obviously satisfied.

As to requirement 2, assume that, for some q_1', $q_1 \rightarrow_1 q_1'$. Because, by the definition of B, $q_1 \in \pi_1.P$, we have that, for some h and q_2'', $\langle q_1, h, q_2'' \rangle \in P$. Then, by definition of p-bisimulation, since $q_1 \rightarrow_1 q_1'$, there exists an extension h' of h to $adom(\mathcal{I}_1(q_1'))$ and a state $q_2''' \in Q_2$ such that $q_2'' \rightarrow_2 q_2'''$ and $\langle q_1', h', q_2''' \rangle \in P$. Notice this implies $q_1' \in \pi_1.P$. Let $g : \Delta_1 \mapsto \Delta_2$ be a bijection extending h' to the whole Δ_1. It is immediate to see that g exists. Because of the way h' extends

[5] Notice that in [13] an analogous, though weaker, result is proved: on generic TSs, p-bisimilarity implies a-bisimilarity.

$$\begin{tikzcd}
q_1 \arrow[r] \arrow[d, dashed, "g"] \arrow[dd, dashed, bend right, "\ell"'] & q_1' \arrow[d, dashed, "g"] \\
q_2'' \arrow[r] \arrow[d, dashed, "f = g^{-1} \circ \ell"] & q_2''' \arrow[d, dashed, "f"] \\
q_2 \arrow[r] & q_2'
\end{tikzcd}$$

Figure 2: Relationships among the states q_1, q_1', q_2, q_2', q_2'', q_2''' involved in the proof of Theorem 7.

h and g extends h', we have that $\mathcal{I}_1(q_1) \sim_g \mathcal{I}_2(q_2'')$. Then, since $\mathcal{I}_1(q_1) \sim_\ell \mathcal{I}_2(q_2)$ and $\mathcal{I}_1(q_1) \sim_g \mathcal{I}_2(q_2'')$, by symmetry and transitivity of $\sim$, and closedness of isomorphism under composition, it follows that $\mathcal{I}_2(q_2) \sim_{\ell^{-1} \circ g} \mathcal{I}_2(q_2'')$.

Now, observe that $q_2, q_2'', q_2''' \in \pi_3.P \subseteq Q_2$ (in particular for q_2, this is a consequence of B's definition) and $q_2'' \rightarrow_2 q_2'''$. Thus, since T_2 is generic, for every bijection $f : \Delta_2 \mapsto \Delta_2$ such that $\mathcal{I}_2(q_2'') \sim_f \mathcal{I}_2(q_2)$, there exists a state $q_2' \in Q_2$ such that $q_2 \rightarrow_2 q_2'$ and $\mathcal{I}_2(q_2''') \sim_f \mathcal{I}_2(q_2')$. Consider, in particular, the bijection $f = g^{-1} \circ \ell$. Because $\mathcal{I}_2(q_2) \sim_{\ell^{-1} \circ g} \mathcal{I}_2(q_2'')$, we obviously have that $\mathcal{I}_2(q_2'') \sim_f \mathcal{I}_2(q_2)$. Then, by genericity, there exists $q_2' \in Q_2$ such that $q_2 \rightarrow_2 q_2'$ and $\mathcal{I}_2(q_2''') \sim_f \mathcal{I}_2(q_2')$. For convenience, the relationships among states q_1, q_1', q_2, q_2', q_2'', q_2''' are depicted in Figure 2.

We now show that $\langle q_1', q_2' \rangle \in B$. To this end, recall first that $q_1' \in \pi_1.P$. As to q_2', we need to show that $q_2' \in \pi_3.P$. This is an immediate consequence of the fact that, by B's definition, $q_2 \in \pi_3.P$, and that $q_2 \rightarrow_2 q_2'$. Indeed, because $q_2 \in \pi_3.P$, there exists a tuple $\langle q_1'', h'', q_2 \rangle \in P$; moreover, because $q_2 \rightarrow_2 q_2'$, and P being a p-bisimulation, there must exist $q_1''' \in Q_1$ and h''', such that $q_1'' \rightarrow_1 q_1'''$ and $\langle q_1''', h''', q_2' \rangle \in P$. Thus, $q_2' \in \pi_3.P$. Finally, we prove that $\mathcal{I}_1(q_1') \sim_\ell \mathcal{I}_2(q_2')$. To this end, observe that $\mathcal{I}_2(q_2''') \sim_f \mathcal{I}_2(q_2')$ and $\mathcal{I}_1(q_1') \sim_g \mathcal{I}_2(q_2''')$ (the latter is a consequence of the fact that g is a bijection extending h'). Then, again, by transitivity and symmetry of $\sim$, and by closedness of isomorphisms under composition, we have that $\mathcal{I}_1(q_1') \sim_{g \circ f} \mathcal{I}_2(q_2')$. Thus, since $g \circ f = g \circ g^{-1} \circ \ell = \ell$, it follows that $\mathcal{I}_1(q_1') \sim_\ell \mathcal{I}_2(q_2')$. Requirement 3 can be proven in essentially the same way, using ℓ^{-1} instead of ℓ. $\square$

As an immediate consequence, we have that if two TSs are p-bisimilar, then, being also bisimilar, by invariance with respect to bisimilarity (Theorem 3) they satisfy the same closed $\mu\mathcal{L}$ formulas.

Theorem 8. *Consider two* generic *TSs* $T_1 = \langle \Delta_1, Q_1, q_{10}, \rightarrow_1, \mathcal{I}_1 \rangle$ *and* $T_2 = \langle \Delta_2, Q_2, q_{20}, \rightarrow_2, \mathcal{I}_2 \rangle$ *with* $|\Delta_1| = |\Delta_2|$ *infinite. If* $T_1 \approx^p T_2$ *then for every closed* $\mu\mathcal{L}$ *formula* Φ*, we have that* $T_1 \models \Phi$ *if and only if* $T_2 \models \Phi$*.*

Next, we study the converse of Theorems 3, 4 and 5. In other words, we are interested in understanding for which TSs we have that if two states satisfy exactly the same $\mu\mathcal{L}$, $\mu\mathcal{L}_a$ or $\mu\mathcal{L}$ formulas then they are, respectively, bisimilar, a-bisimilar or p-bisimilar. In particular, we show that such results hold for *generic finite-active-domain TSs*, which are generic TSs with the additional condition that the active domain of every state is finite (though not necessarily bounded by some given b). Such class of TSs includes generic state-bounded TSs, as well as more general TSs obtained by starting from a database and updating it at each step with a finite number of tuples, as typical of, e.g., artifact-centric dynamic systems [3].

We first prove a key property of generic TSs, i.e., the fact that if the interpretations associated with two states of the same TS are isomorphic with respect to the active domain then they are p-bisimilar.

Lemma 9. *If $T = \langle \Delta, Q, q_0, \rightarrow, \mathcal{I} \rangle$ is a generic TS, then for every two states $q_1, q_2 \in Q$ and every bijection $h : D \mapsto D$ with $adom(\mathcal{I}_1(q_1)) \cup adom(\mathcal{I}_2(q_2)) \subseteq D \subseteq \Delta$ such that $\tilde{\mathcal{I}}_1(q_1) \sim_h \tilde{\mathcal{I}}_2(q_2)$ we have $q_1 \approx^p_h q_2$.*

Proof. By co-induction, we can show that the relation $R = \{\langle q_1, h, q_2 \rangle \mid \tilde{\mathcal{I}}_1(q_1) \sim_h \tilde{\mathcal{I}}_2(q_2)\}$ is a p-bisimulation, by exploiting the very definition of generic TS. □

Notice that Lemma 9 leverages on the fact that the two states belong to the *same* generic TS. If this were not the case, i.e., if the states were taken from different TSs, we could not exploit genericity (which relates states of the same TS) and the claim would not hold. Observe also that the converse of Lemma 9 trivially holds, as a consequence of the definition of p-bisimilarity.

Now, we can show the following result for $\mu\mathcal{L}_p$.

Theorem 10. *Consider two* generic finite-active-domain *TSs $T_1 = \langle \Delta_1, Q_1, q_{10}, \rightarrow_1, \mathcal{I}_1 \rangle$ and $T_2 = \langle \Delta_2, Q_2, q_{20}, \rightarrow_2, \mathcal{I}_2 \rangle$ with Δ_1 and Δ_2 infinite. If for every closed $\mu\mathcal{L}_p$ formula Φ, $T_1 \models \Phi$ if and only if $T_2 \models \Phi$, then $T_1 \approx^p T_2$.*

Proof. We show by co-induction that the relation $R = \{\langle q_1, h, q_2 \rangle \mid$ for all $\Phi \in \mu\mathcal{L}_p$. $T_1, q_1 \models \Phi$ iff $T_2, q_2 \models \Phi$ and $\tilde{\mathcal{I}}_1(q_1) \sim_h \tilde{\mathcal{I}}_2(q_2)\}$ is a p-bisimulation. R satisfies the first condition of p-bisimulation by definition. Suppose towards contradiction that it does not satisfy the second condition, i.e., there exist a tuple $\langle q_1, h, q_2 \rangle$ and a state q'_1 such that $q_1 \rightarrow_1 q'_1$ but there is no extension h' of h and no q'_2 such that: $q_2 \rightarrow_1 q'_2$ and $h'|_{adom(\mathcal{I}_1(q_1))}$ coincides with $h|_{adom(\mathcal{I}_1(q_1))}$; $\tilde{\mathcal{I}}_1(q'_1) \sim_{h'} \tilde{\mathcal{I}}_2(q'_2)$; and q'_1 and q'_2 satisfy the same closed $\mu\mathcal{L}_p$ formulas.

Consider the *isomorphism type* of $\mathcal{I}_1(q'_1)$, i.e., the set of interpretations that are isomorphic to $\mathcal{I}_1(q'_1)$. Since $adom(\mathcal{I}_1(q'_1))$ is finite, there exists a closed first-order formula Ψ, which we call *characteristic formula*, with one existentially quantified variable for each object in the active domain, that characterizes the isomorphism type [21]. From Ψ we can construct an open first-order formula $\Psi(\vec{x})$, by leaving open the variables $\vec{x}$ corresponding to objects already in $adom(\mathcal{I}_1(q_1))$. In this way, $\Psi(\vec{x})$ parameterizes the characteristic formula on $\vec{x}$, forcing the objects coming from $adom(\mathcal{I}_1(q_1))$ to persist.

Now, suppose that for each q_2' there is a closed $\mu\mathcal{L}_p$ formula that is true in q_1' but false in q_2'. Notice that all q_2' belonging to the isomorphism type captured by $\Psi(\vec{x})$ are p-bisimilar by genericity (Lemma 9), hence, by p-bisimulation invariance (Theorem 5), they satisfy the same $\mu\mathcal{L}_p$ formulas. Thus, if such a formula exists it is the same for all states q_2'. Let denote this formula by Φ. Then $T_1, q_1 \models \exists\vec{x}.\textsc{live}(\vec{x}) \wedge \langle -\rangle(\Psi(\vec{x}) \wedge \Phi)$ and $T_2, q_2 \models \forall\vec{x}.\textsc{live}(\vec{x}) \supset [-](\Psi(\vec{x}) \supset \neg\Phi)$. Thus q_1 and q_2 do not satisfy the same $\mu\mathcal{L}_p$ formulas, and we obtain a contradiction. The third condition can be proven analogously. □

By considering that $\mu\mathcal{L}_p$ is a subset of $\mu\mathcal{L}$, as an immediate consequence of Theorems 10 and 7, we obtain an analogous result for $\mu\mathcal{L}$.

Theorem 11. *Consider two* generic finite-active-domain *TSs* $T_1 = \langle \Delta_1, Q_1, q_{10}, \rightarrow_1, \mathcal{I}_1\rangle$ *and* $T_2 = \langle \Delta_2, Q_2, q_{20}, \rightarrow_2, \mathcal{I}_2\rangle$ *with* $|\Delta_1| = |\Delta_2|$ *infinite. If for every closed* $\mu\mathcal{L}$ *formula* Φ*, we have that* $T_1 \models \Phi$ *if and only if* $T_2 \models \Phi$*, then* $T_1 \approx T_2$*.*

Proof. The proof exploits the fact that $\mu\mathcal{L}$ extends $\mu\mathcal{L}_p$, and that equivalence with respect to $\mu\mathcal{L}_p$ formulas guarantees p-bisimilarity, which in turn implies bisimilarity for generic finite-active-domain TSs. □

Considering that $\mu\mathcal{L}_a$ is a subset of $\mu\mathcal{L}$, a similar result holds for $\mu\mathcal{L}_a$ as well. Observe that Theorem 3 and Theorem 11, together, can be seen as the lifting to $\mu\mathcal{L}$ of the classical *μ-calculus characterization of bisimulation* in the propositional setting [11]. Analogously, Theorem 5 and Theorem 10 can be seen as the lifting to $\mu\mathcal{L}_p$, and Theorem 4 and Theorem 11 to $\mu\mathcal{L}_a$.

