History-Dependent Nominal μ-Calculus

Clovis Eberhart*
National Institute of Informatics
Tokyo, Japan

Bartek Klin†
University of Warsaw

***Abstract*—The μ-calculus with atoms, or nominal μ-calculus, is a temporal logic for reasoning about transition systems that operate on data atoms coming from an infinite domain and comparable only for equality. It is, however, not expressive enough to define some properties that are of interest from the perspective of system verification. To rectify this, we extend the calculus with tests for atom freshness with respect to the global history of transitions. Since global histories can grow arbitrarily large, it is not clear whether model checking for the extended calculus is decidable. We prove that it is, by showing that one can restrict attention only to locally relevant parts of the history.**

I. Introduction

Formal verification of infinite-state systems is a rich and active field of study. Sources of infinity are numerous and diverse: one may consider systems with unbounded counters, stacks or FIFO channels, timed systems with unbounded clocks, systems whose states are generated by regular or push-down processes, and so on. In each of these settings, formal verification relies on methods for symbolic representation of infinite models by finite means. One may then decide whether systems thus represented satisfy desirable properties, in the process of *model checking*. The properties involved are often expressed in suitable versions of temporal logics such as the modal μ-calculus [1] or its fragments CTL* [2] and LTL.

In this paper we focus on systems where the source of infinity is an infinite domain of data values that can only be compared for equality. In practical systems such data values may appear, e.g., as randomly generated passwords, unique process identifiers in distributed algorithms, or nonces in cryptographic protocols. A typical property that one might want to verify about a system with access to a black-box password generator is: *a computation path where generated passwords are all different will never reach a bad state*.

Several approaches to defining and checking properties of models equipped with data values have been considered. For example, in [3, 4, 5, 6] the linear time temporal logic LTL was extended with a freeze quantifier that can store a currently observed data value for future reference. This can be used to detect repetitions of data values, a goal followed also in [7, 8]. Another idea is to extend a temporal logic with the ability to express constraints about locally observed data values, as in [9, 10]. Temporal logics are closely connected to automata, and extending them with data values amounts to considering so-called register automata. In particular, the modal μ-calculus corresponds to alternating tree automata, and a register version of those were studied in [11, 12].

In all these works, the authors take care to ensure that their logics have a decidable satisfiability problem (*does a given formula have a model?*). To this end, they impose subtle conditions on their formalisms and inevitably limit their expressivity to some degree. If one gives up on decidable satisfiability, the data-equipped logic can get much more expressive. An example is the first-order μ-calculus of [13, 14], which features constraints over data values from arbitrary infinite domains. There even the model-checking problem (*does a given model satisfy a given formula?*) becomes undecidable, and incomplete proof procedures are the main focus.

This paper is a continuation of [15], where an *atomic μ-calculus* was proposed. Its design relies on the framework of *sets with atoms*, also known as *nominal sets* [16, 17], where data atoms from an infinite set $\mathbb{A}$, with their inherent symmetry, are built already into set-theoretic foundations. Originally intended as an algebraic approach to name binding, sets with atoms have found applications in language and computation theory, with notions such as nominal automata [18] and Turing machines with atoms [19] providing insights into properties of computing devices over infinite alphabets. A key feature of this approach is a relaxed notion of finiteness, called orbit-finiteness. Orbit-finite structures, although usually infinite, can be symbolically presented by finite means and are amenable to algorithmic manipulation.

The atomic μ-calculus of [15] extends the classical calculus with orbit-finite (instead of just finite) disjunctions and conjunctions. It can express properties of Kripke models (alternatively, labeled transition systems) equipped with data values, with formulas such as

$$\bigvee_{a \in \mathbb{A}} (a \wedge \Diamond \mu X.(a \vee \Diamond X))$$

meaning: *some data atom that appears in the present state, appears again on some future path*. Its satisfiability problem is undecidable, but model checking on orbit-finite models remains decidable, so the calculus is potentially applicable to the verification of systems that manipulate data atoms.

However, the calculus of [15] is not expressive enough to be useful in practice. In fact, one of the main results of that paper was the non-obvious fact that the calculus cannot express

[*] Supported by ERATO HASUO Metamathematics for Systems Design Project (No. JPMJER1603), JST.

[†] Supported by the European Research Council (ERC) under the European Unions Horizon 2020 research and innovation programme (ERC consolidator grant LIPA, agreement no. 683080).

a property called #PATH: *there exists an infinite path where no data atom appears more than once*. As a consequence, many practically motivated properties such as the one regarding password generators mentioned above, are also undefinable. #PATH is decidable on orbit-finite models, which makes its undefinability all the more disappointing. It is easy to define it in an extension of CTL* with atoms (which is *not* a fragment of the μ-calculus with atoms), but model checking of that logic is undecidable. The challenge of finding a syntactically economic extension of the μ-calculus with atoms that could express #PATH while retaining the decidability of model checking, was the main open problem left in [15].

Our main contribution is a solution to that problem. We extend the μ-calculus from [15] with simple tests of the form $\sharp p$ (read "fresh p"), saying that a basic predicate p has never held before on the computation path that led to the present state. In a sense, $\sharp$ is an extremely limited past-time modality. Similar freshness conditions have been considered before for automata over finite words [20]. With this extension, #PATH is easily definable, as well as many other properties of potential practical interest. As examples, we present a simple system with a password-protected critical section, and the cryptographic Needham–Schroeder protocol [21] that relies on generating unique nonces.

In the extended calculus, formulas are interpreted in the context of global histories, i.e., sets of basic predicates that have held in the past. Such histories can grow arbitrarily large, and so it becomes unclear whether model checking for the extended calculus remains decidable. Our main technical result (Theorem 10) is that it does. The proof relies on the observation that, since the behavior of each state in an orbit-finite system only depends on a fixed set of atoms, only a limited part of the global history is relevant at any given state and the rest of the history may be conveniently forgotten and later reinvented as needed.

The structure of the paper is as follows. In Sec. II, we recall the classical modal μ-calculus, followed by an introduction to sets with atoms in Sec. III. In Sec. IV, we recall the basics of the calculus from [15], and in Sec. V we extend it to the history-dependent μ-calculus. After showing two examples in Sec. VI, in Sec. VII we prove that model checking is decidable. We conclude in Sec. VIII with a brief discussion of the most pressing directions for future work.

II. MODAL μ-CALCULUS

We shall only need basic notions of the syntax and semantics of the modal μ-calculus; any of the texts [22, 23, 24, 25, 26] contains this and much more material. Everything in this section is completely standard, and we recall it only to fix the notation and to prepare the ground for further sections.

Fix a set $\mathbb{B}$ of *basic predicates* and a set $\mathbb{X}$ of *variables*. The set of μ-calculus formulas is generated by the grammar

$$\varphi ::= \top \mid p \mid \neg\varphi \mid \varphi \vee \varphi \mid \Diamond\varphi \mid X \mid \mu X.\varphi \tag{1}$$

where p ranges over $\mathbb{B}$ and X over $\mathbb{X}$. The set $\mathsf{FV}(\varphi) \subseteq \mathbb{X}$ of *free variables* in a formula φ is defined in the usual way, with the constructor $\mu X.\varphi$ binding X in φ. A formula is *closed* if it does not have free variables.

It is standard to write, as syntactic sugar:

$$\bot = \neg\top, \qquad \varphi \wedge \psi = \neg(\neg\varphi \vee \neg\psi), \qquad \varphi \to \psi = \neg\varphi \vee \psi,$$
$$\Box\varphi = \neg \Diamond (\neg\varphi), \qquad \nu X.\varphi = \neg\mu X.\neg\varphi[X \mapsto \neg X].$$

A formula is considered legal only if it is positive, i.e., if in every subformula $\mu X.\varphi$ the variable X occurs freely in φ only under an even number of nested negations.

Formulas are interpreted in *Kripke models* of the form

$$\mathcal{K} = (K, \longrightarrow, \vDash)$$

where K is a set of *states*, $\longrightarrow \subseteq K \times K$ is the *transition relation* and $\vDash \subseteq K \times \mathbb{B}$ is the *satisfaction relation* for basic predicates. For a state $x \in K$, we denote

$$\mathrm{pred}(x) = \{p \in \mathbb{B} \mid x \vDash p\}.$$

A model is *finite* if it has finitely many states and if $\mathrm{pred}(x)$ is finite for every state x.

The semantics of a formula φ in a model $\mathcal{K}$ is defined for a *context* ξ, which is a finite partial function

$$\xi : \mathbb{X} \rightharpoonup \mathcal{P}K$$

which is defined at least on all the free variables in φ. Then, for any formula φ, the set

$$[\![\varphi]\!]_\xi \subseteq K$$

is defined inductively:

$$\begin{array}{rcl}
x \in [\![\top]\!]_\xi & & \text{for all } x \in K \\
x \in [\![p]\!]_\xi & \iff & p \in \mathrm{pred}(x) \\
x \in [\![\neg\varphi]\!]_\xi & \iff & x \notin [\![\varphi]\!]_\xi \\
x \in [\![\varphi \vee \psi]\!]_\xi & \iff & x \in [\![\varphi]\!]_\xi \text{ or } x \in [\![\psi]\!]_\xi \\
x \in [\![\Diamond\varphi]\!]_\xi & \iff & \exists y \in [\![\varphi]\!]_\xi \text{ s.t. } x \longrightarrow y \\
x \in [\![X]\!]_\xi & \iff & x \in \xi(X) \\
[\![\mu X.\varphi]\!]_\xi & = & \mathrm{lfp}\left(\Psi \mapsto [\![\varphi]\!]_{\xi[X\mapsto\Psi]}\right).
\end{array} \tag{2}$$

In the last clause above, the least fixpoint is taken for a function on the complete lattice of subsets of K ordered by set inclusion. Thanks to the positivity assumption on φ, this function is monotone, and therefore by Tarski's fixpoint theorem it does have a least fixpoint. It can be approximated from below by a transfinite sequence

$$\Psi^0 \subseteq \Psi^1 \subseteq \Psi^2 \subseteq \cdots \subseteq \Psi^\omega \subseteq \Psi^{\omega+1} \subseteq \cdots \subseteq K \tag{3}$$

defined by

$$\begin{array}{rl}
\Psi^{\alpha+1} = [\![\varphi]\!]_{\xi[X\mapsto\Psi^\alpha]} & \\
\Psi^\beta = \displaystyle\bigcup_{\alpha<\beta} \Psi^\alpha & \quad \text{for a limit ordinal } \beta
\end{array}$$

(in particular $\Psi^0 = \varnothing$). For any $\mathcal{K}$ this sequence eventually stabilizes at the desired least fixpoint.

Example 1. Let $\mathbb{B} = \{m, d\}$, where m stands for "mistake", and d for "disaster". The formula

$$\mu X.(d \vee \Diamond X)$$

holds in a state $x \in K$ iff some state where d holds is reachable from x by a series of transitions. Put into words, it says that a disaster is possible on some future path. Its negation:

$$\nu X.(\neg d \wedge \Box X)$$

says that a disaster cannot be reached, and the formula

$$\nu X.((m \to \mu Y.(d \vee \Box Y)) \wedge \Box X)$$

means that every future mistake will inevitably lead to a disaster. □

If the model $\mathcal{K}$ is finite, then for every fixpoint formula, the corresponding sequence of approximants (3) stabilizes after a finite number of steps. This means that the inductive definition (2) actually provides an effective procedure to compute the interpretation of a formula in a finite model. It is easy to see that this computation can be performed in space polynomial in the size of φ and K, which gives a polynomial space algorithm for the following *model checking problem*:

> Given a closed formula φ and a state x in a finite model K, does $x \in [\![\varphi]\!]_\varnothing$?

This algorithm is rather naive. More sophisticated procedures rely on a correspondence between μ-formulas and parity games (see, e.g., [26]), where recent breakthroughs [27] achieved quasi-polynomial time solutions to the model-checking problem.

III. Sets with atoms

There are several, essentially equivalent ways to introduce sets with atoms, also known as nominal sets [16]. We follow the set-theoretic presentation of [17], culminating in the notion of orbit-finite sets [16, 18] and computable operations on them.

Fix a countably infinite set $\mathbb{A}$, whose elements we shall call *atoms*. A bijection on $\mathbb{A}$ will be called an atom automorphism, and the group of atom automorphisms is denoted Aut($\mathbb{A}$).[1] The finite bijection that swaps atoms a and b is denoted $(a\ b)$.

A set with atoms is a set that can have atoms, or other sets with atoms, as elements. Formally, the universe $\mathcal{U}^{\mathbb{A}}$ of sets with atoms is defined by a von Neumann-like hierarchy, by transfinite induction on ordinal numbers α:

$$\mathcal{U}^{\mathbb{A}}_0 = \varnothing, \qquad \mathcal{U}^{\mathbb{A}}_{\alpha+1} = \mathcal{P}(\mathcal{U}^{\mathbb{A}}_\alpha) + \mathbb{A}, \qquad \mathcal{U}^{\mathbb{A}}_\beta = \bigcup_{\alpha<\beta} \mathcal{U}^{\mathbb{A}}_\alpha,$$

for β a limit ordinal; here $\mathcal{P}$ denotes the powerset construction and + the disjoint union of sets.

There is a canonical group action of atom automorphisms on $\mathcal{U}^{\mathbb{A}}$:

$$\_ \cdot \_ : \mathcal{U}^{\mathbb{A}} \times \text{Aut}(\mathbb{A}) \to \mathcal{U}^{\mathbb{A}};$$

an automorphism acts on a set by consistently renaming all atoms in it. Formally, this is defined by transfinite induction.

For a finite set $S \subseteq \mathbb{A}$, let $\text{Aut}_S(\mathbb{A})$ be the group of those automorphisms of $\mathbb{A}$ that fix every element of S. We say that S *supports* a set X if $X \cdot \pi = X$ for every $\pi \in \text{Aut}_S(\mathbb{A})$. It should be clear that every set X is supported by the set of all atoms that appear in it, i.e., are elements of X or appear in some elements of it. But the notion of support is more subtle than that: for example, for any atoms $a, b, c \in \mathbb{A}$, the set $\mathbb{A} \setminus \{a, b, c\}$ is supported by $\{a, b, c\}$. The intuition is that a set X is supported by S if it can be defined in a way that does not mention any specific atoms other than those in S.

A set with atoms is considered legal only if it is hereditarily finitely supported, i.e., if it has a finite support, its every element has some (perhaps different) finite support and so on. We shall only consider sets with this property.

If X has a finite support then it has the least finite support (see [16] for a proof), denoted $\text{supp}(X)$. A set is *equivariant* if $\text{supp}(X) = \varnothing$.

Relations and functions are sets in the usual sense, so the notions of support and equivariance applies to them as well. Unfolding the definitions, for equivariant sets X and Y, a relation $R \subseteq X \times Y$ is equivariant if $(x, y) \in R$ implies $(x \cdot \pi, y \cdot \pi) \in R$ and a function $f : X \to Y$ is equivariant if $f(x \cdot \pi) = f(x) \cdot \pi$, for every $\pi \in \text{Aut}(\mathbb{A})$. The intuition is that a relation or function is equivariant if it can be defined in a way that only relies on comparing atoms for equality, but without mentioning any specific atoms.

For any x with atoms, the *S-orbit* of x is the set

$$\{x \cdot \pi \mid \pi \in \text{Aut}_S(\mathbb{A})\}.$$

For example, if S supports x then the S-orbit of x is the singleton $\{x\}$.

For any S, S-orbits form a partition of the universe $\mathcal{U}^{\mathbb{A}}$. Moreover, for any S-supported set X, the S-orbits of its elements form a partition of X. Then X is *S-orbit-finite* if it is a union of finitely many S-orbits. Then, for any finite $T \supseteq S$, X is also T-orbit-finite. Thanks to this observation, we may drop the qualifier and simply call X *orbit-finite*, meaning "S-orbit-finite for any/every S that supports X".

Example 2.
- Any classical set (without atoms) is an equivariant set with atoms. Its every element forms its own orbit, so such a set is orbit-finite iff it is finite.
- An atom $a \in \mathbb{A}$ has no elements, and it is supported by $\{a\}$. A subset of $\mathbb{A}$ is finitely supported iff it is finite or co-finite; a finite set is supported by itself, and a co-finite one by its complement.
- The set $\mathbb{A}$ of atoms, the set $\binom{\mathbb{A}}{2}$ of two-element sets of atoms, and the set $\mathbb{A}^2$ of ordered pairs of atoms, are equivariant sets. The first two have a single orbit each, and the last one has two equivariant orbits:

$$\mathbb{A}^2 = \{(a, a) \mid a \in \mathbb{A}\} \cup \{(a, b) \mid a \neq b \in \mathbb{A}\}. \qquad (4)$$

Similarly, $\mathbb{A}^n$ is orbit-finite for every $n \in \mathbb{N}$, with the number of orbits equal to the n'th Bell number. The set $\mathbb{A}^*$ of finite sequences of atoms is not orbit-finite.

[1] An alternative definition, taken in [16, 17], is to let Aut($\mathbb{A}$) be the group of all *finite* bijections, i.e., those that fix almost all $a \in \mathbb{A}$. This distinction does not make any difference for our purposes here.

- The powerset $\mathcal{P}(\mathbb{A})$ is equivariant itself, but it contains elements that are not finitely supported, and therefore is not considered a legal set with atoms. For any legal set with atoms X, the set $\mathcal{P}_{\text{fs}}(X)$ of all finitely supported subsets of X and the smaller set $\mathcal{P}_{\text{fin}}(X)$ of all finite subsets of X, are both legal and supported by $\operatorname{supp}(X)$. However, both of them may fail to be orbit-finite even if X is orbit-finite.
- There are four equivariant binary relations on $\mathbb{A}$: the empty relation, equality, inequality and the full relation.
- There is no equivariant function from $\binom{\mathbb{A}}{2}$ to $\mathbb{A}$, but
$$\{(\{a,b\},a) \mid a \neq b \in \mathbb{A}\} \tag{5}$$
is a legal equivariant relation, and the function constant at a is supported by $\{a\}$. The only equivariant function from $\mathbb{A}$ to $\mathbb{A}$ is the identity. The only equivariant functions from $\mathbb{A}^2$ to $\mathbb{A}$ are projections, and the only equivariant function from $\mathbb{A}$ to $\mathbb{A}^2$ is the diagonal $a \mapsto (a,a)$. □

Example 3. The following Kripke model represents a FIFO buffer of size 3 that can input and output atoms. The set of states is $K = \mathbb{A}^3$. The transition relation is:

$$(a,b,c) \longrightarrow (d,e,f) \qquad \iff \qquad e = a,\ f = b.$$

Let the set of basic predicates be

$$\mathbb{B} = \{\text{in}_a \mid a \in \mathbb{A}\} \cup \{\text{out}_a \mid a \in \mathbb{A}\} \tag{6}$$

interpreted by a satisfaction relation $\vDash$ such that:

$$\text{pred}(a,b,c) = \{\text{in}_a, \text{out}_c\}.$$

This is an equivariant Kripke model with atoms, with five orbits of states and two orbits of basic predicates. For every atom $a \in \mathbb{A}$, every state in K satisfies the μ-formula

$$\nu X.((\text{in}_a \to \Box\Box\text{out}_a) \wedge \Box X), \tag{7}$$

meaning that whenever a is input to the buffer, it will be output exactly two steps later. □

It is easy to check that, for any X and Y with atoms,

$$\operatorname{supp}(X,Y) = \operatorname{supp}(X) \cup \operatorname{supp}(Y)$$

and, for any function $f : X \to Y$ and $x \in X$,

$$\operatorname{supp}(f(x)) \subseteq \operatorname{supp}(f,x).$$

The cartesian product of two orbit-finite sets is orbit-finite, and a finitely supported subset of an orbit-finite set is orbit-finite. As a result, any finitely supported relation or function between orbit-finite sets, considered as a set of pairs, is orbit-finite. Furthermore, for any finite $S \subseteq \mathbb{A}$ and any X and Y supported by S, if X is orbit-finite then the set of S-supported functions from X to Y is orbit-finite, and it is finite if additionally Y is orbit-finite. This follows from the fact that for a fixed S, an orbit-finite set can have only finitely many S-supported subsets.

Orbit-finite sets, although usually infinite, can be presented by finite means and are therefore amenable to algorithmic manipulation. There are a few ways to do this. One way, used in [19], is to present orbit-finite sets by formal set-builder expressions of the form

$$\{e \mid v_1, \ldots, v_n \in \mathbb{A}, \varphi\}$$

where e is again an expression, v_i are bound atom variables and φ is a first-order formula with equality. We refer to [19] for a precise formulation (and to [28] for a proof that all orbit-finite sets can be presented this way); suffice it to say that, after adding some mild syntactic sugar, expressions such as (4), (5), or (6) above are of this form. The set defined by such an expression is supported by the atoms that appear freely in the expression.

Under the set-builder representation it is not trivial to check whether two representations define the same set. (Indeed, it is a PSPACE-complete problem, as that is the complexity of the first-order theory of equality [29].) However, set equality and other basic operations on orbit-finite sets are computable in polynomial space on their representations, including:

- checking whether one set is an element (or a subset) of another,
- union and intersection of sets, cartesian product, set difference,
- applying an orbit-finite function to an argument, composing functions or relations,
- finding the image of a subset along a relation,
- checking whether a finite set S supports a given set, calculating the least support of a set,
- partitioning a given set into S-orbits, calculating the S-orbit of a given element.

These basic operations have been implemented as components of atomic programming languages [30, 31].

Originally [16, 17], sets with atoms (known as nominal sets in this context) were developed as an algebraic approach to name binding in syntax, with concepts such as name abstraction space introduced to that end. The modal μ-calculus, with its fixpoint operators, is an archetypical example of a formalism where name binding plays a crucial role, and one could use nominal techniques to study syntactic aspects of it. We emphasize that this is *not* what we aim to do. In fact, we do not even introduce the concepts of name abstraction, α-conversion, etc. Rather, we shall build atoms into models, formulas, variable names, etc., and variable binding in fixpoint formulas will appear on top of that, independently. In this we are similar to studies of nominal rewriting such as [32].

IV. μ-CALCULUS WITH ATOMS

In this section we present the syntax and semantics of the μ-calculus with atoms, following [15]. As an initial motivation, consider the formula (7) again. That formula refers to a particular atom a, but the property that it aims to formulate does not really depend on that atom. One would like to write a formula that says: whenever an atom is input, it will be output exactly two steps later. A formula like

$$\nu X.\left(\bigwedge_{a\in\mathbb{A}} (\text{in}_a \to \Box\,\Box\,\text{out}_a) \wedge \Box X\right) \tag{8}$$

does not fit the syntax of the μ-calculus as defined in (1) because the inner conjunction is infinite. The μ-calculus with atoms extends the classical calculus with such infinitary (but orbit-finitary) Boolean operators.

A. Syntax and semantics

Considering now the sets $\mathbb{X}$ of variables and $\mathbb{B}$ of basic predicates as sets with atoms, we extend the syntax (1) with

$$\varphi ::= \cdots \mid \bigvee_{a \in \mathbb{A}} \varphi_a. \tag{9}$$

The precise meaning of this is that $\varphi_{(-)}$ is a finitely supported function from $\mathbb{A}$ to the set of formulas. Strictly speaking, the a in the expression $\bigvee_{a\in\mathbb{A}} \varphi_a$ is merely decoration, and the expression could be written just as $\bigvee \varphi_{(-)}$. As a syntactic sugar, we use the expected

$$\bigwedge_{a\in\mathbb{A}} \varphi_a = \neg \bigvee_{a\in\mathbb{A}} \neg\varphi_a.$$

The extended calculus is interpreted in Kripke models with atoms, where the set of states, the transition relation and the satisfaction relation are all finitely supported. The semantics extends (2) with:

$$x \in \left[\!\!\left[\bigvee_{a\in\mathbb{A}} \varphi_a \right]\!\!\right]_\xi \iff x \in [\![\varphi_a]\!]_\xi \qquad \text{for some } a \in \mathbb{A}.$$

It is straightforward to check (see [15, Lem. 4.7]) that the function $[\![\_]\!]_{(-)}$ is supported by $\mathrm{supp}(\mathcal{K})$, so:

$$\mathrm{supp}([\![\varphi]\!]_\xi) \subseteq \mathrm{supp}(\varphi, \xi, \mathcal{K}).$$

According to this definition, formulas are not finite objects anymore. However, they remain finitely supported and orbit-finite. Moreover, every formula has a well-defined finite height, so there is no need for transfinite induction in structural reasoning on formulas. If a formula φ is closed, then its every subformula ψ has only finitely many free variables, because each free variable in ψ must have been introduced by a fixpoint operator on the finite path from the root of φ to the root of ψ. As a result, contexts ξ can still be seen as partial functions with finite domains.

Our (9) is not the only way to introduce orbit-finitary disjunction into the μ-calculus. Several equivalent alternatives exist. In [15], a simple syntax

$$\varphi ::= \cdots \mid \bigvee \Phi$$

was used, where Φ is a finitely supported, orbit-finite set of formulas. One could also use

$$\varphi ::= \cdots \mid \bigvee_{a \notin S} \varphi_a$$

where S is a finite set of atoms and $\varphi_{(-)}$ is a function, supported by S, from $\mathbb{A} \setminus S$ to the set of formulas, or even

$$\varphi ::= \cdots \mid \exists a.\varphi$$

where a is an atom and φ a formula. Each approach has its advantages; see Appendix A for a comparison. We stick to (9) as it will make our proofs easier to follow.

B. Model checking and expressivity

Since a formula $\bigvee_{a\in\mathbb{A}}$ (and therefore every formula of the μ-calculus with atoms) is a finitely supported, orbit-finite object, it may be presented by finite means as an instance of a computational problem such as model checking. Indeed, model checking of formulas on orbit-finite models $\mathcal{K}$ is decidable. To see this (see [15, Thm. 5.1] for more details), proceed as for the classical μ-calculus in Section II, inductively computing the (finitely supported) sets $[\![\varphi]\!]_\xi \subseteq K$ for all subformulas φ of a given formula.

For the case $\varphi = \bigvee_{a\in A} \varphi_a$, first calculate $S = \mathrm{supp}(\mathcal{K}, \varphi, \xi)$. Then pick any atom $b \notin S$, inductively compute $R = [\![\varphi_b]\!]_\xi$, then compute $\mathcal{R} \subseteq \mathcal{P}K$ the S-orbit of R, and return

$$\bigcup \mathcal{R} \cup [\![\varphi_{a_1}]\!]_\xi \cup \cdots [\![\varphi_{a_n}]\!]_\xi$$

where the $[\![\varphi_{a_i}]\!]_\xi$ are computed inductively for $\{a_1, \ldots, a_n\} = S$.