To summarize, considering that $\mu\mathcal{L}_p$ is a subset of $\mu\mathcal{L}_a$, which is a subset of $\mu\mathcal{L}$, given two TSs $T_1 = \langle \Delta_1, Q_1, q_{10}, \rightarrow_1, \mathcal{I}_1\rangle$ and $T_2 = \langle \Delta_2, Q_2, q_{20}, \rightarrow_2, \mathcal{I}_2\rangle$ with $|\Delta_1| = |\Delta_2|$ infinite, a state q_1 of T_1, and a state q_2 of T_2, we have that:

- *always*:

$$\begin{array}{lllll}
q_1 \approx q_2 & \text{implies} & q_1 \approx^{a} q_2 & \text{implies} & q_1 \approx^{p} q_2 \\
q_1 \approx^{p} q_2 & \text{implies} & q_1 \approx^{\mu\mathcal{L}_p} q_2 & & \\
q_1 \approx^{a} q_2 & \text{implies} & q_1 \approx^{\mu\mathcal{L}_a} q_2 & & \\
q_1 \approx q_2 & \text{implies} & q_1 \approx^{\mu\mathcal{L}} q_2 & & \\
q_1 \approx^{\mu\mathcal{L}} q_2 & \text{implies} & q_1 \approx^{\mu\mathcal{L}_a} q_2 & \text{implies} & q_1 \approx^{\mu\mathcal{L}_p} q_2
\end{array}$$

- when T_1 and T_2 are *generic*:

$$q_1 \approx^{p} q_2 \quad \text{equivalent} \quad q_1 \approx^{a} q_2 \quad \text{equivalent} \quad q_1 \approx q_2$$

- when T_1 and T_2 are *generic finite-active-domain*:

$$\begin{array}{lllll}
q_1 \approx^{\mu\mathcal{L}_p} q_2 & \text{equivalent} & q_1 \approx^{p} q_2 & & \\
q_1 \approx^{\mu\mathcal{L}_a} q_2 & \text{equivalent} & q_1 \approx^{a} q_2 & & \\
q_1 \approx^{\mu\mathcal{L}} q_2 & \text{equivalent} & q_1 \approx q_2 & & \\
q_1 \approx^{\mu\mathcal{L}_p} q_2 & \text{equivalent} & q_1 \approx^{\mu\mathcal{L}_a} q_2 & \text{equivalent} & q_1 \approx^{\mu\mathcal{L}} q_2
\end{array}$$

where $q_1 \approx^{\mu\mathcal{L}_p} q_2$ denotes that q_1 and q_2 satisfy the same $\mu\mathcal{L}_p$ formulas, analogously for $\mu\mathcal{L}_a$ and $\mu\mathcal{L}$.

5. Finite-state Faithful Abstractions

In this section, we study verification of $\mu\mathcal{L}$ formulas over state-bounded generic TSs. In particular, as in [20, 21, 3, 9], we aim at obtaining decidability of verification by abstracting infinite TSs into finite-state ones. The general idea is to take advantage of what has been shown in the previous section, namely that $\mu\mathcal{L}$ is invariant with respect to p-bisimulation (Theorem 8) Based on this result it might appear that one could search for a finite-state generic TS that is p-bisimilar to the original infinite-state one, and then perform verification on this, as, e.g., in [21]. However, such a finite-state generic TS cannot exist. Indeed, for applying Theorem 8, the finite-state generic TS needs to have an infinite object domain. Unfortunately, by the very definition of genericity, every generic TS with an infinite object domain must be infinite-state: if there exists a transition, then all, infinitely many, isomorphic transitions must exist, each producing a different successor state. To overcome this we need a stronger version of the invariance of $\mu\mathcal{L}$ with respect to p-bisimulation, which also takes into account that we cannot have a finite abstraction that preserves $\mu\mathcal{L}$ and is independent from the formula to check, as discussed at the end of Section 3. The next result establishes such a stronger version of invariance.

Theorem 12. *Consider a finite set Vars of variables and two generic TSs, $T_1 = \langle \Delta_1, Q_1, q_{10}, \rightarrow_1, \mathcal{I}_1 \rangle$, bounded by b and with infinite Δ_1, and $T_2 = \langle \Delta_2, Q_2, q_{20}, \rightarrow_2, \mathcal{I}_2 \rangle$, with $|\Delta_2| \geq 2b + |\mathit{Vars}|$, such that $T_1 \approx^p T_2$. Then, for every closed $\mu\mathcal{L}$ formula Φ with variables renamed apart*[6] *and belonging to Vars, we have that $T_1 \models \Phi$ if and only if $T_2 \models \Phi$.*

To prove Theorem 12, we first establish the claim for the simpler logic $\mathcal{L}$, which is $\mu\mathcal{L}$ without fixpoint constructs. We then generalize it to the infinitary version of $\mathcal{L}$, which captures $\mu\mathcal{L}$, by using a well-known line of reasoning in μ-calculus, see [39, 11] or [21, Lemma 2].

We start by showing a generalization of a classical result in Databases, see, e.g., [30, Theorem 5.6.3]. We denote by $\mathit{vars}(\Phi)$ the set of first-order variables of a formula Φ, and by $\mathit{free}(\Phi)$ the set of its free variables. Note that, for closed formulas, $\mathit{free}(\Phi)$ is empty. For convenience, given an interpretation $\mathcal{I} = \langle \Delta, \cdot^{\mathcal{I}} \rangle$ and a set $D \subseteq \Delta$, we define $\tilde{\mathcal{I}}^D = \langle \mathit{adom}(\mathcal{I}) \cup D, \cdot^{\mathcal{I}} \rangle$. That is, $\tilde{\mathcal{I}}^D$ is the restriction of $\mathcal{I}$ to its active domain, with the interpretation domain augmented with the elements of D.

Lemma 13. *Every first-order formula φ can effectively be rewritten as a formula φ', called the* domain-independent version of φ, *with $\mathit{vars}(\varphi') = \mathit{vars}(\varphi)$, $\mathit{free}(\varphi') = \mathit{free}(\varphi)$, and quantified variables ranging only over the active domain, such that, for every interpretation $\mathcal{I} = \langle \Delta, \cdot^{\mathcal{I}} \rangle$ with $|\Delta| \geq |\mathit{adom}(\mathcal{I})| + |\mathit{vars}(\varphi)|$ and for every valuation v, we have that $\mathcal{I}, v \models \varphi$ iff $\tilde{\mathcal{I}}^D, v \models \varphi'$, where $D = \textsc{img}(v|_{\mathit{free}(\varphi)})$.*

[6] This means that no two quantifiers in the formula range over the same variable.

Proof. By induction on the structure of φ. We assume, without loss of generality, that all variables of φ are renamed apart.

If $\varphi = (t_1 = t_2)$ or $\varphi = F(t_1, \ldots, t_n)$, with t_i $(i = 1, \ldots, n)$ arbitrary terms (i.e., variables or constants), we let $\varphi' = \varphi$. In these cases the thesis follows immediately. Boolean connectives, similarly, propagate unchanged from φ to φ'. If $\varphi = \exists x.\phi$, let

$$\varphi' \doteq (\exists x.\textsc{live}(x) \wedge \phi') \vee (\bigvee_{y \in \mathit{free}(\varphi)} (x = y) \wedge \phi') \vee \psi,$$

where

- ϕ' is the domain-independent version of ϕ;
- ψ is obtained from ϕ' by replacing the atomic formulas $x = x$, $x = y$, $x = c$, and $F(\ldots, x, \ldots)$, where y is a variable distinct from x, c is a constant symbol, and F is a predicate symbol, respectively with $\top$, $\bot$, $\bot$, and $\bot$.

We have that $\mathcal{I}, v \models \exists x.\phi$ iff there exists $d \in \Delta$ such that $\mathcal{I}, v[x/d] \models \phi$. By induction hypothesis, this holds iff $\tilde{\mathcal{I}}^{D_d}, v[x/d] \models \phi'$, with $D_d \doteq \textsc{img}(v[x/d]|_{\mathit{free}(\phi)})$. Notice that $\textsc{img}(v[x/d]|_{\mathit{free}(\phi)}) = \textsc{img}(v|_{\mathit{free}(\varphi)}) \cup \{d\}$, as $\mathit{free}(\phi) = \mathit{free}(\varphi) \cup \{x\}$. We distinguish three cases: *(i)* $d \in \mathit{adom}(\mathcal{I})$; *(ii)* $d \in \textsc{img}(v|_{\mathit{free}(\varphi)})$; *(iii)* $d \notin \mathit{adom}(\mathcal{I}) \cup \textsc{img}(v|_{\mathit{free}(\varphi)})$. Notice that case *(iii)* is possible, in general, only if $|\Delta| \geq |\mathit{adom}(\mathcal{I})| + |\mathit{vars}(\varphi)|$.

In case *(i)*, we have that $\tilde{\mathcal{I}}^{D_d} \doteq \mathit{adom}(\mathcal{I}) \cup \textsc{img}(v[x/d]|_{\mathit{free}(\phi)}) = \mathit{adom}(\mathcal{I}) \cup \textsc{img}(v|_{\mathit{free}(\varphi)}) \cup \{d\} = \mathit{adom}(\mathcal{I}) \cup \textsc{img}(v|_{\mathit{free}(\varphi)}) \doteq \tilde{\mathcal{I}}^{D}$. Consequently, $\tilde{\mathcal{I}}^{D_d}, v[x/d] \models \phi'$ iff $\tilde{\mathcal{I}}^{D}, v[x/d] \models \phi'$. Thus, it can be checked that there exists $d \in \mathit{adom}(\mathcal{I})\ \mathit{such that} \mathcal{I}, v[x/d] \models \phi$ iff $\tilde{\mathcal{I}}^{D}, v \models \exists x.\textsc{live}(x) \wedge \phi'$. Also in case *(ii)*, by the same argument as above, $\tilde{\mathcal{I}}^{D_d} = \tilde{\mathcal{I}}^{D}$. Thus, we have there exists $d \in \textsc{img}(v|_{\mathit{free}(\varphi)})$ such that $\tilde{\mathcal{I}}^{D_d}, v[x/d] \models \phi'$ iff $\tilde{\mathcal{I}}^{D}, v \models \bigvee_{y \in \mathit{free}(\varphi)}(x = y) \wedge \phi'$.

For case *(iii)*, we first show that: $\tilde{\mathcal{I}}^{D_d}, v[x/d] \models \phi'$ iff $\tilde{\mathcal{I}}^{D_d}, v[x/d] \models \psi$. To this end, observe the following. First, by replacing $(x = x)$ with $\top$ in ϕ', we obtain a formula equivalent to ϕ'. Second, notice that y can occur either free or quantified in ϕ' (these two cases are mutually exclusive, as we assume variables are renamed apart). If y occurs free, let d' be the object assigned to y by $v[x/d]$ (and v). Notice that y occurs free also in φ, thus $d' \in \textsc{img}(v|_{\mathit{free}(\varphi)})$. Then, since $d \notin \mathit{adom}(\mathcal{I}) \cup \textsc{img}(v|_{\mathit{free}(\varphi)})$, it follows that $v[x/d](x) = d \neq d' = v[x/d](y)$. Thus, $\tilde{\mathcal{I}}^{D_d}, v[x/d] \not\models (x = y)$. The same occurs if y is quantified as, in this case, y ranges over the active domain of $\tilde{\mathcal{I}}^{D_d}$, which is the same at that of $\mathcal{I}$, i.e., $\mathit{adom}(\mathcal{I})$. Therefore, since $\mathit{adom}(\mathcal{I})$ does not include d, by replacing every occurrence of $(x = y)$ in ϕ' with $\bot$, we obtain a formula that is equivalent to ϕ' with respect to $\tilde{\mathcal{I}}^{D_d}$ and $v[x/d]$. Thirdly, since the interpretation of c is in $\mathit{adom}(\mathcal{I})$, while $d \notin \mathit{adom}(\mathcal{I})$, then we can replace all occurrences of $(x = c)$ in ϕ' with $\bot$, and obtain, again, a formula equivalent to ϕ' with respect to $\tilde{\mathcal{I}}^{D_d}$ and $v[x/d]$. Finally, as $d \notin \mathit{adom}(\mathcal{I})$, we can replace also all occurrences of $F(\ldots, x, \ldots)$ in ϕ' with $\bot$, and obtain a formula equivalent to ϕ' with respect to $\tilde{\mathcal{I}}^{D_d}$ and $v[x/d]$.