For the case $\varphi = \mu X.\psi$, it is important to see that the sequence of approximants Ψ^α (see (3) in Section II) stabilizes after finitely many steps. This is because, by an easy induction on the ordinal α, every Ψ^α is supported by the same set $S = \mathrm{supp}(\mathcal{K}, X, \psi, \xi)$. Since K is orbit-finite, it can only have finitely many S-supported subsets.

C. Expressivity

Our calculus can now express formulas such as (8), but also many others. For example, still over the same set $\mathbb{B}$ of basic predicates as in Example 3, the formula

$$\nu X. \bigwedge_{a\in\mathbb{A}} (\mathrm{in}_a \to \Box(\nu Y. \neg\mathrm{in}_a \wedge \Box Y))$$

says that on no computation path the same atom is ever input more than once. (No state in the model from Example 3 satisfies this formula.)

However, the expressive power of the calculus leaves much to be desired. In a simple setting where $\mathbb{B} = \mathbb{A}$, i.e., basic predicates are just atoms, property #PATH is not definable, and so are many properties that are useful in practice.

V. HISTORY-DEPENDENT μ-CALCULUS

We extend the μ-calculus with atoms with formulas of the form $\sharp p$, where $p \in \mathbb{B}$ is a basic predicate. The intuition is that $\sharp p$ holds if the predicate p has never been true before on the transition path that led to the current state.

Put together, we extend (1) to:

$$\varphi ::= \top \mid p \mid \sharp p \mid \neg\varphi \mid \varphi \vee \varphi \mid \bigvee_{a\in\mathbb{A}} \varphi_a \mid \Diamond\varphi \mid X \mid \mu X.\varphi;$$

some syntactic sugar is introduced as before.

To simplify the setting, from now on we assume that every basic predicate is built of at most one atom. Formally, we assume that

$$\mathbb{B} = B \times \mathbb{A} + C \tag{10}$$

where B and C are some finite sets. Furthermore we assume that every state x in a model satisfies only finitely many basic predicates, i.e., that $\mathrm{pred}(x)$ is finite. All examples considered so far, and further ones that we present in Section VI below, satisfy these assumptions.

For the semantics, a formula φ is now interpreted relative to a *history* $H \in \mathcal{P}_{\text{fin}}\mathbb{B}$, i.e., a finite set of basic predicates. A context ξ also interprets variables relative to histories, so it is of the type:

$$\xi : \mathbb{X} \rightharpoonup \mathcal{P}_{\text{fin}}\mathbb{B} \to \mathcal{P}K.$$

Semantics is formally defined by induction, extending (2) to:

$$\begin{aligned}
x \in [\![\top]\!]^H_\xi & \qquad \text{for all } x \in K \\
x \in [\![p]\!]^H_\xi & \iff p \in \operatorname{pred}(x) \\
x \in [\![\sharp p]\!]^H_\xi & \iff p \notin H \\
x \in [\![\neg\varphi]\!]^H_\xi & \iff x \notin [\![\varphi]\!]^H_\xi \\
x \in [\![\varphi \vee \psi]\!]^H_\xi & \iff x \in [\![\varphi]\!]^H_\xi \text{ or } x \in [\![\psi]\!]^H_\xi \\
x \in \left[\!\!\left[\bigvee_{a \in \mathbb{A}} \varphi_a\right]\!\!\right]^H_\xi & \iff x \in [\![\varphi_a]\!]^H_\xi \text{ for some } a \in \mathbb{A} \\
x \in [\![\Diamond\varphi]\!]^H_\xi & \iff \exists y \in [\![\varphi]\!]^{H \cup \operatorname{pred}(x)}_\xi \text{ s.t. } x \longrightarrow y \\
x \in [\![X]\!]^H_\xi & \iff x \in \xi(X)(H) \\
[\![\mu X.\varphi]\!]_\xi & \;=\; \operatorname{lfp}\left(\Psi \mapsto [\![\varphi]\!]_{\xi[X \mapsto \Psi]}\right).
\end{aligned} \tag{11}$$

Notice how in the clause for $\Diamond\varphi$, the basic predicates in $\operatorname{pred}(x)$ are added to the current history. Also notice how in the clause for $\mu X.\varphi$, the least fixpoint is taken over the set of functions of the type

$$\Psi : \mathcal{P}_{\text{fin}}\mathbb{B} \to \mathcal{P}K,$$

that is, over families (of sets of states) indexed by histories. This is unavoidable: one could not define the set $[\![\mu X.\varphi]\!]^H_\xi$ for a specific H by a fixpoint construction, since the variable X may be evaluated inside φ for a history richer than H.

Functions Ψ as above, ordered by (pointwise) inclusion, form a complete lattice, so the least fixpoint is well-defined. However, unlike in the history-free setting, (11) does not provide a method to compute the semantics of a formula. The trouble is with fixpoint formulas: here approximants Ψ^α used as in (3) to define a least fixpoint are not orbit-finite objects, so it is not clear how to even represent them by finite means, let alone how to compute $\Psi^{\alpha+1}$ from Ψ^α. Even if that was somehow solved, it is not clear why the sequence of approximants should stabilize after finitely many steps. Indeed, unlike in Section IV-B, here approximants are families indexed by elements of an orbit-infinite set and there are infinitely many of them, even for a fixed support S.

We must proceed with care, because the presence of data atoms sometimes makes seemingly innocent problems become undecidable; examples include language equivalence for nondeterministic orbit-finite automata [33], model checking for CTL* with atoms [15], and the existence of a finitely supported homomorphism between two orbit-finite graphs [34]. Nevertheless, in Section VII, we shall deal with the above issues and prove that model checking on orbit-finite models is indeed decidable. For now, let us mention that the problematic property #PATH from Section IV-C is now easy to define by a formula:

$$\nu X.\left(\bigwedge_{a \in \mathbb{A}} (a \to \sharp a) \wedge \Diamond X\right).$$

In fact, as the following examples show, the history-dependent calculus is expressive enough to define some properties that are interesting from the perspective of system verification.

VI. Examples

We present two scenarios which rely on comparing data atoms for equality and can be encoded in our history-dependent μ-calculus. The first one features two processes trying to concurrently access a critical section, while the second one is a version of the cryptographic Needham–Schroeder protocol [21].

A. Critical section

Two concurrent processes A and B have access to a common critical section S. In order not to enter it together, they can lock it with a password they must generate each time before entering it. Only the process that locked the critical section can interact with it. The section can then be unlocked using the same password. A typical interaction is depicted in Figure 1, where the boxes on each thread show what password is remembered, and the dashed arrows are actions that cannot be performed, because of the drawn inequalities.

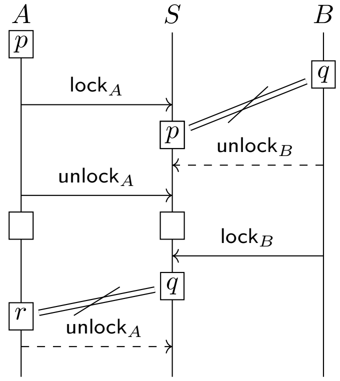

Fig. 1. An interaction between the processes and the critical section

We want to make sure that the two processes will never enter the critical section together. Of course, this is only true under the assumption that the processes will not generate the same passwords. Building that requirement into the model would make it orbit-infinite (an entire history of passwords would have to be remembered in the state), so instead our model will allow processes to generate arbitrary passwords, and the safety condition will be included in the formula that defines the desired property.

Formally, we build an orbit-finite, equivariant model $\mathcal{K} = (K, \longrightarrow, \vDash)$ on the set of predicates

$$\mathbb{B} = \{\mathsf{p}_a \mid a \in \mathbb{A}\} \cup \{\mathsf{lock}_A, \mathsf{lock}_B, \mathsf{unlock}_A, \mathsf{unlock}_B\},$$

where p_a means that the password a has just been generated, and the other predicates have obvious meanings. A state $x \in$

K basically records the passwords each thread remembers; however, for the states to satisfy the right predicates, we need several copies of such a record. Formally, a state s a tuple

$$x = (a_A, a_B, a_S, s, t) \tag{12}$$

where the a_X's are passwords remembered by the three threads (these may be some $a \in \mathbb{A}$ or $\varnothing$), and s and t form a finite component that models the current stage of interaction. Specifically, the value of $s \in \{\mathsf{p}, \mathsf{lock}, \mathsf{unlock}, \varnothing\}$ specifies whether a password is just being generated, the section locked or unlocked (or perhaps none of that), and $t \in \{\mathsf{A}, \mathsf{B}, \varnothing\}$ says which one of the two processes is performing the action. In a state x, if $s \in \{\mathsf{lock}, \mathsf{unlock}\}$ then $x \vDash s_t$; if $s = \mathsf{p}$ then $x \vDash \mathsf{p}_{a_t}$; otherwise x satisfies no basic predicates.

In defining the transition relation we use a simple notation for updating components of states: $x[B \mapsto a, s \mapsto \mathsf{p}]$ denotes the state x as in (12) with the component a_B set to a and the component s set to p, and so on. Transitions from a state x as in (12) are as follows:

- if $s \neq \varnothing$ or $t \neq \varnothing$, then $x \longrightarrow x[s, t \mapsto \varnothing]$ is the only transition (the states with $s = t = \varnothing$ are the "normal" states of interaction, the other ones are there just to satisfy the right predicates); otherwise,
- $x \longrightarrow x[A \mapsto a, s \mapsto \mathsf{p}, t \mapsto \mathsf{A}]$ for every $a \in \mathbb{A}$, and similarly for B (A and B can generate passwords),
- if $a_S = \varnothing$ and $a_A \neq \varnothing$, then $x \longrightarrow x[S \mapsto a_A, s \mapsto \mathsf{lock}, t \mapsto \mathsf{A}]$ and similarly for B (A and B can enter the critical section when it is free),
- if $a_A = a_S \neq \varnothing$, then $x \longrightarrow x[A, S \mapsto \varnothing, s \mapsto \mathsf{unlock}, t \mapsto \mathsf{A}]$ and similarly for B (when A or B unlocks the critical section, they forget their current password and have to generate a new one).

Note that this indeed defines an orbit-finite model.

To define the property, we start by defining a formula

$$\mathsf{safe} = \bigwedge_{a \in \mathbb{A}} (\mathsf{p}_a \to \sharp \mathsf{p}_a)$$

which says that whenever there is a newly generated password, then it has not been generated before. We can then define

$$P_A = \nu X.(\mathsf{safe} \to (\mathsf{unlock}_A \vee (\neg \mathsf{unlock}_B \wedge \Box X))),$$

whose meaning is that, in all safe paths (where passwords are generated at most once) from the current state, B cannot unlock the critical section unless A has already unlocked it. We define P_B similarly. The property we are interested in is

$$\nu X.(\mathsf{safe} \to ((\mathsf{lock}_A \to P_A) \wedge (\mathsf{lock}_B \to P_B) \wedge \Box X)),$$

which means that, for any safe path, if a process locks the critical section, then the same process must unlock it before the other can. This formula indeed holds in the initial state of this model (one where all components are $\varnothing$).

B. The Needham–Schroeder protocol

The Needham–Schroeder protocol allows two agents to verify each other's identities in an unsafe network by using public-key cryptography. A normal interaction according to the protocol is depicted in the left-hand part of Figure 2. It proceeds as follows:

1) A generates a nonce n_A,
2) A sends (A, n_A) to B, encrypted with k_{PB} (B's public key),
3) B deciphers (A, n_A) using k_{SB} (B's secret key),
4) B generates a nonce n_B,
5) B sends (n_A, n_B) to A, encrypted with k_{PA},
6) A deciphers (n_A, n_B) using k_{SA} and checks that the first nonce is the one she sent,
7) A sends n_B to B, encrypted with k_{PB},
8) B deciphers n_B using k_{SB} and checks that it is the nonce he sent.

We model three agents: Alice, Bob, and Eve (A, B, and E). Alice and Bob follow the protocol, while Eve does not necessarily, which can lead to a man-in-the-middle attack [35], as depicted in the right-hand part of Figure 2. We want to see if a state can be reached where Alice or Bob think they know someone's identity but they are actually mistaken.

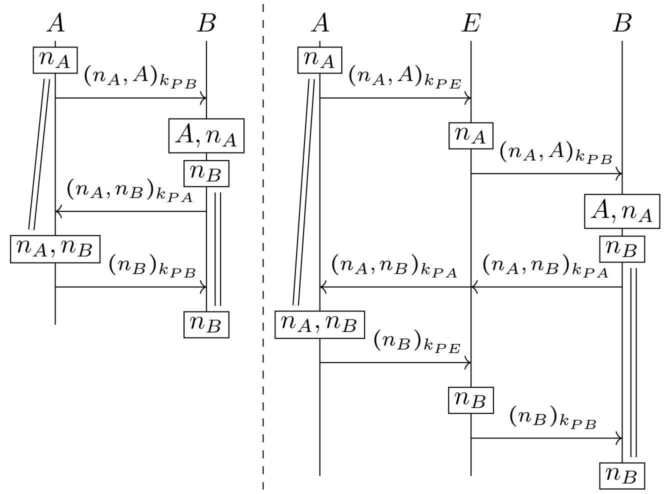

Fig. 2. The protocol and the man-in-the-middle attack

Formally, we build an equivariant model $\mathcal{K} = (K, \longrightarrow, \vDash)$ on the set of predicates

$$\{\mathsf{n}_a \mid a \in \mathbb{A}\} \cup \{\mathsf{k}_a \mid a \in \mathbb{A}\} \cup \{\mathsf{bad}\},$$

where n_a is a nonce (a state satisfies n_a if n_a has just been generated), k_a is a key (for brevity, we only mention secret keys), and bad indicates bad states. Each state $x \in K$ records the state of Alice, Bob, and Eve. As in the previous example, we need a few copies of these states, so that $x \vDash \mathsf{n}_a$ and $x \vDash \mathsf{k}_a$ happen only when a nonce or key is generated.

We model Alice and Bob as 7-tuples (k, n, i, n', i', c, p) of a key k, nonces n (generated by that agent) and n' (received from another agent), identities i (whom they are communicating with) and i' (whom they hope to be communicating with) from $\{A, B, E\}$, a ciphertext c (which is modeled as a tuple of the key needed to decipher it and the information contained in it), and the phase p of the protocol they are at (p is 0 or a pair of a number from 1..7 together with a flag $f \in \{\mathsf{sndr}, \mathsf{rcvr}\}$, depending on whether they sent the first message or received it). All of these variables may be $\varnothing$. Eve is modeled similarly, except she does not need to remember any identities, nor the phase of the protocol she is at, and is thus modeled as a quadruple (k, n, n', c). Formally, a state of $\mathcal{K}$ is then a quintuple $(\sigma_A, \sigma_B, \sigma_E, s, t)$ where σ_A, σ_B, and σ_E are the states of Alice, Bob, and Eve, and $s \in \{\mathsf{n}, \mathsf{k}, \varnothing\}$ and $t \in \{A, B, E, \varnothing\}$ specify if an action is being done, and which agent is doing it. We shall refer to Alice's key in σ_A as k_A, etc. In a state x, if p_A is $(7, \mathsf{sndr})$ or $(8, \mathsf{rcvr})$ (the possible values of p at the end of the protocol) and $i_A \neq i'_A$ (Alice is mistaken on the identity of the agent she is communicating with), then $x \vDash \mathsf{bad}$, and similarly for Bob. If $s \neq \varnothing$ and $t = \mathsf{n}$, then $x \vDash \mathsf{n}_{n_s}$. If $s \neq \varnothing$ and $t = \mathsf{k}$, then $x \vDash \mathsf{k}_{k_s}$.

Any state x where $s \neq \varnothing$ or $t \neq \varnothing$ has a single transition $x \longrightarrow x[s, t \mapsto \varnothing]$. Otherwise, for Alice and Bob, the transition relation follows the protocol (we let variables X, Y, and Z range over identities):

0) they can always generate a public/secret key pair, and must do so before beginning the protocol: $x \longrightarrow x[k_X \mapsto k, p_X \mapsto 0, s \mapsto \mathsf{k}, t \mapsto X]$ for any $k \in \mathbb{A}$,
1) if $p_X = 0$, then $x \longrightarrow x[n_X \mapsto a, p_X \mapsto (1, \mathsf{sndr}), s \mapsto \mathsf{n}, t \mapsto X]$ for any $a \in \mathbb{A}$ (X generates a nonce),
2) if $p_X = (1, \mathsf{sndr})$ and $p_Y = 0$, then $x \longrightarrow x[i_X \mapsto Y, i'_X \mapsto Z, p_X \mapsto (2, \mathsf{sndr}), i_Y \mapsto X, c_Y \mapsto (k_Z, X, n_X), p_Y \mapsto (2, \mathsf{rcvr})]$ (X sends Y a test, thinking they are Z),
3) if $p_Y = (2, \mathsf{rcvr})$, and $c_Y = (k_Y, X, n)$ then $x \longrightarrow x[n'_Y \mapsto n, i'_Y \mapsto X, p_Y \mapsto (3, \mathsf{rcvr})]$ (if the cipher is encrypted with the right key, it is deciphered),
4) if $p_Y = (3, \mathsf{rcvr})$ then $x \longrightarrow x[n_Y \mapsto b, p_Y \mapsto (4, \mathsf{rcvr}), s \mapsto \mathsf{n}, t \mapsto Y]$ for any $b \in \mathbb{A}$ (Y generates a nonce),
5) if $p_X = (2, \mathsf{sndr})$, $i_X = Y$, $p_Y = (4, \mathsf{rcvr})$, and $i_Y = X$, then $x \longrightarrow x[c_X \mapsto (k_{i'_Y}, n'_Y, n_Y), p_X \mapsto (5, \mathsf{sndr}), p_Y \mapsto (5, \mathsf{rcvr})]$ (Y sends a test, encrypted with the key of the agent they think they are communicating with),
6) if $p_X = (5, \mathsf{sndr})$ and $c_X = (k_X, n_X, n')$, then $x \longrightarrow x[n'_X \mapsto n', p_X \mapsto (6, \mathsf{sndr})]$ (if the key matches, X deciphers the message, and goes on with the protocol only if the nonce is the one they sent),
7) if $p_X = (6, \mathsf{sndr})$ and $p_Y = (5, \mathsf{rcvr})$, then $x \longrightarrow x[p_X \mapsto (7, \mathsf{sndr}), c_Y \mapsto (k_{i'_X}, n'_X), p_Y \mapsto (7, \mathsf{rcvr})]$ (X sends an answer, encrypted with the key of the agent they think they are communicating with),
8) if $p_Y = (7, \mathsf{rcvr})$ and $c_Y = (k_Y, n_Y)$, then $x \longrightarrow x[p_Y \mapsto (8, \mathsf{rcvr})]$ (Y acknowledges the answer if they can decipher it and it contains the nonce they sent).

They can also, at any time, decide to reset the protocol: $x \longrightarrow x[n_X, i_X, n'_X, i'_X, c_X \mapsto \varnothing, p \mapsto 0]$. For communications with Eve, the transitions are nearly the same, except there needs be no checks about fields not contained in our model of Eve. There are also transitions corresponding to what Eve can do:

- she can forge messages with a false identity, meaning she can send messages as in steps 2), 5), and 7) above, except she does not need send E as her identity, she can also send A or B.
- when deciphering messages, she decides what to do with nonces (e.g., she may throw them away if they are useless to her),
- she can "proxy" ciphertexts, meaning, if X (Alice or Bob) is in phase 0, $(2, \mathsf{sndr})$, or $(5, \mathsf{rcvr})$ (i.e., ready to receive a message), and $c_E \neq \varnothing$, then $x \longrightarrow x[c_X \mapsto c_E, p_X \mapsto p_X^+]$, where $0^+ = (2, \mathsf{rcvr})$, $(2, \mathsf{sndr})^+ = (5, \mathsf{sndr})$, and $(5, \mathsf{rcvr})^+ = (7, \mathsf{rcvr})$.

Note that this indeed defines an orbit-finite model.

Obviously, if there is a collision between keys or nonces (e.g., if $k_E = k_A$ or $n_E = n_A$) then the protocol can be broken. We thus define the formula

$$\mathsf{safe} = \bigwedge_{a \in \mathbb{A}} ((k_a \to \sharp k_a) \wedge (n_a \to \sharp n_a))$$

meaning there are no such collisions. The formula we are interested in is then

$$\neg \mu X.(\mathsf{safe} \wedge (\mathsf{bad} \vee \Diamond X)), \tag{13}$$

meaning that no safe path can reach a bad state.

Because of a well-known attack, this formula does not hold in our model. We can modify this to model a correction of the protocol [35] by sending (n_A, n_B, B) encrypted with k_{PA} in step 5), and A accepting only if B is the agent they think they are communicating with. Formula (13) then holds. Note that this does not mean that there are no attacks against the modified protocol, but only that Eve equipped with capabilities defined in our model cannot break it.

VII. Decidability of model checking

In this section we shall prove that model checking formulas of the history-dependent μ-calculus against orbit-finite models is decidable. More specifically, our aim is to show how to inductively compute the set $[\![\varphi]\!]_{\varnothing}^{\varnothing}$, defined according to (11), from a given closed formula φ and an orbit-finite model $\mathcal{K}$.

To make the proof easier to follow, we will make two further simplifying assumptions. From now on, assume that the model $\mathcal{K}$ is equivariant and, strengthening the assumption (10), that basic predicates are simply atoms: $\mathbb{B} = \mathbb{A}$. At the end of this section we explain how these assumptions can be dropped.

Since we have already assumed in Section V that $\mathrm{pred}(x)$ is finite for every state x, from $\mathbb{B} = \mathbb{A}$ it follows that

$$\mathrm{pred}(x) \subseteq \mathrm{supp}(x),$$

a useful property to keep in mind.

A. First attempts

Ignoring for a moment the fact that ξ interprets variables as families indexed by arbitrary histories and it is not clear how to present it by finite means, how could we compute $[\![\varphi]\!]_{\xi}^{H}$ from a given φ, ξ and H? Here is an initial idea: intuitively, to check whether $x \in [\![\varphi]\!]_{\xi}^{H}$, it should not matter whether an atom a belongs to H if that atom is not present in the support of the current situation, that is in the support of x, φ and perhaps ξ. Such "locally fresh" atoms should be irrelevant for the validity of the statement. So perhaps one could restrict attention only to those statements where $H \subseteq \mathrm{supp}(x, \varphi, \xi)$, which would provide some bound on the size of histories considered.

This idea does not quite work, and it is not enough to keep track of how global histories intersect with local supports. To see why, consider the equivariant formula

$$\varphi = \bigvee_{a \in \mathbb{A}} \neg \sharp a$$

which says that *some atom is not fresh*. Since the formula is closed, it can be interpreted in the empty context $\xi = \varnothing$. If one wants to check whether $x \in \llbracket \varphi \rrbracket^H_\xi$ for an equivariant state $x \in K$, then the local support $\mathrm{supp}(x, \varphi, \xi)$ is empty. However, to check whether $x \in \llbracket \varphi \rrbracket^H_\xi$ we must know whether the history H is empty, and this information is lost in the (necessarily empty) intersection of H with the local support.

This motivates a refinement of the initial idea: together with the intersection of a global history H with a local support S, one should remember the number (but not the identities) of those atoms in H which do not belong to S. This idea will turn out to work, so we shall now present it more formally.

B. Local-history semantics

A *local history* consists of a history together with a natural number:

$$(n, H) \in \mathbb{N} \times \mathcal{P}_{\mathrm{fin}}\mathbb{A}.$$

The number n is the *anonymous part*, and H is the *non-anonymous part* of the local history.

Basic operations for anonymizing and de-anonymizing atoms in local histories will be useful. For a local history (n, H) and a finite set of atoms S, define:

$$(n, H) \curvearrowleft S = (n + |H \setminus S|, H \cap S). \tag{14}$$

This operation restricts the local view of a history to atoms in S, and anonymizes all other atoms. The symbol $\curvearrowleft$ is the middle ground between the set intersection symbol (as the non-anonymous part H is intersected with S), and a left-to-right arrow (as atoms are transferred from the non-anonymous to the anonymous part).

Dually, define:

$$(n, H) \hookrightarrow S = (n - |S \setminus H|, H \cup S). \tag{15}$$

This operation is defined only if $|S \setminus H| \leq n$. It forces all atoms in S into the non-anonymous part, drawing them from the anonymous part if needed.

The anonymization operation can be applied to global histories as well, writing $H \curvearrowleft S$ for $(0, H) \curvearrowleft S$. Note that $H \curvearrowleft S$ is a representation of the S-orbit of H, in the sense that $H \curvearrowleft S = H' \curvearrowleft S$ if and only if H and H' are in the same S-orbit. Our informal idea now is that $x \in \llbracket \varphi \rrbracket^H_\xi$ depends only on the S-orbit of H for $S = \mathrm{supp}(x, \varphi, \xi)$ and so our semantics can be computed with only local histories in focus.

To this end, we show an alternative *local-history semantics* of our calculus, where a formula is interpreted for a local history (n, H), and a *localized context* ρ that interprets variables relative to local histories:

$$\rho : \mathbb{X} \rightharpoonup (\mathbb{N} \times \mathcal{P}_{\mathrm{fin}}\mathbb{A}) \to \mathcal{P}K.$$

The semantics is defined inductively as follows:

$$\begin{aligned}
x \in \llparenthesis \top \rrparenthesis^{n,H}_\rho & \qquad \text{for all } x \in K \\
x \in \llparenthesis a \rrparenthesis^{n,H}_\rho & \iff a \in \mathrm{pred}(x) \\
x \in \llparenthesis \sharp a \rrparenthesis^{n,H}_\rho & \iff a \notin H \\
x \in \llparenthesis \neg\varphi \rrparenthesis^{n,H}_\rho & \iff x \notin \llparenthesis \varphi \rrparenthesis^{n,H}_\rho \\
x \in \llparenthesis \varphi \vee \psi \rrparenthesis^{n,H}_\rho & \iff x \in \llparenthesis \varphi \rrparenthesis^{n,H}_\rho \text{ or } x \in \llparenthesis \psi \rrparenthesis^{n,H}_\rho \\
x \in \llparenthesis \bigvee_{a \in \mathbb{A}} \varphi_a \rrparenthesis^{n,H}_\rho & \iff \begin{cases} \text{for some } a \in \mathbb{A},\ x \in \llparenthesis \varphi_a \rrparenthesis^{n,H}_\rho, \text{ or} \\ n > 0 \text{ and for some } a \notin H \cup S, \\ \quad x \in \llparenthesis \varphi_a \rrparenthesis^{((n,H) \curvearrowleft S) \hookrightarrow \{a\}}_\rho, \\ \quad \text{where } S = \mathrm{supp}\left(x, \bigvee_{a \in \mathbb{A}} \varphi_a, \rho\right), \end{cases} \\
x \in \llparenthesis \Diamond\varphi \rrparenthesis^{n,H}_\rho & \iff \begin{cases} \text{there is } y \in K \text{ and } D \subseteq \mathbb{A} \text{ s.t.} \\ x \longrightarrow y \in \llparenthesis \varphi \rrparenthesis^{((n, H \cup \mathrm{pred}(x)) \curvearrowleft S) \hookrightarrow D}_\rho, \\ |D| \leq n, \\ D \subseteq \mathrm{supp}(y) \setminus (H \cup \mathrm{supp}(x, \varphi, \rho)), \\ \text{where } S = \mathrm{supp}(y, \varphi, \rho), \end{cases} \\
x \in \llparenthesis X \rrparenthesis^{n,H}_\rho & \iff x \in \rho(X)(n, H) \\
\llparenthesis \mu X.\varphi \rrparenthesis_\rho & \quad = \quad \mathrm{lfp}\left(\Phi \mapsto \llparenthesis \varphi \rrparenthesis_{\rho[X \mapsto \Phi]}\right).
\end{aligned}$$

In the last equation, the least fixpoint is taken over the set of functions:

$$\Phi : (\mathbb{N} \times \mathcal{P}_{\mathrm{fin}}\mathbb{A}) \to \mathcal{P}K.$$

The equations for the single-orbit disjunction and the diamond modality are considerably more complex than the corresponding ones for the global history semantics (11), so some explanation is in order.