Since the replacements above are those that transform ϕ' into ψ, we have that $\tilde{\mathcal{I}}^{D_d}, v[x/d] \models \phi'$ iff $\tilde{\mathcal{I}}^{D_d}, v[x/d] \models \psi$. However, ψ does not contain any occurrence of x, so we have that $\tilde{\mathcal{I}}^{D_d}, v[x/d] \models \psi$ iff $\tilde{\mathcal{I}}^{D_d}, v \models \psi$. Moreover, observe that, by induction hypothesis, the quantified variables of ψ range only over the active domain, and that $\tilde{\mathcal{I}}^{D_d}$ and $\tilde{\mathcal{I}}^{D}$ differ only for the fact that the domain of the former contains d, outside the active domain. Thus, it can be checked that $\tilde{\mathcal{I}}^{D_d}$ and $\tilde{\mathcal{I}}^{D}$ are indistinguishable through ψ, and hence $\tilde{\mathcal{I}}^{D_d}, v[x/d] \models \psi$ iff $\tilde{\mathcal{I}}^{D}, v \models \psi$. We can therefore conclude that there exists $d \in \Delta \setminus (adom(\mathcal{I}) \cup \text{IMG}(v|_{free(\varphi)}))$ such that $\mathcal{I}, v[x/d] \models \phi$ iff $\tilde{\mathcal{I}}^{D}, v \models \psi$. Since *(i)*, *(ii)*, and *(iii)* cover all possible cases for $d \in \Delta$, the thesis easily follows. □

The next result establishes invariance of $\mathcal{L}$ under p-bisimulation even between a TS with infinite object domain and a TS with finite object domain (provided the latter TS contains a number *large enough* of objects in its object domain).

Lemma 14. *Let*

- *Vars be a finite set of variables;*
- $T_1 = \langle \Delta_1, Q_1, q_{10}, \rightarrow_1, \mathcal{I}_1 \rangle$ *a generic TS bounded by b and with infinite* Δ_1*;*
- $T_2 = \langle \Delta_2, Q_2, q_{20}, \rightarrow_2, \mathcal{I}_2 \rangle$ *a generic TS with* $|\Delta_2| \geq 2b + |Vars|$*;*
- $q_1 \in Q_1$ *and* $q_2 \in Q_2$ *two states such that, for some h,* $q_1 \approx_h^p q_2$*;*
- v_1*,* v_2 *two individual variable valuations, mapping variables in Vars to* Δ_1 *and* Δ_2*, respectively;*
- Φ *an open* $\mathcal{L}$ *formula with variables renamed apart and belonging to Vars.*

If there exists a bijection $\hat{h}$ *between* $adom(\mathcal{I}_1(q_1)) \cup \text{IMG}(v_1|_{free(\Phi)})$ *and* $adom(\mathcal{I}_2(q_2)) \cup \text{IMG}(v_2|_{free(\Phi)})$*, whose restriction* $\hat{h}|_{adom(\mathcal{I}_1(q_1))}$ *coincides with h and such that, for every individual variable* $x \in free(\Phi)$*,* $\hat{h}(v_1(x)) = v_2(x)$*, then* $T_1, q_1, v_1 \models \Phi$ *if and only if* $T_2, q_2, v_2 \models \Phi$*.*

Proof. By induction on the structure of Φ. For $\Phi = \varphi$, by Lemma 13, we have that $\mathcal{I}_1(q_1), v_1 \models \varphi$ iff $\tilde{\mathcal{I}}_1^{D_1}(q_1), v_1 \models \varphi'$, with $D_1 = \text{IMG}(v_1|_{free(\varphi)})$. Moreover, by the existence of $\hat{h}$, it follows that, up to object renaming, $\tilde{\mathcal{I}}_1^{D_1}(q_1)$ and v_1 match, respectively, $\tilde{\mathcal{I}}_2^{D_2}(q_2)$ and v_2, with $D_2 = \text{IMG}(v_2|_{free(\varphi)})$. Thus, we have that $\tilde{\mathcal{I}}_1^{D_1}(q_1), v_1 \models \varphi'$ iff $\tilde{\mathcal{I}}_2^{D_2}(q_2), v_2 \models \varphi'$. But then, observing that $|\Delta_2| \geq |adom(\mathcal{I}_2)| + |Vars|$, (because by boundedness, $|adom(\mathcal{I}_2)| \leq b$), by Lemma 13 it follows that $\tilde{\mathcal{I}}_2^{D_2}(q_2), v_2 \models \varphi'$ iff $\mathcal{I}_2(q_2), v_2 \models \varphi$.

Boolean connectives are straightforward.

For $\Phi = \exists y.\Phi'$, suppose that $T_1, q_1, v_1 \models \Phi$ (the other direction is proven in an analogous way). This implies that there exists an object $d_1 \in \Delta_1$ such that $T_1, q_1, v_1[y/d_1] \models \Phi'$. The following cases are possible: either $d_1 \in adom(\mathcal{I}_1(q_1)) \cup \text{IMG}(v_1|_{free(\Phi)})$ or $d_1 \notin adom(\mathcal{I}_1(q_1)) \cup \text{IMG}(v_1|_{free(\Phi)})$. In the former case, using the bijection $\hat{h}$, it can be easily proven by induction that, for $d_2 = \hat{h}(d_1)$, we have $T_2, q_2, v_2[y/d_2] \models \Phi'$, that is, $T_2, q_2, v_2 \models \exists y.\Phi'$.

For the latter case, consider a bijection $\hat{h}' : (\mathit{adom}(\mathcal{I}_1(q_1)) \cup \textsc{img}(v_1|_{\mathit{free}(\Phi)})) \uplus \{d_1\} \mapsto (\mathit{adom}(\mathcal{I}_2(q_2)) \cup \textsc{img}(v_2|_{\mathit{free}(\Phi)})) \uplus \{d_2\}$ (with $\uplus$ denoting the disjoint union operator), obtained by extending $\hat{h}$ to d_1 in such a way that $\hat{h}(d_1) = d_2$, with $d_2 \in \Delta_2 \setminus (\mathit{adom}(\mathcal{I}_2(q_2)) \cup \textsc{img}(v_2|_{\mathit{free}(\Phi)}))$. It can be seen, by a cardinality argument, that such a bijection exists. Indeed, since $\mathit{free}(\Phi) \subseteq \mathit{vars}(\Phi)$, $y \notin \mathit{free}(\Phi)$, and $|\mathit{free}(\Phi)| \geq |\textsc{img}(v_1|_{\mathit{free}(\Phi)})|$, we have that $|\mathit{vars}(\Phi)| \geq |\textsc{img}(v_1|_{\mathit{free}(\Phi)})|+1$. Thus, since $|\Delta_2| \geq |\mathit{adom}(\mathcal{I}_2)|+|\mathit{Vars}|$ and $\mathit{vars}(\Phi) \subseteq \mathit{Vars}$, we also have that $|\Delta_2| \geq |\mathit{adom}(\mathcal{I}_2)| + |\textsc{img}(v_1|_{\mathit{free}(\Phi)})| + 1$. This is enough to ensure that d_2 as above exists, which in turn guarantees the existence of $\hat{h}'$. Now, consider the assignments $v_1[y/d_1]$ and $v_2[y/d_2]$. Because $y \in \mathit{free}(\Phi')$, we have that $\hat{h}'$ defines a bijection from $\mathit{adom}(\mathcal{I}_1(q_1)) \cup \textsc{img}(v_1[y/d_1]|_{\mathit{free}(\Phi')})$ to $\mathit{adom}(\mathcal{I}_2(q_2)) \cup \textsc{img}(v_2[y/d_2]|_{\mathit{free}(\Phi')})$. Also, it is immediate to see that $\hat{h}'$ satisfies, by construction, the lemma hypothesis, with respect to $v_1[y/d_1]$ and $v_2[y/d_2]$. Thus, by induction hypothesis, we can conclude that $T_2, q_2, v_2[y/d_2] \models \Phi'$, that is, $T_2, q_2, v_2 \models \exists y.\Phi'$.

For $\Phi = \langle - \rangle \Phi'$, suppose that $T_1, q_1, v_1 \models \langle - \rangle \Phi'$. Then, there exists a transition $q_1 \rightarrow_1 q_1'$ such that $T_1, q_1', v_1 \models \Phi'$. Since $q_1 \approx^p_h q_2$, there exist: *(i)* a transition $q_2 \rightarrow_2 q_2'$, and *(ii)* a bijection $h' : \mathit{adom}(\mathcal{I}_1(q_1)) \cup \mathit{adom}(\mathcal{I}_1(q_1')) \mapsto \mathit{adom}(\mathcal{I}_2(q_2)) \cup \mathit{adom}(\mathcal{I}_2(q_2'))$ such that $h'|_{\mathit{adom}(\mathcal{I}_1(q_1))}$ coincides with h, and $q_1' \approx^p_{h'|_{\mathit{adom}(\mathcal{I}_1(q_1'))}} q_2'$. We would like to inductively apply the lemma using $\Phi', q_1', q_2', h'|_{\mathit{adom}(\mathcal{I}_1(q_1'))}, v_1, v_2$, and a suitable bijection $\hat{h}'$ that extends $h'|_{\mathit{adom}(\mathcal{I}_1(q_1'))}$ and satisfies the lemma hypothesis. Unfortunately, for q_1' and q_2', such a $\hat{h}'$ may not exist, in general. However, we can show that there exist another state $q_2'' \in Q$ bisimilar to q_1' and such that $q_2 \rightarrow_2 q_2''$, and a bijection $\hat{h}'$ such that the lemma applies to $\Phi', q_1', q_2'', \hat{h}', v_1, v_2$. This, by induction hypothesis, implies that $T_2, q_2'', v_2 \models \Phi'$, thus that $T_2, q_2, v_2 \models \Phi$. The rest of the proof is devoted to derive q_2'' and $\hat{h}'$.

Let $g : \mathit{adom}(\mathcal{I}_1(q_1)) \cup \mathit{adom}(\mathcal{I}_1(q_1')) \cup \textsc{img}(v_1|_{\mathit{free}(\Phi')}) \mapsto \mathit{adom}(\mathcal{I}_2(q_2)) \cup \mathit{adom}(\mathcal{I}_2(q_2')) \cup \textsc{img}(v_2|_{\mathit{free}(\Phi')})$, be a bijection extending $\hat{h}$. This exists because: $|\mathit{adom}(\mathcal{I}_1(q_1))| = |\mathit{adom}(\mathcal{I}_2(q_2))|$ (by the existence of h); $|\textsc{img}(v_1|_{\mathit{free}(\Phi')})| = |\textsc{img}(v_2|_{\mathit{free}(\Phi')})|$; and $|\mathit{adom}(\mathcal{I}_1(q_1'))| = |\mathit{adom}(\mathcal{I}_2(q_2'))|$ (by the existence of h and h'). Consider the composition $f = g|_{\mathit{adom}(\mathcal{I}_1(q_1)) \cup \mathit{adom}(\mathcal{I}_1(q_1'))} \circ h'^{-1}$. Being a composition of bijections, f is a bijection too, namely from $\mathit{adom}(\mathcal{I}_2(q_2)) \cup \mathit{adom}(\mathcal{I}_2(q_2'))$ into itself, as g extends $\hat{h}$. Moreover, since both g and h' extend h, we have that f is the identity on $\mathit{adom}(\mathcal{I}_2(q_2))$. Consequently, $f|_{\mathit{adom}(\mathcal{I}_2(q_2'))}$ is a bijection from $\mathit{adom}(\mathcal{I}_2(q_2'))$ to $\mathit{adom}(\mathcal{I}_2(q_2'))$. Also, f can obviously be extended to a bijection f' from Δ_2 to Δ_2. For convenience, Figure 3 depicts the functions defined above and the relationships among their domain and images. A directed arc from set A to set B, labelled with function names, expresses that set A is mapped into set B through (any of) the labeling functions. The symbol $\leq$ is used to express that the lefthand function extends the righthand one on the origin set of the arc. Notice that any function mentioned in an arc label (including those that are extended) maps the origin to the destination set.