For a formula $\varphi = \bigvee_{a \in \mathbb{A}} \varphi_a$ to hold in a state x given a local history (n, H), it must be that some φ_a holds in the same state. If a belongs to H, or if it does not belong to the global history whence (n, H) emerged, then the local history relevant for φ_a in x is still (n, H). However, there is also a possibility that a belongs to the global history, but it is absent from the local support S and it had therefore been relegated to the anonymous part of the local history, i.e., to the number n. It should then be "de-anonymized". Before that happens, it is prudent to restrict the local view by intersecting it with S, to ensure that the non-anonymous part H remains small.

Similar intuitions apply to $\Diamond\varphi$, but here an additional complication appears: the supports of y and x can in general be completely unrelated, so some atoms may have to be anonymized while other ones are de-anonymized. To this end, while making a transition from the state x to y, a set D of de-anonymized atoms is chosen. There must be at most n of them, since this is the size of the anonymous part of the local history. There is no point in de-anonymizing an atom that is already in H, so D and H must be disjoint. Furthermore, if an atom is *not* in H but has been present in the current support, then the local history claims for a fact that the atom does not belong to the global history either; so D and $\mathrm{supp}(x, \varphi, \rho)$ must be disjoint too. Finally, each de-anonymized atom must be part of the new current support after the transition S, which in this situation is equivalent to belonging to $\mathrm{supp}(y)$.

Once D is chosen, a new local history is constructed as expected; note that the elements of the set $\mathrm{pred}(x)$ become part of the global history, but some of them (those that do not belong to S) immediately go into the anonymous part of the local history.

C. Global vs. local histories

We now wish to show a formal correspondence between the global and the local semantics, arguing by induction on the structure of formulas. The inductive claim turns out to be rather subtle, so we shall first explain why other, seemingly simpler approaches do not work.

For a finite set S of atoms, a local-history context

$$\rho : \mathbb{X} \rightharpoonup (\mathbb{N} \times \mathcal{P}_{\mathrm{fin}}\mathbb{A}) \to \mathcal{P}K$$

can be extended ("de-localized") to one that depends on global histories, denoted by

$$\rho{\uparrow}^{S} : \mathbb{X} \rightharpoonup \mathcal{P}_{\mathrm{fin}}\mathbb{A} \to \mathcal{P}K$$

and defined by:

$$\rho{\uparrow}^{S}(X)(H) = \rho(X)(H \curvearrowleft S).$$

Comparing local and global history semantics, at first sight it is natural to expect that for every formula φ, a local-history context ρ, a global history H and every sufficiently large set S of atoms:

$$\llparenthesis\varphi\rrparenthesis_{\rho}^{H \curvearrowleft S} = \llbracket\varphi\rrbracket_{\rho{\uparrow}^{S}}^{H} . \qquad (16)$$

More specifically, it is reasonable to expect this for $S \supseteq \mathrm{supp}(\varphi, \rho)$, since the formula φ and its context ρ form the local view on history.

Unfortunately, Equation (16) fails. For a counterexample, consider a state $x \in K$ with $\mathrm{pred}(x) = \mathrm{supp}(x) = \{a\}$ and the formula

$$\varphi = \bigvee_{a \in A} (a \wedge \sharp a)$$

with a history $H = \{a\}$ (the context ρ is irrelevant as φ is closed). Since φ is an equivariant formula, one may take $S = \varnothing$ so that $H \curvearrowleft S = (1, \varnothing)$. It is then easy to check that

$$x \in \llparenthesis\varphi\rrparenthesis_{\rho}^{H \curvearrowleft S} \qquad \text{but} \qquad x \notin \llbracket\varphi\rrbracket_{\rho{\uparrow}^{S}}^{H} .$$

Intuitively, the problem here is that we allowed S which is too small, in that it does not contain the support of the state x, which should rightly be a part of the local view on history.

To fix that, one may weaken (16) and postulate instead that

$$x \in \llparenthesis\varphi\rrparenthesis_{\rho}^{H \curvearrowleft S} \iff x \in \llbracket\varphi\rrbracket_{\rho{\uparrow}^{S}}^{H} \qquad (17)$$

whenever $S \supseteq \mathrm{supp}(x, \varphi, \rho)$. This, however, is ill-suited as a claim to prove by structural induction on φ; specifically, the inductive step for fixpoint formulas $\mu X.\varphi$ is problematic. Indeed, for the inductive step one would want to prove that whenever

$$x \in \Phi(H \curvearrowleft S) \iff x \in \Psi(H) \qquad (18)$$

for some $\Phi : (\mathbb{N} \times \mathcal{P}_{\mathrm{fin}}\mathbb{A}) \to \mathcal{P}K$ and $\Psi : \mathcal{P}_{\mathrm{fin}}\mathbb{A} \to \mathcal{P}K$ such that $S \supseteq \mathrm{supp}(x, \varphi, \rho, \Phi, \Psi)$, then also

$$x \in \llparenthesis\varphi\rrparenthesis_{\rho[X \mapsto \Phi]}^{H \curvearrowleft S} \iff x \in \llbracket\varphi\rrbracket_{\rho{\uparrow}^{S}[X \mapsto \Psi]}^{H} .$$

To use the inductive assumption (17) about φ, one would need:

$$(\rho[X \mapsto \Phi]){\uparrow}^{S} = (\rho{\uparrow}^{S})[X \mapsto \Psi],$$

but this does not follow from the assumption (18). It would follow from the stronger assumption $\Phi(H \curvearrowleft S) = \Psi(H)$, but that would amount to assuming (16), which – as we have seen – fails in general.

To find the right claim to prove by induction, we look for the middle ground between (16) and (17), by relaxing the relationship between contexts ρ and $\rho{\uparrow}^{S}$ allowed in (17).

For any $\Phi : (\mathbb{N} \times \mathcal{P}_{\mathrm{fin}}\mathbb{A}) \to \mathcal{P}K$, $\Psi : \mathcal{P}_{\mathrm{fin}}\mathbb{A} \to \mathcal{P}K$ and a finite set of atoms S, we write

$$\Phi \sim_{S} \Psi \qquad (19)$$

if and only if for every $x \in K$, $T \supseteq S \cup \mathrm{supp}(x)$ and $H \in \mathcal{P}_{\mathrm{fin}}\mathbb{A}$:

$$x \in \Phi(H \curvearrowleft T) \iff x \in \Psi(H)$$

(compare (18)). From the definition it easily follows that

$$\text{if} \qquad \Phi \sim_{S} \Psi \qquad \text{and} \qquad S \subseteq S' \qquad \text{then} \qquad \Phi \sim_{S'} \Psi.$$

We will use this observation extensively in the following.

This relation $\sim$ is extended to contexts: for $\rho : \mathbb{X} \rightharpoonup (\mathbb{N} \times \mathcal{P}_{\mathrm{fin}}\mathbb{A}) \to \mathcal{P}K$ and $\xi : \mathbb{X} \rightharpoonup \mathcal{P}_{\mathrm{fin}}\mathbb{A} \to \mathcal{P}K$ with the same domain, we write

$$\rho \approx \xi$$

if $\rho(X) \sim_{\mathrm{supp}(X,\rho)} \xi(X)$ for every X in that domain.

Theorem 4. *For every formula φ and contexts*

$$\rho : \mathbb{X} \rightharpoonup (\mathbb{N} \times \mathcal{P}_{\mathrm{fin}}\mathbb{A}) \to \mathcal{P}K$$
$$\xi : \mathbb{X} \rightharpoonup \mathcal{P}_{\mathrm{fin}}\mathbb{A} \to \mathcal{P}K$$

defined on the free variables of φ, if $\rho \approx \xi$ then

$$\llparenthesis\varphi\rrparenthesis_{\rho} \sim_{\mathrm{supp}(\varphi,\rho)} \llbracket\varphi\rrbracket_{\xi} .$$

Proof. By structural induction on φ; see Appendix B. □

Corollary 5. *For any closed formula φ, $\llparenthesis\varphi\rrparenthesis_{\varnothing}^{0,\varnothing} = \llbracket\varphi\rrbracket_{\varnothing}^{\varnothing}$.*

Proof. By Theorem 4, $\llparenthesis\varphi\rrparenthesis_{\varnothing} \sim_{\mathrm{supp}(\varphi)} \llbracket\varphi\rrbracket_{\varnothing}$, and the conclusion easily follows from the definition of $\sim$ in (19), as $\varnothing \curvearrowleft T = (0, \varnothing)$ for any T. □

D. Relevant local histories

By Corollary 5, to find the semantics of a closed formula in a given model, one may equivalently calculate its local-history semantics. This, however, does not imply yet that model checking is decidable. Indeed it looks like a step back, since in the local-history semantics fixpoint formulas are approximated by families indexed by elements of $\mathbb{N} \times \mathcal{P}_{\mathrm{fin}}\mathbb{A}$, a clearly orbit-infinite set. We shall now show how the set of indices can be limited to an essentially orbit-finite set, thus ensuring decidability.

The inductive definition of $\llparenthesis\varphi\rrparenthesis^{0,\varnothing}_{\varnothing}$ involves calculating many sets of the form $\llparenthesis\psi\rrparenthesis^{n,H}_{\rho}$ for ψ subformulas of φ. We will show that the family of all such sets (more precisely, tuples $(\psi, \rho, (n, H))$) that appear in this calculation for a fixed formula φ and a fixed orbit-finite model $\mathcal{K}$, is essentially orbit-finite. It is clear that a fixed formula φ has only an orbit-finite set of subformulas ψ, but we need to find bounds on the remaining three ingredients.

To this end, we shall need a more refined analysis of the inductive definition of the semantics $\llparenthesis\_\rrparenthesis$. For a fixed closed formula φ, define the set of *relevant* tuples $(\psi, \rho, (n, H))$ as the least set closed under the following rules:

- $(\varphi, \varnothing, (0, \varnothing))$ is relevant;
- if $(\neg\psi, \rho, (n, H))$ is relevant, then $(\psi, \rho, (n, H))$ is relevant;
- if $(\psi \vee \theta, \rho, (n, H))$ is relevant, then $(\psi, \rho, (n, H))$ and $(\theta, \rho, (n, H))$ are relevant;
- if $(\bigvee_{a\in\mathbb{A}} \psi_a, \rho, (n, H))$ is relevant, then:
  - $(\psi_a, \rho, (n, H))$ is relevant for every $a \in \mathbb{A}$ and
  - if $n > 0$, then $(\psi_a, \rho, ((n, H) \curvearrowleft S) \hookrightarrow \{a\})$ is relevant for every $a \notin H \cup S$, where S is as in the definition of $\llparenthesis\bigvee_{a\in\mathbb{A}} \psi_a\rrparenthesis_{\rho}$;
- if $(\Diamond\psi, \rho, (n, H))$ is relevant, then
  $$(\psi, \rho, ((n, H \cup \mathrm{pred}(x)) \curvearrowleft S) \hookrightarrow D)$$
  is relevant for every $x, y \in K$, and for every D and S as in the definition of $\llparenthesis\Diamond\psi\rrparenthesis_{\rho}$;
- if $(\mu X.\psi, \rho, (n, H))$ is relevant, then, for every ordinal α, the tuple $(\psi, \rho[X \mapsto \Phi^{\alpha}], (n, H))$ is relevant, where $\Phi^{\alpha} : (\mathbb{N} \times \mathcal{P}_{\mathrm{fin}}\mathbb{A}) \to \mathcal{P}K$ is defined by induction on α:
  - Φ^0 is constant at $\varnothing$,
  - $\Phi^{\alpha+1} = \llparenthesis\psi\rrparenthesis_{\rho[X\mapsto\Phi^{\alpha}]}$,
  - $\Phi^{\alpha} = \bigcup_{\beta<\alpha} \Phi^{\beta}$ for a limit ordinal α;
- if $(X, \rho, (n, H))$ is relevant, then $(\mu X.\psi, \rho', (n, H))$ is relevant, where $\mu X.\psi$ is the binding occurrence of X in φ, and ρ' is ρ restricted to the set of free variables in $\mu X.\psi$.

Looking at the definition of $\llparenthesis\_\rrparenthesis$, it is easy to see that the inductive definition of $\llparenthesis\varphi\rrparenthesis^{0,\varnothing}_{\varnothing}$ involves only those (and perhaps not all those) sets $\llparenthesis\psi\rrparenthesis^{n,H}_{\rho}$ where $(\psi, \rho, (n, H))$ is relevant for φ. For the case of fixpoint formulas, this is because

$$\llparenthesis\mu X.\psi\rrparenthesis^{n,H}_{\rho} = \llparenthesis\psi\rrparenthesis^{n,H}_{\rho[X\mapsto\Phi^{\alpha}]}$$

for some ordinal α. Moreover, in inductive computation of the rightmost expression, $\rho[X \mapsto \Phi^{\alpha}]$ is used for subformulas of ψ. As a result, for the case of variables we have

$$\llparenthesis X\rrparenthesis^{n,H}_{\rho} = \Phi^{\alpha}(n, H) = \llparenthesis\psi\rrparenthesis^{n,H}_{\rho'[X\mapsto\Phi^{\beta}]}$$

for some $\beta < \alpha$ (e.g., $\beta = \alpha - 1$ if α is not a limit ordinal), and the tuple $(\psi, \rho'[X \mapsto \Phi^{\beta}], (n, H))$ is relevant because $(\mu X.\psi, \rho', (n, H))$ is.

Our purpose now will be to show that the set of relevant tuples is "almost" orbit-finite, and that it can be usefully approximated by an orbit-finite set.

First, let us define some auxiliary measures of the complexity of formulas. Let *mu-depth*(φ) denote the maximal number of nested fixpoint operators in φ. Note that every subformula of φ has at most *mu-depth*(φ) free variables. Moreover, let *max-var-supp*(φ) denote the maximal size of the least support of a variable name in φ, and let *max-subf-supp*(φ) be the maximal size of the least support of a subformula of φ. These are all well-defined natural numbers.

Lemma 6. *In every relevant tuple $(\psi, \rho, (n, H))$, the context ρ has a support of size at most*

$$\textit{mu-depth}(\varphi) \cdot (\textit{max-var-supp}(\varphi) + \textit{max-subf-supp}(\varphi)). \quad (20)$$

Proof. See Appendix C. □

Note that the upper bound (20) on the size of a support of ρ in a relevant tuple $(\psi, \rho, (n, H))$ depends only on φ but not on ψ, n or H. Denote this upper bound by *max-ctx-supp*(φ). Furthermore, let *dim*$(\mathcal{K})$ be the *dimension* of $\mathcal{K}$, i.e., the maximal size of the least support of a state of $\mathcal{K}$.

Lemma 7. *In every relevant tuple $(\psi, \rho, (n, H))$, the history H contains at most*

$$\textit{dim}(\mathcal{K}) + \textit{max-subf-supp}(\varphi) + \textit{max-ctx-supp}(\varphi) + 1 \quad (21)$$

atoms.

Proof. See Appendix C. □

The upper bound (21) on the size of a local history H in a relevant tuple $(\psi, \rho, (n, H))$ depends only on φ and on the model $\mathcal{K}$ but not on ψ, ρ or n. Denote this upper bound by *max-loc-hist*$(\varphi, \mathcal{K})$.

Lemmas 6 and 7 imply that the only potentially unbounded component of a tuple $(\psi, \rho, (n, H))$ relevant for a fixed formula φ, is the number n, the size of the anonymous part of the local history. To deal with this final component, we shall use a relative version of the notion of relevant tuple.

Specifically, for a relevant tuple $Q = (\psi, \rho, (n, H))$ for a fixed formula φ, we say that another tuple $(\psi', \rho', (n', H'))$ is *relevant for* Q if it belongs to the least set of tuples containing Q and closed under all rules that define the set of relevant tuples except the one for $(\varphi, \varnothing, (0, \varnothing))$. Obviously, if Q is relevant then every tuple relevant for Q is itself relevant.

Lemma 8. *For every relevant tuple $Q = (\psi, \rho, (n, H))$, and for every tuple $(\psi', \rho', (n', H'))$ relevant for Q,*

$$n + |H| \leq n' + |H'|.$$

Proof. See Appendix C. □

The next lemma applies to any closed formula φ and any orbit-finite model $\mathcal{K}$. Intuitively, the lemma says that, if the anonymous part of a local history grows large enough, its precise size ceases to matter and it may be seen as an inexhaustible supply of non-fresh, anonymous atoms.

Lemma 9. *For any numbers $m_1 \geq m_2$ and any ψ, ρ, and H such that both tuples $(\psi, \rho, (m_1, H))$ and $(\psi, \rho, (m_2, H))$ are relevant, if*

$$m_2 + |H| > \textit{max-loc-hist}(\varphi, \mathcal{K}) + \textit{dim}(\mathcal{K}) \qquad (22)$$

then the two tuples are equivalent, i.e., $\llparenthesis \psi \rrparenthesis_{\rho}^{m_1,H} = \llparenthesis \psi \rrparenthesis_{\rho}^{m_2,H}$.

Proof. Relies on Lemmas 7 and 8; see Appendix C. □

E. Decidability

We are now ready to prove our main technical result:

Theorem 10. *Given a closed formula φ and a state x in an equivariant, orbit-finite model K, it is decidable whether $x \in [\![\varphi]\!]_{\varnothing}^{\varnothing}$.*

Proof. By Theorem 4, it is enough to decide whether $x \in \llparenthesis \varphi \rrparenthesis_{\varnothing}^{0,\varnothing}$. To this end, one computes the local-history semantics of φ and all its subformulas by induction, much as in Section IV-B for the history-free μ-calculus with atoms. As before, the most complex case is for a fixpoint formula $\mu X.\psi$, where a least fixpoint of an operator working over families:

$$\Phi : (\mathbb{N} \times \mathcal{P}_{\mathrm{fin}}\mathbb{A}) \to \mathcal{P}K$$

needs to be computed. By Lemma 7, these families may be safely restricted to indices (n, H) such that $|H| \leq \textit{max-loc-hist}(\varphi, \mathcal{K})$, an upper bound computable from φ and $\mathcal{K}$. Then, by Lemma 9, for any m_1, m_2 such that (22) holds (another bound computable from φ and $\mathcal{K}$), all families Φ considered satisfy

$$\Phi(m_1, H) = \Phi(m_2, H),$$

so one may restrict attention to families Φ indexed by (n, H) such that $n \leq \textit{max-loc-hist}(\varphi, \mathcal{K}) + \textit{dim}(\mathcal{K})$, augmented with one more index for each H that represents a "very large n".

Altogether, to compute $\llparenthesis \mu X.\psi \rrparenthesis_{\rho}$, one builds a series of approximating families Φ, each of them indexed by the same orbit-finite set. The set of indices is equivariant, and the families themselves have a common finite support $\mathrm{supp}(X, \psi, \rho)$. Since K is orbit-finite, this implies that there are only finitely many families Φ to consider, therefore the approximation process will reach a fixpoint after finitely many steps.

Remaining cases work as in Section IV-B. □

Our decidability proof was formulated under the assumption that $\mathcal{K}$ is equivariant. This is easy to lift: for an arbitrary orbit-finite model the proof is the same, but with $\mathrm{supp}(\mathcal{K})$ added to all the relevant supports such as $\mathrm{supp}(x, \varphi, \rho)$. Whenever a function or relation is claimed to be equivariant in the proof, it should now be supported by $\mathrm{supp}(\mathcal{K})$.

The other simplifying assumption, $\mathbb{B} = \mathbb{A}$, can be relaxed to the original (10) with a little more effort. To this end, one needs to change the definition of a local history so that, in addition to a finite subset of $\mathbb{B}$, it includes a number $n_I \in \mathbb{N}$ for each nonempty subset $I \subseteq B$ of the orbits of $\mathbb{B}$. Intuitively, such a number says that for exactly n_I anonymous atoms a, the global history contains all basic predicates (i, a) for $i \in I$ and no basic predicates (j, a) for $j \in B \setminus I$. The local-history semantics and its decidability proof becomes considerably more complex, but the essence of anonymizing atoms and finding upper bounds on local histories remains the same.

VIII. Future work

To facilitate our decidability proof, we restricted the design of the history-dependent μ-calculus in several ways. The restrictions help identify local histories as a key tool and still let us cover substantial examples, but it should be interesting to see how our techniques extend to richer scenarios.

Perhaps the easiest extension is to allow infinitely many basic predicates to hold in a state. This is particularly simple when each basic predicate is built of a single atom, since an infinite but finitely supported set of atoms must be co-finite. A cofinite set of basic predicates that hold in a single state leaves only finitely many predicates fresh, so visiting such states on a transition path makes freshness properties easy to check.

The situation becomes more complicated when arbitrary orbit-finite sets of basic predicates are considered. If a basic predicate can contain two or more atoms, then the notion of freshness itself needs to be reconsidered, with a few natural options to choose from. Further in this direction, simple freshness tests $\sharp p$ are but one of many properties of histories than one may want to check. Others include, e.g., checking whether the global history is of even size, or some regular properties such as checking that, historically, every occurrence of a particular predicate was followed by an occurrence of another one. For properties that can be restricted to local views of histories, our proof technique should be applicable.

In the forthcoming [36], an extended version of [15], a *vectorial* version of the μ-calculus with atoms is considered, where orbit-finitely many variables may be bound by a single fixpoint operator. Intuitively, this amounts to equipping fixpoint variables with atoms as parameters. Unlike in the atom-less case, the vectorial calculus with atoms is a proper extension of the "scalar" one. Adding freshness tests to the vectorial calculus is a worthwhile task, and we conjecture that model checking remains decidable for that case. The vectorial μ-calculus is closely connected to alternating tree automata and parity games, and the next challenge is to draw that connection in the presence of freshness tests. A connection to parity games should be particularly enlightening, as the idea of forgetting the past is well established in game theory in the form of various memoryless strategy theorems.

As argued in [18], the theory of sets with atoms can be formulated more generally, for an infinite relational structure $\mathbb{A}$ subject to some model-theoretic conditions. An example is the framework of *ordered atoms*, where atoms are rational numbers that can be compared not only for equality but also for order. Ordered data values arise naturally in various systems, e.g., as timestamps or as process identifiers in distributed protocols such as leader election. In [36] a μ-calculus with ordered atoms is introduced, and the question of extending it with freshness tests remains open. One difficulty there is that for ordered or otherwise structured atoms, the concept of freshness becomes less clear.

References

[1] D. Kozen, “Results on the propositional μ-calculus,” *Theor. Comp. Sci.*, vol. 27, no. 3, pp. 333 – 354, 1983.

[2] E. A. Emerson and J. Y. Halpern, ““sometimes” and “not never” revisited: On branching versus linear time temporal logic,” *J. ACM*, vol. 33, no. 1, pp. 151–178, 1986.

[3] L. Segoufin, “Automata and logics for words and trees over an infinite alphabet,” in *Procs. CSL 2006*, ser. Lecture Notes in Computer Science, vol. 4207, 2006, pp. 41–57.

[4] S. Demri, R. Lazic, and D. Nowak, “On the freeze quantifier in constraint LTL: decidability and complexity,” *Inf. Comput.*, vol. 205, no. 1, pp. 2–24, 2007.

[5] S. Demri and R. Lazic, “LTL with the freeze quantifier and register automata,” *ACM Trans. Comput. Log.*, vol. 10, no. 3, pp. 16:1–16:30, 2009.

[6] D. Figueira and L. Segoufin, “Future-looking logics on data words and trees,” in *Procs. MFCS 2009*, ser. Lecture Notes in Computer Science, vol. 5734, 2009, pp. 331–343.

[7] S. Demri, D. D’Souza, and R. Gascon, “Temporal logics of repeating values,” *J. Log. Comput.*, vol. 22, no. 5, pp. 1059–1096, 2012.

[8] S. Demri, D. Figueira, and M. Praveen, “Reasoning about data repetitions with counter systems,” *Logical Methods in Computer Science*, vol. 12, no. 3, 2016.

[9] S. Demri and D. D’Souza, “An automata-theoretic approach to constraint LTL,” *Inf. Comput.*, vol. 205, no. 3, pp. 380–415, 2007.

[10] C. Carapelle and M. Lohrey, “Temporal logics with local constraints (invited talk),” in *Procs. CSL 2015*, ser. LIPIcs, vol. 41, 2015, pp. 2–13.

[11] D. Figueira, “Alternating register automata on finite words and trees,” *Logical Methods in Computer Science*, vol. 8, no. 1, 2012.

[12] M. Jurdzinski and R. Lazic, “Alternating automata on data trees and xpath satisfiability,” *ACM Trans. Comput. Log.*, vol. 12, no. 3, pp. 19:1–19:21, 2011.

[13] J. F. Groote and R. Mateescu, “Verification of temporal properties of processes in a setting with data,” in *Procs. AMAST’99*, 1999, pp. 74–90.

[14] J. F. Groote and T. A. Willemse, “Model-checking processes with data,” *Science of Computer Programming*, vol. 56, no. 3, pp. 251–273, 2005.

[15] B. Klin and M. Łełyk, “Modal μ-Calculus with Atoms,” in *Procs. CSL 2017*, ser. LIPIcs, vol. 82, 2017, pp. 30:1–30:21.