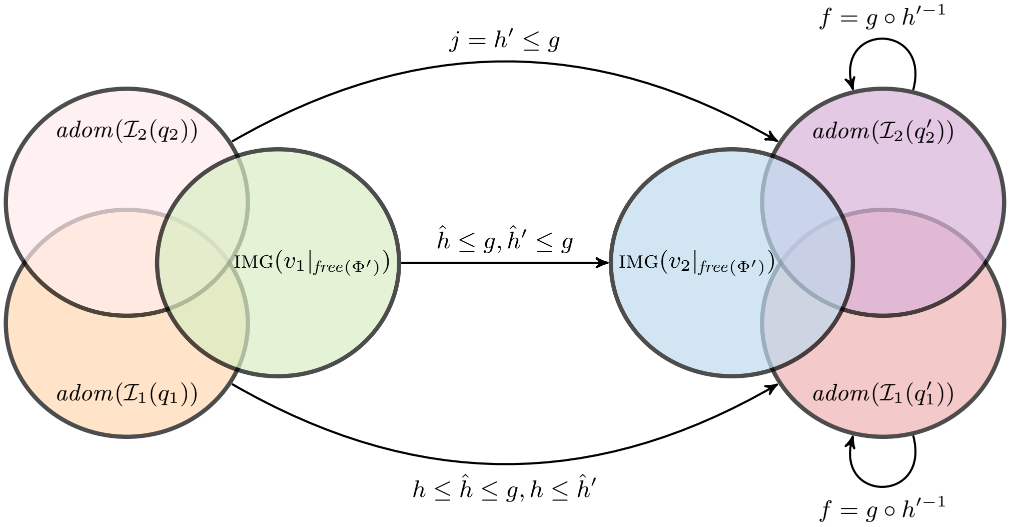
$f = g \circ h'^{-1}$
$j = h' \leq g$
$adom(\mathcal{I}_2(q_2))$
$adom(\mathcal{I}_2(q'_2))$
$\hat{h} \leq g, \hat{h}' \leq g$
$\text{IMG}(v_1|_{free(\Phi')})$
$\text{IMG}(v_2|_{free(\Phi')})$
$adom(\mathcal{I}_1(q_1))$
$adom(\mathcal{I}_1(q'_1))$
$h \leq \hat{h} \leq g, h \leq \hat{h}'$
$f = g \circ h'^{-1}$

Figure 3: Relationships among sets and functions used in the proof of Lemma 14.

$$\begin{tikzcd}
q_1 \arrow[r] \arrow[d, dashed, "h \leq h'"'] & q'_1 \arrow[d, dashed, "h'"] \arrow[dd, dashed, bend left, "j"] \\
q_2 \arrow[r] \arrow[dr] & q'_2 \arrow[d, dashed, "f"] \\
 & q''_2
\end{tikzcd}$$

Figure 4: Relationships among the states q_1, q'_1, q_2, q'_2 involved in the proof of Lemma 14.

Now, consider q_2 and q_2'. Because T_2 is generic, f' is a bijection from Δ_2 to Δ_2 such that $\mathcal{I}_2(q_2) \sim_{f'} \mathcal{I}_2(q_2)$, then, by Definition 1, there exists a state $q_2'' \in Q_2$ such that $q_2 \rightarrow_2 q_2''$ and $\mathcal{I}_2(q_2') \sim_{f'} \mathcal{I}_2(q_2'')$. This, in turn, implies that $\tilde{\mathcal{I}}_2(q_2') \sim_{f'|_{adom \mathcal{I}_2(q_2')}} \tilde{\mathcal{I}}_2(q_2'')$, or, equivalently, $\tilde{\mathcal{I}}_2(q_2') \sim_{f|_{adom \mathcal{I}_2(q_2')}} \tilde{\mathcal{I}}_2(q_2'')$. Then, by Lemma 9, we have that $q_2' \approx^p_{f|_{adom(\mathcal{I}_2(q_2')))}} q_2''$. Now, consider the bijection $j = f|_{adom(\mathcal{I}_2(q_2')))} \circ h'|_{adom(\mathcal{I}_1(q_1'))}$. Because $q_1' \approx^p_{h'|_{adom(\mathcal{I}_1(q_1'))}} q_2'$ and $q_2' \approx^p_{f|_{adom(\mathcal{I}_2(q_2')))}} q_2''$, by transitivity of bisimulation, we have that $q_1' \approx^p_j q_2''$. However, since by the definition of f, $f|_{adom(\mathcal{I}_2(q_2')))} = g \circ h'^{-1}|_{adom(\mathcal{I}_2(q_2'))}$, it follows that $j = g \circ h'^{-1}|_{adom(\mathcal{I}_2(q_2'))} \circ h'|_{adom(\mathcal{I}_1(q_1'))}$, it follows that $j = g|_{adom(\mathcal{I}_1(q_1'))}$. Finally, let $\hat{h}' = g|_{adom(\mathcal{I}_1(q_1')) \cup \textsc{img}(v_1|_{\mathit{free}(\Phi')})}$. Obviously, since g is a bijection, so is $\hat{h}'$. Further, because g matches $\hat{h}$ on $\textsc{img}(v_1|_{\mathit{free}(\Phi')})$, also $\hat{h}'$ matches $\hat{h}$ on $\textsc{img}(v_1|_{\mathit{free}(\Phi')})$, therefore, for $x \in \mathit{free}(\Phi')$, we have that $\hat{h}'(v_1(x)) = v_2(x)$. The relationships among q_1, q_1', q_2, q_2' are depicted for convenience in Figure 4). To conclude the proof, it is enough to observe that $\hat{h}'$ is a bijection between $adom(\mathcal{I}_1(q_1')) \cup \textsc{img}(v_1|_{\mathit{free}(\Phi')})$ and $adom(\mathcal{I}_2(q_2'')) \cup \textsc{img}(v_2|_{\mathit{free}(\Phi')})$, and that $\hat{h}'$ matches j on $adom(\mathcal{I}_1(q_1'))$, as $j = g|_{adom(\mathcal{I}_1(q_1'))}$. The other direction is proven in a similar way. $\square$

We now generalize the above result to $\mu\mathcal{L}$ formulas.

Lemma 15. *Lemma 14 holds also for $\mu\mathcal{L}$ formulas Φ closed on predicate variables.*

Proof. By inspection of its proof, it is immediate to see that Lemma 14 holds also for Φ belonging to the *infinitary version* of $\mathcal{L}$ [39]. This is an extension of $\mathcal{L}$ that supports arbitrary infinite disjunction and conjunction of formulas sharing the same free variables. Let Ψ be a possibly infinite set of open $\mathcal{L}$ formulas. Given a transition system T, a variable valuation v, and a state q of T, we have that $T, q, v \models \bigvee \Psi$ if and only if $T, q, v \models \psi$ for some $\psi \in \Psi$. Analogously, we have that $T, q, v \models \bigwedge \Psi$ if and only if $T, q, v \models \psi$ for all $\psi \in \Psi$. Now, we can express *approximates* of $\mu\mathcal{L}$ fixpoint formulas in infinitary $\mathcal{L}$ in a standard way [11, 39]. Let us denote the approximate of index α by $\mu^\alpha Z.\Phi$, for least fixpoint formulas $\mu Z.\Phi$, and $\nu^\alpha Z.\Phi$, for greatest fixpoint formulas $\nu Z.\Phi$. Then, such approximates are as follows:

$$\begin{array}{rcl@{\qquad}rcl}
\mu^0 Z.\Phi & = & \mathtt{false} & \nu^0 Z.\Phi & = & \mathtt{true} \\
\mu^{\beta+1} Z.\Phi & = & \Phi[Z/\mu^\beta Z.\Phi] & \nu^{\beta+1} Z.\Phi & = & \Phi[Z/\nu^\beta Z.\Phi] \\
\mu^\lambda Z.\Phi & = & \bigvee_{\beta<\lambda} \mu^\beta Z.\Phi & \nu^\lambda Z.\Phi & = & \bigwedge_{\beta<\lambda} \nu^\beta Z.\Phi
\end{array}$$

where λ is a limit ordinal, and the notation $\Phi[Z/\mu^\beta Z.\Phi]$ (resp. $\Phi[Z/\nu^\beta Z.\Phi]$) denotes the formula obtained from Φ by replacing each occurrence of Z by $\mu^\beta Z.\Phi$ (resp. $\nu^\beta Z.\Phi$). By the Tarski-Knaster Theorem [38], given a transition system T and a state q of T, the fixpoints and their approximates are connected by the following properties:

- $q \in (\mu Z.\Phi)^{T}_{(v,V)}$ if and only if there exists an ordinal α such that $q \in (\mu^{\alpha} Z.\Phi)^{T}_{(v,V)}$ and, for every $\beta < \alpha$, it holds that $q \notin (\mu^{\beta} Z.\Phi)^{T}_{(v,V)}$;
- $q \notin (\nu Z.\Phi)^{T}_{(v,V)}$ if and only if there exists an ordinal α such that $q \notin (\nu^{\alpha} Z.\Phi)^{T}_{(v,V)}$ and, for every $\beta < \alpha$, it holds that $q \in (\nu^{\beta} Z.\Phi)^{T}_{(v,V)}$.

Hence every $\mu\mathcal{L}$ formula, closed on predicate variables, can be written as an infinitary $\mathcal{L}$ formula, thus implying the thesis. □

The proof of Theorem 12 is a direct consequence of Lemma 14. To see this, observe that Theorem 12 is, in fact, a specialization of Lemma 14, to the case where $\mu\mathcal{L}$ formulas are closed on first-order predicates, and $q_1 = q_{10}$ and $q_2 = q_{20}$.

We can also show constructively that every state-bounded and generic TS can be abstracted into a p-bisimilar finite-state generic TS with a (finite) object domain of a suitable size.

Theorem 16. *Consider a transition system $T_1 = \langle \Delta_1, Q_1, q_{10}, \rightarrow_1, \mathcal{I}_1 \rangle$ that is generic, bounded by b, and with infinite Δ_1. Then, for every $k \geq 0$, there exists a finite-state generic TS $T_2 = \langle \Delta_2, Q_2, q_{20}, \rightarrow_2, \mathcal{I}_2 \rangle$, with $|\Delta_2| = 2b + k$ such that $T_1 \approx^p T_2$.*

Proof. T_2 is defined as follows. The object domain Δ_2 is a subset of Δ_1 such that $|\Delta_2| = 2b + k$ and $adom(\mathcal{I}_1(q_{10})) \subseteq \Delta_2$ (notice that $|adom(\mathcal{I}_1(q_{10}))| \leq b$). The set of states is $Q_2 = Int^{\mathcal{F},C}_{\Delta_2}$, which is the (finite) set of interpretations of $\mathcal{F}$ and C over Δ_2. The initial state q_{20} is the interpretation such that $\tilde{q}_{20} = \tilde{\mathcal{I}}_1(q_{10})$. The transition relation $\rightarrow_2$ is such that $q_2 \rightarrow_2 q_2'$ iff there exist two states $q_1, q_1' \in Q_1$ such that $q_1 \rightarrow_1 q_1'$, $\tilde{\mathcal{I}}_1(q_1) \sim_h \tilde{\mathcal{I}}_2(q_2)$, and $\tilde{\mathcal{I}}_1(q_1') \sim_h \tilde{\mathcal{I}}_2(q_2')$, for some isomorphism h (notice that here genericity comes into play).[7] Finally, $\mathcal{I}_2$ is the identity function. Obviously, T_2 is finite. Moreover, it can be shown that T_2 is generic and that for each state $q_1 \in Q_1$ and every state q_2 such that $\tilde{\mathcal{I}}_1(q_1) \sim \tilde{\mathcal{I}}_2(q_2)$, including the initial states q_{10} and q_{20}, we have that $q_1 \approx^p q_2$. □

As a direct consequence of Theorems 12 and 16, we obtain:

Theorem 17. *Given a finite set Vars of variables and a generic TS $T_1 = \langle \Delta_1, Q_1, q_{10}, \rightarrow_1, \mathcal{I}_1 \rangle$, bounded by b and with infinite Δ_1, there exists a TS $T_2 = \langle \Delta_2, Q_2, q_{20}, \rightarrow_2, \mathcal{I}_2 \rangle$, with $|\Delta_2| \geq 2b + |Vars|$, such that $T_1 \approx^p T_2$ and, hence, such that for every closed $\mu\mathcal{L}$ formula Φ with variables renamed apart and belonging to Vars, we have that $T_1 \models \Phi$ if and only if $T_2 \models \Phi$.*

Obviously, the case of interest is when Δ_2, and hence the TS T_2, is finite. In this case, the finite T_2 is effectively computable (as in the proof of Theorem 16) when the interpretation $adom(\mathcal{I}_1(q_{10}))$ of the initial state of T_1 restricted to the

[7] Actually, Q_2 can be restricted to the set of states in $Int^{\mathcal{F},C}_{\Delta_2}$ reachable through $\rightarrow_2$.

active domain is known, and one can effectively check whether there exist two states q_1 and q_1' such that $q_1 \rightarrow_1 q_1'$, $\tilde{\mathcal{I}}_1(q_1) = \tilde{\mathcal{I}}_2(q_2)$, and $\tilde{\mathcal{I}}_1(q_1') = \tilde{\mathcal{I}}_2(q_2')$. If T_2 can be effectively computed, Theorem 17 shows decidability of verification of $\mu\mathcal{L}$ formulas. This is the case, e.g., for TSs induced by models of situation calculus bounded action theories (see Section 7).

6. Undecidability of Linear-Time Verification

We now consider linear-time verification of generic transition systems against properties expressed in a first-order variant of LTL with active domain quantification, called LTL-FO$_a$. This logic can be seen as the LTL version of $\mu\mathcal{L}_a$. We show that, differently from the case of $\mu\mathcal{L}_a$ and $\mu\mathcal{L}$, in the linear-time setting boundedness is not sufficient to obtain decidability of verification. This implicitly yields that, surprisingly, lifting such temporal logics to a first-order setting does *not* retain the well-known property that μ-calculus captures LTL.

The logic LTL-FO$_a$ extends propositional LTL with the possibility of querying the system states using first-order formulas with (active domain) quantification across. The syntax of LTL-FO$_a$ is:

$$\Phi ::= \varphi \mid \neg\Phi \mid \Phi_1 \wedge \Phi_2 \mid \exists x.\textsc{live}(x) \wedge \Phi \mid \mathbf{X}\,\Phi \mid \Phi_1 \,\mathbf{U}\, \Phi_2$$

where φ is a first-order formula expressed using predicates in $\mathcal{F}$ and constants in C. We make use of the following standard abbreviations: *(i)* $\Phi_1 \vee \Phi_2 = \neg(\neg\Phi_1 \wedge \neg\Phi_2)$, *(ii)* $\forall x.\textsc{live}(x) \supset \Phi = \neg\exists x.(\textsc{live}(x) \wedge \neg\Phi)$, *(iii)* $\mathbf{F}\,\Phi = \mathtt{true}\,\mathbf{U}\,\Phi$, *(iv)* $\mathbf{G}\,\Phi = \neg\,\mathbf{F}\,\neg\Phi$.

Formulas of LTL-FO$_a$ are interpreted over (infinite runs of) transition systems with first-order states (cf. Section 2), with the additional requirement that they must be *serial*, i.e., every state has at least one successor state. An *(infinite) run* τ over a (serial) TS $T = \langle \Delta, Q, q_0, \rightarrow, \mathcal{I}\rangle$ is an infinite sequence $q_0 q_1 \cdots$ of states in Q, where the first state of the sequence corresponds to the initial state of T, and for every $i \in \mathbb{N}$, it is true that $q_i \rightarrow q_{i+1}$. Given $j \in \mathbb{N}$, by $\tau(j)$ we denote the j-th state q_j of τ.

In details, an LTL-FO$_a$ formula is interpreted over a run τ of T relatively to a position $i \in \mathbb{N}$. Since LTL-FO$_a$ formulas may have free variables, we also use an individual variable valuation v. We then inductively define when τ *satisfies an* LTL-FO$_a$ *formula* Φ *at position* i *under* v, written $\langle \tau, i, v\rangle \models_{\textsc{ltl}} \Phi$, as shown in Figure 5. If Φ is closed, we omit v and simply write $\langle \tau, i\rangle \models_{\textsc{ltl}} \Phi$. For closed formulas, we say that T *satisfies* Φ, written, with a slight abuse of notation, $T \models_{\textsc{ltl}} \Phi$, if for every run τ of T, we have that $\langle \tau, 0\rangle \models_{\textsc{ltl}} \Phi$. Given a TS T and a closed LTL-FO$_a$ property Φ, the *linear-time verification problem* amounts to checking whether $T \models_{\textsc{ltl}} \Phi$.

By appealing to [25], we show that linear-time verification of LTL-FO$_a$ properties is undecidable over bounded, generic TSs. This is in contrast with the decidability result for $\mu\mathcal{L}_a$, and $\mu\mathcal{L}$, and the folklore assumption that the ability

$$\begin{array}{ll}
\langle \tau, i, v\rangle \models_{\text{LTL}} \varphi & \text{if } \langle \mathcal{I}(\tau(i)), v\rangle \models \varphi \\
\langle \tau, i, v\rangle \models_{\text{LTL}} \neg\Phi & \text{if it is not the case that } \langle \tau, i, v\rangle \models_{\text{LTL}} \Phi \\
\langle \tau, i, v\rangle \models_{\text{LTL}} \Phi_1 \wedge \Phi_2 & \text{if } \langle \tau, i, v\rangle \models_{\text{LTL}} \Phi_1 \text{ and } \langle \tau, i, v\rangle \models_{\text{LTL}} \Phi_2 \\
\langle \tau, i, v\rangle \models_{\text{LTL}} \exists x.\text{LIVE}(x) \wedge \Phi & \text{if there exists } d \in \mathit{adom}(\mathcal{I}(\tau(i)) \\
 & \quad \text{such that } \langle \tau, i, v[x/d]\rangle \models_{\text{LTL}} \Phi \\
\langle \tau, i, v\rangle \models_{\text{LTL}} \mathbf{X}\,\Phi & \text{if } \langle \tau, i+1, v\rangle \models_{\text{LTL}} \Phi \\
\langle \tau, i, v\rangle \models_{\text{LTL}} \Phi_1 \,\mathbf{U}\, \Phi_2 & \text{if there exists } k \geq i \text{ such that } \langle \tau, k, v\rangle \models_{\text{LTL}} \Phi_2 \\
 & \quad \text{and for every } j,\ \text{if } i \leq j < k \text{ then } \langle \tau, j, v\rangle \models_{\text{LTL}} \Phi_1
\end{array}$$

Figure 5: Semantics of LTL-FO$_a$

of the μ-calculus to capture LTL in a propositional setting, lifts also to a first-order setting. This result, together with Theorem 17, shows that $\mu\mathcal{L}$ cannot capture LTL-FO$_a$, thus neither LTL-FO, and hence that $\mu\mathcal{L}$ does not have the ability to "isolate" runs of a TS.

We call a TS $T = \langle \Delta, Q, q_0, \rightarrow, \mathcal{I}\rangle$ *simple-infinite* if it is generic and serial, Δ is infinite, and for each pair of states $q_1, q_2 \in \Delta$ it is effectively decidable whether $q_1 \rightarrow q_2$ and whether $\mathcal{I}(q_1) \sim \mathcal{I}(q_2)$. Such transition systems have an infinite domain, and can be considered simple because they are generic, serial, and it is possible to effectively decide, given a state, which are its successor states, and whether two states are isomorphic.

Theorem 18. *There exists a simple-infinite TS bounded by* 1*, over which linear-time verification of* LTL-FO$_a$ *formulas is undecidable.*

Proof. The proof is by reduction from the validity of (a fragment of) LTL with freeze quantifiers over infinite data words, shown to be undecidable in [25, Theorem 5.2]. Given a finite alphabet Σ of labels, an infinite data word w over Σ and Δ is an infinite sequence of key-value pairs over $\Sigma \times \Delta$, i.e., w has the form $\langle p_0, d_0\rangle\langle p_1, d_1\rangle \cdots$, where for every i, we have that $p_i \in \Sigma$ and $d_i \in \Delta$. The logic considered in [25, Theorem 5.2] is LTL$_1^{\downarrow}(\mathbf{X}, \mathbf{U})$, which has the following syntax:

$$\phi ::= p \mid \texttt{true} \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid \mathbf{X}\,\phi \mid \phi_1 \,\mathbf{U}\, \phi_2 \mid \downarrow_1\phi \mid \uparrow_1$$

where $p \in \Sigma$. From now one, we implicitly assume that ϕ is closed, i.e., that every subformula of the form $\uparrow_1$ is in the the scope of a $\downarrow_1\phi$ formula. Intuitively, LTL$_1^{\downarrow}(\mathbf{X}, \mathbf{U})$ extends LTL with the ability to store the currently processed value into a single register (via $\downarrow_1$), and to check whether the currently processed value is equal to the one stored in that register (via $\uparrow_1$). Atomic formulae are used to predicate on the key propositions of the data word. More specifically, consider a data word $w = \langle p_0, d_0\rangle\langle p_1, d_1\rangle \cdots$, and a position i over w. The satisfaction relation for LTL$_1^{\downarrow}(\mathbf{X}, \mathbf{U})$ is defined w.r.t. a valuation v that assigns a value from Δ to the (single) register, and is defined as for standard LTL, except for the following three cases:

- The atomic formula p holds at position i of w with valuation v, if $p_i = p$;

- A "store value" formula $\downarrow_1 \phi$ holds at position i of w with valuation v, if ϕ holds at position i of w by considering a new valuation v' that assigns the register to the current value d_i;
- A "check value" formula $\uparrow_1$ holds at position i of w with valuation v, if the current value d_i is equal to the one assigned to the register by v.

For a detailed description of this logic, see [25].

We encode validity of $\text{LTL}_1^{\downarrow}(\mathbf{X}, \mathbf{U})$ over infinite data words as a linear-time verification problem of LTL-FO_a over a simple-infinite TS $T = \langle \Delta, Q, q_0, \rightarrow, \mathcal{I} \rangle$, bounded by 1. More specifically:

1. We define T in such a way that its runs exactly correspond to all infinite data words over Σ and Δ.
2. We define a translation function *FreezeToFO* that, given an $\text{LTL}_1^{\downarrow}(\mathbf{X}, \mathbf{U})$ formula ϕ, produces a corresponding LTL-FO_a formula $\Phi = \mathit{FreezeToFO}(\phi)$.
3. We recast validity of ϕ as a linear-time verification problem that employs T and Φ.

We consider the two predicates in $\mathcal{F}$: *(i)* $\mathit{Key}_p/0$ for $p \in \Sigma$, used to mirror the finitely many key propositions in Σ; *(ii)* $\mathit{Val}/1$, containing a single tuple that stores the current value of the data word.

The TS T is defined as follows. Q is the least set of states satisfying the following conditions: $q_0 \in Q$, and for every $p \in \Sigma$, and for every $d \in \Delta$, there exists a state q_p^d such that $q_p^d \in Q$, and $\mathcal{I}(q_p^d)$ is defined as follows:

- $\mathit{Key}_p^{\mathcal{I}(q_p^d)} = \{\langle\rangle\}$ (i.e., Key_p holds in $\mathcal{I}(q_p^d)$);
- $\mathit{Val}^{\mathcal{I}(q_p^d)} = \{d\}$ (i.e., the extension of Val in $\mathcal{I}(q_p^d)$) is d.

The transition relation $\rightarrow$, in turn, is defined as the least set of transitions satisfying the following conditions:

- For each state $q_p^d \in Q$ such that $q_p^d \neq q_0$, we have that $q_0 \rightarrow q_p^d$ (the initial state is connected to any other state);
- For each pair of states $q_p^d, q_{p'}^{d'} \in Q$ such that $q_p^d \neq q_0$ and $q_{p'}^{d'} \neq q_0$, we have that $q_p^d \rightarrow q_{p'}^{d'}$ (each non-initial state is connected to itself and to any other non-initial state).

It is immediate to see that T is simple-infinite and 1-bounded. It is also immediate to see that there exists a bijection between the set of infinite data words over Σ and Δ and the runs of T, where a data word $\langle p_0, d_0 \rangle \langle p_1, d_1 \rangle \cdots$ is mirrored into a run of T of the form $q_0 q_{p_0}^{d_0} q_{p_1}^{d_1} \cdots$.

Next we define the translation function *FreezeToFO*. Formulas of $\text{LTL}_1^{\downarrow}(\mathbf{X}, \mathbf{U})$ are translated into LTL-FO_a by replacing each "store value" formula with an existential first-order quantification over *Val*, and each "check value" formula by checking whether the quantified variable is in Val. In other words, the single register of $\text{LTL}_1^{\downarrow}(\mathbf{X},\mathbf{U})$ is simulated by a first-order variable. More specifically, given an $\text{LTL}_1^{\downarrow}(\mathbf{X},\mathbf{U})$ formula ϕ, $\mathit{FreezeToFO}(\phi)$ replaces: *(i)* each atomic formula p of ϕ with Key_p, *(ii)* each subformula $\downarrow_1\psi$ of ϕ with $\exists x.\, \mathit{Val}(x) \wedge \mathit{FreezeToFO}(\psi)$, and *(iii)* each atomic formula $\uparrow_1$ of ϕ with $\mathit{Val}(x)$.[8] For example, the formula $\phi_{ex} = \downarrow_1 \mathbf{X}\,\mathbf{G}(a \supset \neg\uparrow_1)$, stating that the data values assigned to the key proposition a at positions greater than one are all different from the value present in the initial position of the data word, is translated into $\mathit{FreezeToFO}(\phi_{ex}) = \exists x.\, \mathit{Val}(x) \wedge \mathbf{X}\,\mathbf{G}(\mathit{Key}_a \supset \neg\, \mathit{Val}(x))$.

Finally to recast validity as verification, we start by noticing that, from the semantics of the two logics $\text{LTL}_1^{\downarrow}(\mathbf{X},\mathbf{U})$ and LTL-FO_a, one can directly show that for every $\text{LTL}_1^{\downarrow}(\mathbf{X},\mathbf{U})$ formula ϕ, ϕ holds over an infinite data word if and only if $\mathit{FreezeToFO}(\phi)$ holds over an infinite run of T, provided that the initial state q_0 is skipped. Consequently, we obtain that ϕ is valid if and only if $T \models_{\text{LTL}} \mathbf{X}\,\mathit{FreezeToFO}(\phi)$. □

7. Bounded Situation Calculus Action Theories

In this section, we show how the results obtained in the previous sections find application in the context of *bounded situation calculus action theories.* The *situation calculus* [31, 36] is a logical language for representing and reasoning about dynamic worlds. The language has terms of three sorts, namely *objects*, *actions*, and *situations*: objects represent entities in the domain of interest, other than actions and situations; actions model events that trigger changes in the world; and situations represent world histories, i.e., sequences of actions applied in the situation resulting from previous applications. Situations are built through the function symbol *do*, with $do(a, s)$ denoting the successor situation resulting from the execution of action a in situation s. The constant S_0 denotes the initial situation where no action has been performed. We assume to have countably infinitely many object constants, on which the *unique name assumption* (UNA) is enforced. However, we do not assume domain closure for objects.

Predicates and *functions* whose value depends on the situation are called *fluents.* These are denoted by symbols that take a situation term as last argument (e.g., $Holding(x, s)$, meaning that the robot is holding object x in situation s). Without loss of generality, we assume that there are no functions other than constants and no non-fluent predicates. We denote fluents by F and the finite set of primitive fluents by $\mathcal{F}$. The arguments of fluents, apart from the last one which is of sort situation, are of sort object. Also, we consider a finite number of *action types*, each of which takes a tuple of objects as arguments.

Using the situation calculus, one can formulate action theories describing how the world changes as a result of actions. A well studied class of such theories are

[8] Notice that the formula produced by *FreezeToFO* is indeed an LTL-FO_a formula, since $\exists x.\, \mathit{Val}(x) \wedge \Psi$ implies $\exists x.\text{LIVE}(x) \wedge \Psi$.

basic action theories [36]. A basic action theory $\mathcal{D}$ is the union of the following disjoint sets of first-order (FO) and second-order (SO) axioms:

- $\mathcal{D}_0$: (FO) *initial situation description axioms* describing the initial configuration of the world (such a description may be complete or incomplete);
- $\mathcal{D}_{poss}$: (FO) *precondition axioms* of the form

  $$Poss(A(\vec{x}), s) \equiv \phi_A(\vec{x}, s),$$

  one per action type, stating the conditions $\phi_A(\vec{x}, s)$ under which an action $A(\vec{x})$ can be legally performed in situation s; these axioms use the special predicate $Poss(a, s)$, expressing that action a is executable in situation s; $\phi_A(\vec{x}, s)$ is a formula of the situation calculus that is *uniform* in situation s, that is, a formula mentioning neither any situation term but s, nor *Poss* (see [36] for a formal definition);
- $\mathcal{D}_{ssa}$: (FO) *successor state axioms* of the form

  $$F(\vec{x}, do(a, s)) \equiv \phi_F(\vec{x}, a, s),$$

  one per fluent F, describing how fluent F changes when action a is executed in situation s; the right-hand side (RHS) of the axiom, i.e., $\phi_F(\vec{x}, a, s)$ is, again, a situation calculus formula uniform in s;
- $\mathcal{D}_{ca}$: (FO) unique name axioms for actions and (FO) domain closure on action types;
- $\mathcal{D}_{uno}$: (FO) unique name axioms for object constants;
- Σ: (SO) foundational, domain independent, axioms of the situation calculus [36].

We say that a situation s is *executable*, if every action performed in reaching s is executable in the situation in which it is performed. We denote by C the set of constants explicitly mentioned in the initial situation description or in precondition or successor state axioms. For simplicity, and without loss of generality, we assume that all constants in C appear in the initial situation description. Notice that these are the constants we actually predicate on (while on the others we only predicate existence and unique name assumption).

Bounded Action Theories. A situation calculus (basic) action theory $\mathcal{D}$ is *bounded* if, for a given natural number n, at every executable situation, the number of distinct object tuples occurring in the extension of each fluent of $\mathcal{D}$ is at most n. Thus, the interpretation of a fluent at every situation does not use more than n distinct tuples, though these can change from situation to situation and collectively be infinitely many [20, 21]. For convenience, with a little abuse of notation, we say that an action theory is *bounded by b* when in each situation the number of objects occurring in the extension of all fluents is at most b. Notice that, when $\mathcal{D}$ is bounded by b, we have that a value for the bound n is given by $|\mathcal{F}| \cdot b^k$, where k is the maximal arity of fluents.

Example 4 (Avid Reader)**.** A prototypical example of boundedness is provided by a bookshelf. An agent is an avid reader and has a bookshelf of a given size. He acquires books, puts them in the bookshelf, reads them, and then puts them back in the bookshelf or gives them away. The available space in the bookshelf is given in units and each book consumes a certain number of units (e.g., one for simplicity). The reader cannot acquire a book if there is not enough space in the bookshelf.

The possible actions are the following:

- *acquire*(*book*). Pre: *book* not already in the bookshelf, space available in the bookshelf. Post: *book* in the bookshelf and one less unit available in the bookshelf.

- *read*(*book*). Pre: *book* in the bookshelf. Post: *book* in the hand of the avid reader, *book* not in the bookshelf.

- *store*(*book*). Pre: *book* in the hand of the avid reader, space available in the bookshelf. Post: *book* in the bookshelf and one less unit available in the bookshelf.

- *discard*(*book*). Pre: *book* in the hand of the avid reader. Post: *book* not in the hand of the avid reader and not in the bookshelf.

It is easy to write explicitly precondition and successor state axioms, which we omit for sake of brevity. It is also easy to see that the resulting action theory is indeed bounded. ■

Transition Systems Induced by Situation Calculus Models. When focussing on verification of temporal properties we do not need to deal directly with full action theory models, since both actions and situations (both of which do not appear explicitly in the formulas to verify) can essentially be disregarded [20, 21]. Among the various TSs, we are interested in those *induced* by models of the situation calculus action theory $\mathcal{D}$. Consider a model M of $\mathcal{D}$ with object domain Δ[9] and situation domain $\mathcal{S}$. Given a situation s, we can associate to s a first-order interpretation $\mathcal{I}_M(s) \doteq \langle \Delta, \cdot^{\mathcal{I}} \rangle$, where: *(i)* for every $c \in C$, $c^{\mathcal{I}} = c^M$ and *(ii)* for every (situation-suppressed) fluent F of $\mathcal{D}$, $F^{\mathcal{I}} = \{\vec{d} \mid \langle \vec{d}, s \rangle \in F^M\}$. Then, we can define the TS *induced by M* as the labelled TS $T_M = \langle \Delta, Q, q_0, \mathcal{I}, \rightarrow \rangle$ such that:

- $Q = \mathcal{S}$ is the set of *possible states*, each corresponding to a distinct executable situation in $\mathcal{S}$;
- $q_0 = S_0^M \in Q$ is the *initial state*, with S_0^M the initial situation of $\mathcal{D}$;
- $\rightarrow \subseteq Q \times Q$ is the *transition relation* such that $q \rightarrow q'$ iff there exists some action a such that $\langle a, q \rangle \in Poss^M$ and $q' = do^M(a, q)$.

[9] Note that Δ is infinite for the theories we are considering, since we have assumed that they include infinitely many constants with UNA.

- $\mathcal{I} : Q \mapsto Int_{\Delta}^{\mathcal{F},C}$ is the *labeling function* associating to each state (situation) q the interpretation $\mathcal{I}(q) = \mathcal{I}_M(q)$.

The TS induced by a model M is essentially the tree of executable situations, with each situation labelled by an interpretation of fluents (and constants), corresponding to the interpretation that M associates to that situation. Notice that transitions do not carry any information about the corresponding triggering action. As expected, we have that situation calculus action theories give rise to generic TSs.

Theorem 19. *For every model M of a situation calculus action theory $\mathcal{D}$, the generated TS T_M is generic.*

Proof. By construction of T_M. □

Moreover, we have that bounded situation calculus action theories give rise to TSs that are also state-bounded.

Theorem 20. *For every model M of a situation calculus action theory $\mathcal{D}$ bounded by b, the generated TS T_M is state-bounded, with each state bounded by b.*

Proof. Follows directly from the definition of action theory bounded by b. □

Verification of Bounded Action Theories. We show that given a bounded situation calculus action theory with infinite object domain, there exists a new action theory with finite object domain, that preserves p-bisimilarity between the TSs of the respective models of the theories.

Theorem 21. *Let $\mathcal{D}$ be a situation calculus action theory bounded by b, $\mathcal{D}_{uno}$ the part of $\mathcal{D}$ stating the existence, with unique name assumption, of infinitely many constants, and n the maximum among the number of variables occurring in the precondition and successor state axioms of $\mathcal{D}$. Let C' be a finite set of constants such that $C \subseteq C'$ and $|C'| \geq b+n$. Define the theory $\mathcal{D}' = (\mathcal{D} \backslash \mathcal{D}_{uno}) \cup \mathcal{D}'_{uno} \cup \mathcal{D}'_{dc}$, where:*

$$\mathcal{D}'_{uno} = \{\textstyle\bigwedge_{c,c' \in C', c, c' \ distinct} c \neq c'\}, \quad \mathcal{D}'_{dc} = \{\forall x. \textstyle\bigvee_{c \in C'} x = c\}.$$

Then, for every model M of $\mathcal{D}$, there is a model M' of $\mathcal{D}'$, such that $T_M \approx^p T_{M'}$. Similarly, for every model M' of $\mathcal{D}'$ there is a model M of $\mathcal{D}$, such that $T_M \approx^p T_{M'}$.

Proof. Let M be a model of $\mathcal{D}$ with (infinite) domain Δ. The model M' can be obtained by fixing a (finite) domain $\Delta' \subset \Delta$, with cardinality $|C'|$, that includes the interpretation of the constants in C, and taking the interpretation of the initial situation so that $\tilde{\mathcal{I}}_M(S_0) = \tilde{\mathcal{I}}_{M'}(S_0)$. Observe that once the interpretation of the initial situation is fixed, M' is fully determined by $\mathcal{D}'$. Then, consider T_M and $T_{M'}$ and build the relation $R = \{\langle q, h, q' \rangle \mid \tilde{\mathcal{I}}(q) \sim_h \tilde{\mathcal{I}}'(q')\}$. The proof consists in showing that R is a p-bisimulation such that $\langle q_0, h_0, q'_0 \rangle \in R$, for h_0 the identity on $adom(\mathcal{I}(q_0))$. To this end, let $\langle q, h, r \rangle \in R$ and consider the definition of p-bisimulation.

Obviously, requirement 1 of the definition is trivially satisfied, by the definition of R. As to requirements 2 and 3, we will use the following known results:

1. Any possibly open situation calculus first-order formula $\phi(\vec{x}, s)$ with variables $\vec{x}$ and s of object and situation sort, respectively, can be rewritten as a formula ϕ' where action terms do not occur, such that ϕ and ϕ' are equivalent with respect to $\mathcal{D}_{ca}$ [21, Theorem 5].

2. Evaluating a possibly open situation calculus formula ϕ uniform in s against a model M and a situation s is equivalent to evaluating the situation-suppressed version of ϕ against $\mathcal{I}_M(s)$, under the same assignment to free variables [21, Theorem 7].

3. Given a model M of a bounded action theory, an executable situation s, and a ground action $a = A^M(\vec{o})$ (of type $A(\vec{y})$), for every fluent F, there exists a situation-suppressed action-term-free formula $\phi = \phi(\vec{x}, \vec{y})$, such that $\langle \vec{p}, do^M(a, s)\rangle \in F^M$ iff $\mathcal{I}_M(s), v \models \phi(\vec{x}, \vec{y})$, with $v(\vec{x}) = \vec{p}$ and $v(\vec{x}) = \vec{o}$. In words, the interpretation of a fluent F in M, after the execution of action a at situation s, can be obtained as the answer to a suitable query $\phi(\vec{x}, \vec{y})$ over $\mathcal{I}_M(s)$, with $\vec{y}$ assigned to $\vec{o}$.

Let $q' \in Q$ be such that $q \to q'$. By the definition of induced TS, it follows that there exists a (ground) action a, say $a = A^M(\vec{o})$, such that $\langle a, q\rangle \in \mathit{Poss}^M$ and $q' = do^M(a, q)$. To prove requirement 2a of the definition of bisimulation, we next show that there exists also an action a' such that $\langle a', r\rangle \in \mathit{Poss}^{M'}$. This, by definition of induced TS, implies that there exists a state $r' \in Q'$, namely the situation $r' = do^{M'}(a', r)$, such that $r \to' r'$.

Recall that precondition axioms have the form $\mathit{Poss}(A(\vec{x}), s) \equiv \phi_A(\vec{x}, s)$, where we can assume, by result 1 above, ϕ_A not containing action terms (if not, ϕ_A can be rewritten). Because $\langle a, q\rangle \in \mathit{Poss}^M$, we have that $M, v \models \phi_A(\vec{x}, s)$, for v such that $v(s) = q$ and $v(\vec{x}) = \vec{o}$. Consequently, by result 2, $\mathcal{I}_M(q), v \models \phi_A(\vec{x})$, for $\phi_A(\vec{x})$ the situation-suppressed version of $\phi_A(\vec{x}, s)$. Then, since $|\Delta| \geq |\mathit{adom}(\mathcal{I}_M(q))| + |\mathit{vars}(\phi)|$ (as Δ is infinite and $\mathcal{D}$ bounded by b), by Lemma 13, we have that $\tilde{\mathcal{I}}_M^{D_v}(q), v \models \phi'_A$, for $D_v = \textsc{img}(v|_{\mathit{free}(\phi_A)})$ and ϕ'_A the domain-independent version of ϕ_A. Now, observe that the definition of R implies that, for some h, $\tilde{\mathcal{I}}(q) \sim_h \tilde{\mathcal{I}}'(r)$, thus, since $\mathcal{I}(q) = \mathcal{I}_M(q)$, it follows that $\tilde{\mathcal{I}}_M(q) \sim_h \tilde{\mathcal{I}}'(r)$. Consider the valuation $v' = \hat{h} \circ v$, for $\hat{h}$ any bijection extending h to $\vec{o}$ (notice that only the values assigned to $\vec{x}$ are relevant, thus v' can be undefined on the other variables). Such an $\hat{h}$ exists by the boundedness of $\mathcal{D}$ and the cardinality constraint on Δ'. Also, it can be easily seen that $\tilde{\mathcal{I}}_M^{D_v}(q) \sim_{\hat{h}} \tilde{\mathcal{I}}'^{D_{v'}}(r)$, for $D_{v'} = \textsc{img}(v'|_{\mathit{free}(\phi_A)})$. Thus, by the invariance of first-order logic under isomorphic interpretations, we have that $\tilde{\mathcal{I}}'^{D_{v'}}(r), v' \models \phi'_A$.

Now, since by definition of induced TS we have $\mathcal{I}'(r) = \mathcal{I}_{M'}(r)$, which implies $\tilde{\mathcal{I}}'(r) = \tilde{\mathcal{I}}_{M'}(r)$, it follows that $\tilde{\mathcal{I}}_{M'}^{D_{v'}}(r), v' \models \phi'_A$. Finally, observe that, by $\mathcal{D}'_{uno}$ and $\mathcal{D}'_{dc}$, $|\Delta'| = |C'| \geq b + n$, and, because $\tilde{\mathcal{I}}'(r) \sim \tilde{\mathcal{I}}(q)$ and $\mathcal{D}$ is bounded by b, $|\mathit{adom}(\mathcal{I}_{M'}(r))| \leq b$. Thus, by Lemma 13, $\tilde{\mathcal{I}}_{M'}^{D_{v'}}(r), v' \models \phi'_A$ implies that

$\mathcal{I}_{M'}(r), v' \models \phi'_A$ which, by result 2, implies that $M', w \models \phi_A(\vec{x}, s)$, for w the extension of v' to s such that $w(s) = r$. Therefore, $a' = A^{M'}(\vec{p})$, with $\vec{p} \doteq w(\vec{x})$, is an action such that $\langle a', r\rangle \in Poss^{M'}$.

For requirement 2b, recall that successor-state axioms have the form $F(\vec{y}, do(act, s)) \equiv \phi_F(\vec{y}, act, s)$, with ϕ_F uniform in s. Thus, for act assigned to $a = A^M(\vec{o})$ as above, we have that $\phi_F(\vec{y}, act, s)$ is equivalent to $\phi_F(\vec{y}, A(\vec{x}), s)$. This, by result 1, can be rewritten as an action-term-free formula $\phi_{FA}(\vec{y}, \vec{x}, s)$, hence the successor-state axioms can be rewritten as $F(\vec{y}, do(A(\vec{x}), s)) \equiv \phi_{FA}(\vec{y}, \vec{x}, s)$. Hence, by result 2, the interpretation $\mathcal{I}_M(q')$ can be obtained by evaluating, for every fluent F, the situation-suppressed version of $\phi_{FA}(\vec{y}, \vec{x}, s)$, i.e., $\phi_{FA}(\vec{y}, \vec{x})$, against the (FO) interpretation $\mathcal{I}_M(q)$, with $\vec{x}$ assigned to $\vec{o}$. Then, because $\mathcal{I}(q) = \mathcal{I}_M(q)$ and $\mathcal{I}(q') = \mathcal{I}_M(q')$, we can obtain $\mathcal{I}(q')$ by evaluating each $\phi_{FA}(\vec{y}, \vec{x})$ against $\mathcal{I}(q)$, with $\vec{x}$ assigned to $\vec{o}$. By an analogous argument, for $a' = A^{M'}(\vec{p})$ assigned to act, $\mathcal{I}'(r')$ can be otained by evaluating each $\phi_{FA}(\vec{y}, \vec{x})$, against $\mathcal{I}'(r)$, with $\vec{x}$ assigned to $\vec{p}$.

Let $\phi'_{FA}(\vec{y}, \vec{x})$ be the domain-independent version of $\phi_{FA}(\vec{y}, \vec{x})$. By Lemma 13, for every assignment u, we have that $\mathcal{I}(q), u \models \phi_{FA}(\vec{y}, \vec{x})$ iff $\tilde{\mathcal{I}}^{D_u}(q), u \models \phi'_{FA}(\vec{y}, \vec{x})$, for $D_u = \textsc{img}(u|_{free(\phi_{FA})})$. Now, observe that, since $\mathcal{D}$ is bounded, so is $adom(\mathcal{I}(q'))$. This, together with the fact that Δ is infinite, can be shown to imply that $adom(\mathcal{I}(q'))$ contains only objects from $adom(\mathcal{I}(q)) \cup \vec{o}$ (otherwise $adom(\mathcal{I}(q'))$ would be infinite, see [21, Theorem 10]). The same holds for $adom(\mathcal{I}'(r'))$, which contains only objects from $adom(\mathcal{I}'(r)) \cup \vec{p}$. Therefore, by fixing a bijection $\hat{h}$ between $adom(\mathcal{I}(q)) \cup \vec{o}$ and $adom(\mathcal{I}(r)) \cup \vec{p}$ (this exists by the boundedness of $\mathcal{D}$ and the cardinality constraint on Δ'), since $\tilde{\mathcal{I}}(q) \sim_h \tilde{\mathcal{I}}'(r)$ and by the invariance of first-order with respect to isomorphism, we have that, for every assignment u, $\tilde{\mathcal{I}}^{D_u}(q), u \models \phi'_{FA}(\vec{y}, \vec{x})$ iff $\tilde{\mathcal{I}'}^{D_{u'}}(r), u' \models \phi'_{FA}(\vec{y}, \vec{x})$, for $u' = \hat{h} \circ u$ and $D_{u'} = \textsc{img}(u'|_{free(\phi_{FA})})$. But then, we have that $\tilde{\mathcal{I}}(q') \sim_{\hat{h}|_{adom(\mathcal{I}(q'))}} \tilde{\mathcal{I}}'(r')$, therefore $\langle q', \hat{h}|_{adom(\mathcal{I}(q'))}, r'\rangle \in R$. The proof for requirement 3 follows a similar argument.

Finally, given a model M' of $\mathcal{D}'$, the corresponding M can be obtained in the same way as above. Also the fact that $T_M \approx^p T_{M'}$ can be shown in the same way. □

As expected, TSs induced by models of finite-state action theories can be made finite.

Theorem 22. *Let $\mathcal{D}'$ be a (bounded) situation calculus action theory defined as in Theorem 21, for some finite C'. Then, for every model M' of $\mathcal{D}'$ with (finite) object domain Δ', the corresponding induced TS $T_{M'}$ is p-bisimilar to a TS T_F that is generic, finite-state, and effectively computable from $\mathcal{D}'$, $\tilde{\mathcal{I}}_{M'}(S_0)$, and Δ'.*

Proof. We prove the result by providing an algorithm to compute $T_F = \langle \Delta_F, Q_F, q_{F0}, \rightarrow_F, \mathcal{I}_F\rangle$. We set $\Delta_F = \Delta'$, and $\mathcal{I}_F$ as the identity function, and we initialize $q_{F_0} = \mathcal{I}_{M'}(S_0)$, $Q_F = \{q_{F0}\}$, and $\rightarrow_F = \emptyset$. Then, starting with $q = q_{F0}$, we consider all actions a that, in M', are executable in those situations

s such that $\mathcal{I}_{M'}(s) = q$. This requires evaluating only the (situation-suppressed) precondition axiom of a against $\mathcal{I}(q)$. Notice that since Δ_F is finite, there are only finitely many actions. For every a, we then compute the interpretation of situation $s' = do^{M'}(a, s)$, for s as above. To this end, it is enough to evaluate the (situation-suppressed) right-hand side of each successor-state axiom against q (i.e., $\mathcal{I}_{M'}(s)$, for s as above), with the action assigned to a, thus producing a new interpretation $q' = \mathcal{I}_{M'}(s')$. Observe that the finiteness of Δ_F guarantees that both precondition and successor-state axioms can be effectively evaluated. Then, if not already present, we add the obtained q' to Q_F, and let $q \to_F q'$. Finally, we iterate these steps on the newly added states, until no new states are added. Termination is an obvious consequence of Δ_F's finiteness, which implies that only finitely many states can be a added to Q_F. Genericity is a consequence of the fact that the interpretation of states is obtained by answering first-order queries, which are unable to distinguish objects outside the active domain. □

With these results in place next prove that if we are given a model M of $\mathcal{D}$, then checking whether $T_M \models \Phi$ is decidable. That is, we have decidability in case of complete information on the initial situation. Furthermore, we can extend this result to deal with verification in presence of incomplete information. We write $\mathcal{D} \models \Phi$ if $T_M \models \Phi$, for every model M of $\mathcal{D}$.

Theorem 23. *Let $\mathcal{D}$ be a situation calculus bounded action theory (with infinite object domain) and Φ a closed $\mu\mathcal{L}$ formula with all variables renamed apart and belonging to a finite set $Vars$. Then, it is decidable to check wether $\mathcal{D} \models \Phi$.*

Proof. Given $\mathcal{D}$, let $\mathcal{D}'$ be an action theory as in Theorem 21, with $|C'| = 2b+m$, for m the maximum between $|Vars|$ and the maximum number of variables occurring in the action precondition and successor-state axioms of $\mathcal{D}$ (n of Theorem 21). By Theorem 21, every model M of $\mathcal{D}$ with infinite object domain Δ, has a corresponding p-bisimilar model M' of $\mathcal{D}'$ with finite object domain Δ' of size $|C'|$, and viceversa. Thus, by Theorem 12, for corresponding M and M', we have that $T_M \models \Phi$ iff $T_{M'} \models \Phi$. Hence, since by Theorem 21, the models of $\mathcal{D}'$ "cover" those of $\mathcal{D}$ and viceversa, it follows that $\mathcal{D} \models \Phi$ iff $\mathcal{D}' \models \Phi$. Finally, decidability is easily obtained by observing that the models M' of $\mathcal{D}'$ are finitely many, up to object renaming, and that by Theorem 22, checking whether $T_{M'} \models \Phi$ is decidable. □

Finally, by exploiting the undecidability result on first-order LTL (Theorem 18), we show that verification for LTL-FO$_a$ over bounded situation calclulus action theories is also undecidable. Following the notation above, given a situation calculus action theory $\mathcal{D}$ and an LTL-FO$_a$ formula Φ, we define $\mathcal{D} \models_{\text{LTL}} \Phi$ if $T_M \models_{\text{LTL}} \Phi$ for every model M of $\mathcal{D}$. Given $\mathcal{D}$ and Φ, the linear-time verification problem amounts to check whether $\mathcal{D} \models_{\text{LTL}} \Phi$.

Theorem 24. *There exists a situation calculus bounded action theory (with infinite object domain) $\mathcal{D}$ for which the linear-time verification problem against* LTL-FO$_a$ *formulas is undecidable.*

Proof. We construct a situation calculus bounded action theory (with infinite object domain Δ) $\mathcal{D}$ using as fluents the predicates introduced in the proof of Theorem 18. In particular, given a finite set Σ of key propositions, we employ the following situation-suppressed fluents: $Key_p/0$ for $p \in \Sigma$, and $Val/1$. In particular, $\mathcal{D}$ contains $|\Sigma|$ 1-parameter actions $GuessVal_{p_i}$, one per key proposition in Σ. Each such actions is always executable, makes its corresponding key proposition true, and guesses the next value. Technically, for every $i \in \{1, \ldots, |\Sigma|\}$, the (extremely simple) successor state axioms are:

- $Key_{p_i}(do(a, s)) \equiv (a = GuessVal_{p_i}(x))$ – for the fluents Key_{p_i},
- $Val(x, do(a, s)) \equiv (a = GuessVal_{p_i}(x))$ – for the fluent Val.

It is easy to see that the runs produced by $\mathcal{D}$ closely match with those present in the TS T used in the proof of Theorem 18. In particular, there exists a model M' for $\mathcal{D}$ for which $T_{M'} = T$. For such choice, from Theorem 18 we obtain immediately that linear-time verification is undecidable over $T_{M'}$. Since $\mathcal{D} \models_{\text{LTL}} \Phi$ if $T_M \models_{\text{LTL}} \Phi$ for every possible model M (including M'), also checking whether $\mathcal{D} \models_{\text{LTL}} \Phi$ is undecidable in general. □

8. Conclusions

In this paper we have studied first-order μ-calculus with quantification across states, in the three main variants proposed in literature. We have seen that the three corresponding notions of bisimulation collapse for the class of generic transition systems, which includes all transition systems generated by formalisms for reasoning about actions based on first-order representation of states, and logical mechanisms to generate the successor state from the current, in particular the situation calculus. From this, we have derived decidability of verification for $\mu\mathcal{L}$ over state-bounded transition systems and over bounded situation calculus action theories. These results contrast with verification for first-order LTL, which is instead undecidable.

This work opens several research avenues. An important extension consists in considering object domains with embedded *predefined types*, such as naturals or rationals with controlled operators and predicates [29]. In these cases, the difficulty arises from the fact that predicates can be infinite but must be considered part of the state, which yields unbounded states and calls for some form of quantifier elimination on objects of the embedded types. Related work dealing with such predicates include [35], in the context of situation calculus action theories, and [12, 19], in the context of data-centric services.

Also, we observe that our decidability result for $\mu\mathcal{L}$ verification on bounded transition systems relies on a notion of abstraction [15, 7]. In particular, our abstraction is *faithful* for all formulas with a given number of object variables. For future work, it is also of interest to consider forms of abstraction that are weaker, i.e., that are faithful with respect to specific formulas only, or even abstraction that are only sound or only complete, in order to decrease the size of the abstract transition system.

Finally, we mention that we are relying on the techniques presented in this work as the foundational basis for the development of actual model checking tools for data-aware processes. In particular, we are exploiting relational technology to compute faithful, finite abstractions for generic, state-bounded data-aware processes. Preliminary results are discussed in [14]. Interestingly, such abstraction techniques do not only apply to bounded situation calculus action theories, but also to a number of other formal models for data-aware processes, such as [3, 2, 9, 33].

Acknowledgments. We thank several colleagues for insightful discussions on the issues of this paper, including Babak Bageri Hariri, Giovanna D'Agostino, Riccardo De Masellis, Alin Deutsch, Paolo Felli, Giacomo Lenzi, Yves Lesperance, Alessio Lomuscio, Keishi Okamoto, and Moshe Y. Vardi.

References

[1] Abiteboul, S., Hull, R., Vianu, V., 1995. Foundations of Databases. Addison Wesley Publ. Co.

[2] Bagheri Hariri, B., Calvanese, D., De Giacomo, G., De Masellis, R., Felli, P., Montali, M., 2013. Description logic knowledge and action bases. J. of Artificial Intelligence Research 46, 651–686.
URL `http://www.jair.org/papers/paper3826.html`

[3] Bagheri Hariri, B., Calvanese, D., De Giacomo, G., Deutsch, A., Montali, M., 2013. Verification of relational data-centric dynamic systems with external services. In: Proc. of the 32nd ACM SIGACT SIGMOD SIGAI Symp. on Principles of Database Systems (PODS). pp. 163–174, extended version available at `http://arxiv.org/abs/1203.0024`.

[4] Bagheri Hariri, B., Calvanese, D., Deutsch, A., Montali, M., 2014. State-boundedness in data-aware dynamic systems. In: Proc. of the 14th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR). AAAI Press.

[5] Bagheri Hariri, B., Calvanese, D., Montali, M., De Giacomo, G., De Masellis, R., Felli, P., 2013. Description logic Knowledge and Action Bases. J. of Artificial Intelligence Research 46, 651–686.

[6] Baier, C., Katoen, J.-P., 2008. Principles of Model Checking. The MIT Press.

[7] Belardinelli, F., Lomuscio, A., Patrizi, F., 2011. A computationally-grounded semantics for artifact-centric systems and abstraction results. In: Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI'11). pp. 738–743.

[8] Belardinelli, F., Lomuscio, A., Patrizi, F., 2012. An abstraction technique for the verification of artifact-centric systems. In: Proc. of the 13th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR). pp. 319–328.

[9] Belardinelli, F., Lomuscio, A., Patrizi, F., 2014. Verification of agent-based artifact systems. J. of Artificial Intelligence Research 51, 333–376.

[10] Bhattacharya, K., Caswell, N. S., Kumaran, S., Nigam, A., Wu, F. Y., 2007. Artifact-centered operational modeling: Lessons from customer engagements. IBM Systems J. 46 (4), 703–721.

[11] Bradfield, J., Stirling, C., 2007. Modal mu-calculi. In: Handbook of Modal Logic. Vol. 3. Elsevier, pp. 721–756.

[12] Calvanese, D., De Giacomo, G., Hull, R., Su, J., 2009. Artifact-centric workflow dominance. In: Proceedings of the 7th International Joint Conference on Service-Oriented Computing (ICSOC'09). pp. 130–143.

[13] Calvanese, D., De Giacomo, G., Montali, M., Patrizi, F., 2016. On first-order μ-calculus over situation calculus action theories. In: Proc. of the 15th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR). AAAI Press, pp. 411–420.

[14] Calvanese, D., Montali, M., Patrizi, F., Rivkin, A., 2015. Implementing data-centric dynamic systems over a relational DBMS. In: Calì, A., Vidal, M.-E. (Eds.), Proc. of the 9th Alberto Mendelzon Int. Workshop on Foundations of Data Management (AMW). Vol. 1378. CEUR Workshop Proceedings, `http://ceur-ws.org/`.

[15] Clarke, E. M., Grumberg, O., Jha, S., Lu, Y., Veith, H., 2003. Counterexample-guided abstraction refinement for symbolic model checking. J. ACM 50 (5), 752–794.

[16] Clarke, E. M., Grumberg, O., Peled, D. A., 1999. Model Checking. The MIT Press, Cambridge, MA, USA.

[17] Claßen, J., Lakemeyer, G., 2008. A logic for non-terminating Golog programs. In: Proc. of the 11th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR). pp. 589–599.

[18] Cohn, D., Hull, R., 2009. Business artifacts: A data-centric approach to modeling business operations and processes. Bull. of the IEEE Computer Society Technical Committee on Data Engineering 32 (3), 3–9.

[19] Damaggio, E., Deutsch, A., Vianu, V., 2012. Artifact systems with data dependencies and arithmetic. ACM Trans. Database Syst. 37 (3), 22.

[20] De Giacomo, G., Lesperance, Y., Patrizi, F., 2012. Bounded Situation Calculus action theories and decidable verification. In: Proc. of the 13th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR). pp. 467–477.

[21] De Giacomo, G., Lesperance, Y., Patrizi, F., 2016. Bounded situation calculus action theories. Artificial Intelligence 237, 172–203.

[22] De Giacomo, G., Lesperance, Y., Patrizi, F., Sardina, S., 2016. Verifying ConGolog programs on bounded situation calculus theories. In: Proc. of the 30th AAAI Conf. on Artificial Intelligence (AAAI). pp. 950–956.

[23] De Giacomo, G., Lespérance, Y., Patrizi, F., Vassos, S., 2014. LTL verification of online executions with sensing in bounded situation calculus. In: Proc. of the 21st Eur. Conf. on Artificial Intelligence (ECAI). pp. 369–374.

[24] De Giacomo, G., Ternovskaia, E., Reiter, R., 1997. Non-terminating processes in the situation calculus. In: Proc. of the AAAI 1997 Workshop on Robots, Softbots, Immobots: Theories of Action, Planning and Control. pp. 18–28.

[25] Demri, S., Lazic, R., 2009. LTL with the freeze quantifier and register automata. ACM Trans. Comput. Log. 10 (3).

[26] Deutsch, A., Hull, R., Patrizi, F., Vianu, V., 2009. Automatic verification of data-centric business processes. In: Proc. of the 12th Int. Conf. on Database Theory (ICDT). pp. 252–267.

[27] Emerson, E. A., 1996. Model checking and the mu-calculus. In: Descriptive Complexity and Finite Models. AMS, DIMACS, pp. 185–214.

[28] Enderton, H. B., 2001. A Mathematical Introduction to Logic, 2nd Edition. Academic Press.

[29] Kanellakis, P. C., Kuper, G. M., Revesz, P. Z., 1995. Constraint query languages. J. Comput. Syst. Sci. 51 (1), 26–52.

[30] Libkin, L., 2007. Embedded finite models and constraint databases. In: Finite Model Theory and Its Applications. Springer, pp. 257–338.

[31] McCarthy, J., Hayes, P. J., 1969. Some philosophical problems from the standpoint of artificial intelligence. Machine Intelligence 4, 463–502.

[32] Milner, R., 1971. An algebraic definition of simulation between programs. In: Proc. of the 2nd Int. Joint Conf. on Artificial Intelligence (IJCAI). pp. 481–489.

[33] Montali, M., Rivkin, A., 2016. Model checking Petri nets with names using data-centric dynamic systems. Formal Aspects of Computing.

[34] Okamoto, K., 2010. Comparing expressiveness of first-order modal μ-calculus and first-order CTL*. RIMS Kokyuroku 1708, 1–14.

[35] Patrizi, F., Vassos, S., 2014. Action theories over generalized databases with equality constraints. In: Proceedings of the 14th European Conference on Logics in Artificial Intelligence (JELIA'16). pp. 472–485.

[36] Reiter, R., 2001. Knowledge in Action. Logical Foundations for Specifying and Implementing Dynamical Systems. The MIT Press.

[37] Stirling, C., 2001. Modal and Temporal Properties of Processes. Springer.

[38] Tarski, A., 1955. A lattice-theoretical fixpoint theorem and its applications. Pacific Journal of Mathematics 5 (2), 285–309.

[39] van Benthem, J., 1983. Modal Logic and Classical Logic. Bibliopolis, Napoli.

[40] Zarrieß, B., Claßen, J., 2014. Verifying CTL* properties of Golog programs over local-effect actions. In: Proc. of the 21st Eur. Conf. on Artificial Intelligence (ECAI). pp. 939–944.