[16] A. M. Pitts, *Nominal Sets: Names and Symmetry in Computer Science*. Cambridge University Press, 2013.

[17] M. Gabbay and A. M. Pitts, “A new approach to abstract syntax with variable binding,” *Formal Asp. Comput.*, vol. 13, no. 3-5, pp. 341–363, 2002.

[18] M. Bojańczyk, B. Klin, and S. Lasota, “Automata theory in nominal sets,” *Log. Meth. Comp. Sci.*, vol. 10, 2014.

[19] B. Klin, E. Kopczyński, J. Ochremiak, and S. Toruńczyk, “Locally finite constraint satisfaction problems,” in *Procs. LICS 2015*, 2015, pp. 475–486.

[20] N. Tzevelekos, “Fresh-register automata,” in *ACM SIGPLAN Notices*, vol. 46, no. 1. ACM, 2011, pp. 295–306.

[21] R. M. Needham and M. D. Schroeder, “Using encryption for authentication in large networks of computers,” *Communications of the ACM*, vol. 21, no. 12, pp. 993–999, 1978.

[22] A. Arnold and D. Niwiński, *Rudiments of μ-calculus*, ser. Studies in logic and the foundations of mathematics. Amsterdam: London, 2001.

[23] J. Bradfield and C. Stirling, “Modal logics and mu-calculi: an introduction,” in *Handbook of Process Algebra*. North-Holland, 2001, pp. 293–330.

[24] Y. Venema, *Lectures on the modal μ-calculus*. ILLC, Univ. of Amsterdam, 2007.

[25] J. Bradfield and C.Stirling, “Modal mu-calculi,” in *Handbook of Modal Logic*. Elsevier, 2007, vol. 3, pp. 721 – 756.

[26] J. Bradfield and I. Walukiewicz, “The mu-calculus and model-checking,” in *Handbook of Model Checking*, H. V. E. Clarke, T. Henzinger, Ed. Springer-Verlag, 2015.

[27] C. S. Calude, S. Jain, B. Khoussainov, W. Li, and F. Stephan, “Deciding parity games in quasipolynomial time,” in *Procs. 49th STOC*. ACM, 2017, pp. 252–263.

[28] J. Ochremiak, “Extended constraint satisfaction problems,” Ph.D. dissertation, University of Warsaw, 2016.

[29] L. J. Stockmeyer and A. R. Meyer, “Word problems requiring exponential time (preliminary report),” in *Procs. STOC’73*, 1973, pp. 1–9.

[30] B. Klin and M. Szynwelski, “SMT solving for functional programming over infinite structures,” in *MFSP*, vol. 207, 2016, pp. 57–75.

[31] E. Kopczyński and S. Toruńczyk, “Lois: Syntax and semantics,” in *Procs. of POPL 2017*, 2017, pp. 586–598.

[32] M. Fernández and M. J. Gabbay, “Nominal rewriting,” *Information and Computation*, vol. 205, no. 6, pp. 917–965, 2007.

[33] F. Neven, T. Schwentick, and V. Vianu, “Towards regular languages over infinite alphabets,” in *MFCS*, 2001, pp. 560–572.

[34] B. Klin, S. Lasota, J. Ochremiak, and S. Toruńczyk, “Homomorphism problems for first-order definable structures,” in *Procs. FSTTCS’16*, ser. LIPIcs, vol. 65, 2016, pp. 14:1–14:15.

[35] G. Lowe, “Breaking and fixing the Needham–Schroeder public-key protocol using FDR,” in *International Workshop on Tools and Algorithms for the Construction and Analysis of Systems*. Springer, 1996, pp. 147–166.

[36] B. Klin and M. Łełyk, “Scalar and vectorial μ-calculus with atoms,” 2019, to appear. Available from https://arxiv.org/abs/1803.06752.

Appendix A
Syntactic variants of orbit-finite disjunction

Our (9) is not the only way to introduce orbit-finitary disjunction into the μ-calculus. Several equivalent alternatives exist, each with its own advantages.

In [15], a simple syntax

$$\varphi ::= \cdots \mid \bigvee \Phi \tag{23}$$

was used, where Φ is a finitely supported, orbit-finite set of formulas, with semantics defined by:

$$x \in [\![\bigvee \Phi]\!]_\xi \iff x \in [\![\varphi]\!]_\xi \text{ for some } \varphi \in \Phi.$$

One can then dispense with the finite disjunction $\varphi \vee \psi$, as it becomes a special case. It is easy to subsume (9) with this syntax: for $\bigvee_{a \in \mathbb{A}} \varphi_a$ one may write $\bigvee \Phi$ where Φ is the range (i.e., direct image) of the function $\varphi_{(-)}$.

An encoding in the other direction is a little more complex. First, consider $\bigvee \Phi$ where Φ consists of a single S-orbit for $S = \mathrm{supp}(\Phi)$. Pick any formula $\varphi \in \Phi$ and let

$$T = \mathrm{supp}(\varphi) \setminus S = \{b_1, \ldots, b_n\}.$$

Then instead of $\bigvee \Phi$ one may write

$$\bigvee_{a_1 \in \mathbb{A}} \bigvee_{a_2 \in \mathbb{A}} \cdots \bigvee_{a_n \in \mathbb{A}} \psi_{a_1, \ldots, a_n}$$

with the function ψ from $\mathbb{A}^n$ to the set of formulas defined by:

$$\psi(a_1, \ldots, a_n) = \begin{cases} \varphi \cdot \pi_{\bar{b}\bar{a}} & \text{if } a_1 \neq \cdots \neq a_n \notin S \\ \bot & \text{otherwise} \end{cases},$$

where $\pi_{\bar{b}\bar{a}}$ denotes any atom automorphism that maps b_i to a_i for $i = 1..n$ and fixes every atom in S. Finally, for Φ with more than orbit, one can encode each of the orbits separately and combine the formulas thus obtained with finite disjunction.

In this paper we choose the syntax (9) over (23) because it enforces tighter control on the supports of disjuncts in an orbit-finite disjunction, which will come handy for our proofs. Specifically, for $\varphi = \bigvee_{a \in \mathbb{A}} \varphi_a$ we have

$$\mathrm{supp}(\varphi_a) \subseteq \mathrm{supp}(\varphi) \cup \{a\} \qquad \text{for every } a \in \mathbb{A},$$

whereas for $\varphi = \bigvee \Phi$, the least support of a formula $\psi \in \Phi$ may considerably differ from $\mathrm{supp}(\varphi)$. Another reason is that in practical examples, such as (8), orbit-finite operators tend to be of the form prescribed by (9).

Actually, in most examples, orbit-finite disjunctions and conjunctions have an even simpler form, where in $\bigvee_{a \in \mathbb{A}} \varphi_a$ all the formulas φ_a have the same shape; formally, they belong to a single orbit. Indeed, one could replace the syntax (9) with the following variant:

$$\varphi ::= \cdots \mid \bigvee_{a \notin S} \varphi_a \tag{24}$$

where S is a finite set of atoms and $\varphi_{(-)}$ is a function, supported by S, from $\mathbb{A} \setminus S$ to the set of formulas. It then follows that the range of $\varphi_{(-)}$ is a single S-orbit. In the presence of finite disjunction this syntax is easily equivalent to (9). Indeed, for $\bigvee_{a \notin S} \varphi_a$ one may write $\bigvee_{a \in \mathbb{A}} \psi_a$ where

$$\psi_a = \begin{cases} \varphi_a & \text{if } a \notin S \\ \bot & \text{otherwise} \end{cases}.$$

In the other direction, for $\bigvee_{a \in \mathbb{A}} \varphi_a$, putting $S = \{a_1, \ldots, a_n\} = \mathrm{supp}(\varphi_{(-)})$, one may write

$$\varphi_{a_1} \vee \cdots \vee \varphi_{a_n} \vee \bigvee_{a \notin S} \varphi_a.$$

The syntax (24) is quite convenient in practice, as it lets one write formulas such as $\bigvee_a \bigvee_{a \neq b} \bigvee_{c \notin \{a,b\}} \varphi_{a,b,c}$, guaranteeing the atoms a, b, c to be distinct in $\varphi_{a,b,c}$.

One could even simplify (24) to write:

$$\varphi ::= \cdots \mid \exists a.\varphi \tag{25}$$

where $a \in \mathbb{A}$, with $\exists a.\varphi$ standing for $\bigvee_{b \notin \mathrm{supp}(\varphi) \setminus \{a\}} \psi_b$, where $\psi_b = \varphi \cdot (a\ b)$. This syntax is particularly appealing, as formulas over it appear as finite objects. It also represents the useful intuition that orbit-finite disjunction introduces an atom to the scope of a formula, quantifying it existentially.

For all its simplicity, the syntax (25) may be tricky to work with for a reader who has not much experience in nominal techniques. For example, it would be wrong to see a formula $\exists a.\varphi$ as a formal pair built of a and φ. Indeed, the atom a belongs to the support of such a pair, but the intention is that $a \notin \mathrm{supp}(\exists a.\varphi)$. Rather, $\exists a.\varphi$ belongs to a so-called abstraction space $[\mathbb{A}]\mathcal{F}$, where $\mathcal{F}$ is a set of formulas. The reader familiar with the book [16] would have no difficulty in following this approach, but we prefer to avoid the need to formally introduce abstraction spaces and α-conversion, and we stick to the more elementary (albeit not finitary) syntax (9).

Appendix B
Proof of Theorem 4

For technical convenience, we shall assume that in every fixpoint formula $\mu X.\varphi$,

$$\mathrm{supp}(\varphi) \subseteq \mathrm{supp}(X). \tag{26}$$

This is not true for a μ-formula in general; consider for example

$$\mu X.(a \vee \Diamond X)$$

for some fixed $a \in \mathbb{A}$ and an equivariant variable name X. However, every formula can again be transformed to an equivalent one with this property, by renaming variables. Indeed, the variable X in $\mu X.\varphi$ can be replaced with one with the name $X^{a_1, \ldots, a_n}$, where $(a_1, \ldots, a_n)$ is any fixed enumeration of $\mathrm{supp}(\varphi)$. For example, the formula above becomes

$$\mu X^a.(a \vee \Diamond X^a),$$

which satisfies the assumption and is clearly semantically equivalent both under the global- and the local-history semantics.

We shall need a few simple observations on the anonymisation and de-anonymisation operators. All of the following are easy to prove from the definitions (14) and (15).

For all local histories (n, H) and sets S, T of atoms, we have:

$$((n, H) \curvearrowleft S) \curvearrowleft T = (n, H) \curvearrowleft (S \cap T), \tag{27}$$

if $T \subseteq H$ then:

$$((n, H) \curvearrowleft S) \hookrightarrow T = (n, H) \curvearrowleft (S \cup T), \tag{28}$$

if $T \cap H \subseteq S$ then:

$$(n, H) \curvearrowleft S = (n, H) \curvearrowleft (S \cup T), \tag{29}$$

and finally, if $(n', H') = (n, H) \curvearrowleft S$, and $|T| \leq n$ and $(n'', H'') = (n, H) \hookrightarrow T$ then:

$$n' + |H'| = n + |H| = n'' + |H''|. \tag{30}$$

The proof of Theorem 4 proceeds by induction on the structure of φ. There are nine cases to consider:

- $\varphi = \top$ is immediate.
- $\varphi = a$: check
$$x \in \llparenthesis a \rrparenthesis_{\rho}^{H \curvearrowleft T} \iff a \in \mathrm{pred}(x) \iff x \in \llbracket a \rrbracket_{\xi}^{H}.$$
- $\varphi = \sharp a$: the assumption that $T \supseteq \mathrm{supp}(\varphi) \ni a$ ensures the middle equivalence in:
$$x \in \llparenthesis \sharp a \rrparenthesis_{\rho}^{H \curvearrowleft T} \iff a \notin H \cap T \iff a \notin H \iff x \in \llbracket \sharp a \rrbracket_{\xi}^{H}.$$
- $\varphi = \neg\psi$ is immediate.
- $\varphi = \psi \vee \theta$: by the inductive assumption,
$$\llparenthesis \psi \rrparenthesis_{\rho} \sim_{\mathrm{supp}(\psi,\rho)} \llbracket \psi \rrbracket_{\xi} \qquad \text{and} \qquad \llparenthesis \theta \rrparenthesis_{\rho} \sim_{\mathrm{supp}(\theta,\rho)} \llbracket \theta \rrbracket_{\xi}.$$
Assume that $x \in \llparenthesis \varphi \rrparenthesis_{\rho}^{H \curvearrowleft T}$ for $T \supseteq \mathrm{supp}(x, \varphi, \rho)$; without loss of generality, let $x \in \llparenthesis \psi \rrparenthesis_{\rho}^{H \curvearrowleft T}$. Since $\mathrm{supp}(\psi) \subseteq \mathrm{supp}(\varphi)$, we have $T \supseteq \mathrm{supp}(x, \psi, \rho)$ and we may use the inductive assumption to conclude that $x \in \llbracket \psi \rrbracket_{\xi}^{H}$, therefore $x \in \llbracket \varphi \rrbracket_{\xi}^{H}$ as required. Other cases are similar.
- $\varphi = X$: the assumption that $\rho \approx \xi$, hence $\rho(X) \sim_{\mathrm{supp}(X,\rho)} \xi(X)$, ensures the middle equivalence in:
$$\begin{aligned} x \in \llparenthesis X \rrparenthesis_{\rho}^{H \curvearrowleft T} &\iff x \in \rho(X)(H \curvearrowleft T) \\ &\iff x \in \xi(X)(H) \iff x \in \llbracket X \rrbracket_{\xi}^{H}. \end{aligned}$$
- $\varphi = \mu X.\psi$: the semantics $\llbracket \mu X.\psi \rrbracket_{\xi}$ is the limit of the increasing transfinite sequence:
$$\Psi^0 \subseteq \Psi^1 \subseteq \Psi^2 \subseteq \cdots \subseteq \Psi^{\omega} \subseteq \Psi^{\omega+1} \subseteq \cdots$$
where $\Psi^{\alpha} : \mathcal{P}_{\mathrm{fin}}\mathbb{A} \to \mathcal{P}K$ (with the inclusion order above understood pointwise) are defined by ordinal induction, and in particular
$$\Psi^{\alpha+1}(H) = \llbracket \psi \rrbracket_{\xi[X \mapsto \Psi^{\alpha}]}^{H}.$$
Similarly, $\llparenthesis \mu X.\psi \rrparenthesis_{\rho}$ is the limit of:
$$\Phi^0 \subseteq \Phi^1 \subseteq \Phi^2 \subseteq \cdots \subseteq \Phi^{\omega} \subseteq \Phi^{\omega+1} \subseteq \cdots$$
with $\Phi^{\alpha} : (\mathbb{N} \times \mathcal{P}_{\mathrm{fin}}\mathbb{A}) \to \mathcal{P}K$ and in particular:
$$\Phi^{\alpha+1}(n, H) = \llparenthesis \psi \rrparenthesis_{\rho[X \mapsto \Phi^{\alpha}]}^{n,H}$$
It is enough to prove that for every ordinal α:
$$\Phi^{\alpha} \sim_{\mathrm{supp}(\varphi,\rho)} \Psi^{\alpha}. \tag{31}$$
In the ordinal induction, the case of α a limit ordinal is easy. For a successor ordinal $\alpha + 1$ we need to show, under the assumption (31), that
$$\llparenthesis \psi \rrparenthesis_{\rho[X \mapsto \Phi^{\alpha}]} \sim_{\mathrm{supp}(\varphi,\rho)} \llbracket \psi \rrbracket_{\xi[X \mapsto \Psi^{\alpha}]}. \tag{32}$$
Aiming to use the inductive assumption about ψ, we first prove that
$$\rho[X \mapsto \Phi^{\alpha}] \approx \xi[X \mapsto \Psi^{\alpha}], \tag{33}$$
or, in other words, that for all $Y \in \mathsf{FV}(\psi)$
$$\rho[X \mapsto \Phi^{\alpha}](Y) \sim_{\mathrm{supp}(Y,\rho[X \mapsto \Phi^{\alpha}])} \xi[X \mapsto \Psi^{\alpha}](Y).$$
For $Y \neq X$, this follows from $\rho \approx \xi$. Indeed, as $\rho(Y) \sim_{\mathrm{supp}(Y,\rho)} \xi(Y)$ and $\mathrm{supp}(\rho) \subseteq \mathrm{supp}(\rho[X \mapsto \Phi^{\alpha}])$, we have
$$\rho[X \mapsto \Phi^{\alpha}](Y) = \rho(Y) \sim_{\mathrm{supp}(Y,\rho[X \mapsto \Phi^{\alpha}])} \xi(Y) = \xi[X \mapsto \Psi^{\alpha}](Y).$$
For $Y = X$, we rely on our syntactic convention (26) which guarantees that
$$\mathrm{supp}(\varphi, \rho) \subseteq \mathrm{supp}(X, \rho[X \mapsto \Phi^{\alpha}]).$$
Using this inclusion together with (31), calculate
$$\rho[X \mapsto \Phi^{\alpha}](X) = \Phi^{\alpha} \sim_{\mathrm{supp}(X,\rho[X \mapsto \Phi^{\alpha}])} \Psi^{\alpha} = \xi[X \mapsto \Psi^{\alpha}](X),$$
concluding the proof of (33). We may therefore apply the inductive assumption about ψ to conclude
$$\llparenthesis \psi \rrparenthesis_{\rho[X \mapsto \Phi^{\alpha}]} \sim_{\mathrm{supp}(\psi,\rho[X \mapsto \Phi^{\alpha}])} \llbracket \psi \rrbracket_{\xi[X \mapsto \Psi^{\alpha}]}.$$
To infer (32) from this, it is enough to show that
$$\mathrm{supp}(\psi, \rho[X \mapsto \Phi^{\alpha}]) \subseteq \mathrm{supp}(\varphi, \rho).$$
Obviously though, $\mathrm{supp}(\psi, \rho, X) = \mathrm{supp}(\varphi, \rho)$, and
$$\mathrm{supp}(\Phi^{\alpha}) \subseteq \mathrm{supp}(\psi, X, \rho)$$
follows by easy ordinal induction on α.
- $\varphi = \bigvee_{a \in \mathbb{A}} \varphi_a$: for any $H \in \mathcal{P}_{\mathrm{fin}}\mathbb{A}$, $x \in K$ and $T \supseteq \mathrm{supp}(x, \varphi, \rho)$, assuming that $\rho \approx \xi$, we must show
$$x \in \llparenthesis \varphi \rrparenthesis_{\rho}^{H \curvearrowleft T} \iff x \in \llbracket \varphi \rrbracket_{\xi}^{H}. \tag{34}$$
We begin with the left-to-right implication. Unfolding the definition of $\llparenthesis \varphi \rrparenthesis$, consider three cases:
  (a) There exists some $a \notin H$ or $a \in T$ such that $x \in \llparenthesis \varphi_a \rrparenthesis_{\rho}^{H \curvearrowleft T}$. Then it is easy to check that $H \curvearrowleft T = H \curvearrowleft (T \cup \{a\})$. Moreover, $T \cup \{a\} \supseteq \mathrm{supp}(x, \varphi_a, \rho)$, so we may apply the inductive assumption about φ_a to conclude that
$$\begin{aligned} x \in \llparenthesis \varphi_a \rrparenthesis_{\rho}^{H \curvearrowleft T} &\iff x \in \llparenthesis \varphi_a \rrparenthesis_{\rho}^{H \curvearrowleft (T \cup \{a\})} \\ &\iff x \in \llbracket \varphi_a \rrbracket_{\xi}^{H} \implies x \in \llbracket \varphi \rrbracket_{\xi}^{H}. \end{aligned}$$
  (b) There exists some $a \in H \setminus T$ such that $x \in \llparenthesis \varphi_a \rrparenthesis_{\rho}^{H \curvearrowleft T}$. Then pick any $b \notin H \cup T$ and let $\pi = (a\ b)$ be the atom

automorphism that swaps a and b. Since $a, b \notin T$ we have that:

$$x \cdot \pi = x, \qquad \varphi \cdot \pi = \varphi, \qquad \rho \cdot \pi = \rho. \tag{35}$$

Moreover, we have $(H \cdot \pi) \curvearrowleft T = H \curvearrowleft T$, hence

$$x \in \llparenthesis \varphi_a \rrparenthesis_\rho^{(H\cdot\pi)\curvearrowleft T}.$$

The relation $\approx$ is equivariant, so we have $\rho = \rho \cdot \pi \approx \xi \cdot \pi$. Since $a \notin H \cdot \pi$, by case (a) above we obtain

$$x \in \llbracket \varphi \rrbracket_{\xi\cdot\pi}^{H\cdot\pi}$$

and, applying π to everything involved, by equivariance and by (35) we get $x \in \llbracket \varphi \rrbracket_\xi^H$ as required.

(c) $|H \smallsetminus T| > 0$ and (denoting $S = \mathrm{supp}(x, \varphi, \rho)$ from now on) there exists some

$$a \notin (H \cap T) \cup S$$

such that

$$x \in \llparenthesis \varphi_a \rrparenthesis_\rho^{((H\curvearrowleft T)\curvearrowleft S)\hookrightarrow\{a\}}. \tag{36}$$

First, we notice that

$$\begin{aligned} &((H \curvearrowleft T) \curvearrowleft S) \hookrightarrow \{a\} \\ &= (H \curvearrowleft (T \cap S)) \hookrightarrow \{a\} && \text{by (27)} \\ &= (H \curvearrowleft S) \hookrightarrow \{a\} && \text{because } S \subseteq T. \end{aligned}$$

Then pick any $b \in H \smallsetminus T$ (it exists since $|H \smallsetminus T| > 0$) and let $\pi = (a\ b)$ be the atom automorphism that swaps a and b (note that it may well be that $a = b$ and $\pi = id_{\mathbb{A}}$). Since $a \notin S$ and $b \notin T$, the identities in (35) hold. Moreover,

$$\begin{aligned} &((H \curvearrowleft S) \hookrightarrow \{a\}) \cdot \pi \\ &= (H \curvearrowleft S) \cdot \pi \hookrightarrow \{a\} \cdot \pi && \text{by equivariance} \\ &= (H \curvearrowleft S) \hookrightarrow \{b\} && \text{because } a, b \notin H \cap T \supseteq H \cap S \\ &= H \curvearrowleft (S \cup \{b\}) && \text{by (28), because } b \in H, \end{aligned}$$

so, applying π to everything in (36), by equivariance and using (35), we obtain

$$x \in \llparenthesis \varphi_a \cdot \pi \rrparenthesis_\rho^{H\curvearrowleft(S\cup\{b\})}.$$

By definition of S we have $S \cup \{a\} \supseteq \mathrm{supp}(x, \varphi_a, \rho)$, which implies that $S \cup \{b\} \supseteq \mathrm{supp}(x, \varphi_a \cdot \pi, \rho)$. Thus we may apply the inductive assumption about $\varphi_a \cdot \pi$ to conclude that

$$x \in \llbracket \varphi_a \cdot \pi \rrbracket_\xi^H.$$

Applying π to everything here, by equivariance and using (35) and $\pi^{-1} = \pi$, we obtain

$$x \in \llbracket \varphi_a \rrbracket_{\xi\cdot\pi}^{H\cdot\pi}, \qquad \text{hence} \qquad x \in \llbracket \varphi \rrbracket_{\xi\cdot\pi}^{H\cdot\pi}.$$

Again applying π everywhere and using (35), we conclude that $x \in \llbracket \varphi \rrbracket_\xi^H$ as required.

We now move to the right-to-left implication of (34). Assume that there exists some $a \in \mathbb{A}$ such that $x \in \llbracket \varphi_a \rrbracket_\xi^H$. There are two cases to consider:

(a) $a \notin H$ or $a \in T$. Then it is easy to check that $H \curvearrowleft T = H \curvearrowleft (T \cup \{a\})$. Moreover, $T \cup \{a\} \supseteq \mathrm{supp}(x, \varphi_a, \rho)$, so we may use the inductive assumption about φ_a to conclude that

$$x \in \llparenthesis \varphi_a \rrparenthesis_\rho^{H\curvearrowleft(T\cup\{a\})} = \llparenthesis \varphi_a \rrparenthesis_\rho^{H\curvearrowleft T}.$$

(b) $a \in H \smallsetminus T$. Then obviously $|H \smallsetminus T| > 0$. Moreover, since $T \supseteq S = \mathrm{supp}(x, \varphi, \rho)$, we get that

$$a \notin (H \cap T) \cup S,$$

so by definition of $\llparenthesis \varphi \rrparenthesis$ the only thing that remains to be checked is:

$$x \in \llparenthesis \varphi_a \rrparenthesis_\rho^{((H\curvearrowleft T)\curvearrowleft S)\hookrightarrow\{a\}}.$$

But, by the same reasoning as in (c) above and using (28):

$$((H \curvearrowleft T) \curvearrowleft S) \hookrightarrow \{a\} = (H \curvearrowleft S) \hookrightarrow \{a\} = H \curvearrowleft (S \cup \{a\}).$$

Since $S \cup \{a\} \supseteq \mathrm{supp}(x, \varphi_a, \rho)$, we may use the inductive assumption about φ_a to conclude that $x \in \llparenthesis \varphi_a \rrparenthesis_\rho^{H\curvearrowleft(S\cup\{a\})}$ as required.

- $\varphi = \Diamond\psi$: for any $H \in \mathcal{P}_{\text{fin}}\mathbb{A}$, $x \in K$ and $T \supseteq \mathrm{supp}(x, \varphi, \rho)$, assuming that $\rho \approx \xi$, we must show

$$x \in \llparenthesis \Diamond\psi \rrparenthesis_\rho^{H\curvearrowleft T} \iff x \in \llbracket \Diamond\psi \rrbracket_\xi^H. \tag{37}$$

Note that $\mathrm{supp}(\varphi) = \mathrm{supp}(\psi)$; we shall use these two interchangeably without further warnings.

We begin with the left-to-right implication. Unfolding the definition of $\llparenthesis \varphi \rrparenthesis$, assume that $x \longrightarrow y$ and $D \subseteq \mathbb{A}$ such that:

(i) $D \subseteq \mathrm{supp}(y) \smallsetminus ((H \cap T) \cup \mathrm{supp}(x, \varphi, \rho))$,

(ii) $|D| \leq |H \smallsetminus T|$, and

(iii) $y \in \llparenthesis \psi \rrparenthesis_\rho^{(((H\curvearrowleft T)\cup\mathrm{pred}(x))\curvearrowleft S)\hookrightarrow D}$, where $S = \mathrm{supp}(y, \varphi, \rho)$ and $(n, H) \cup X$ stands for $(n, H \cup X)$.

Pick any $E \subseteq H \smallsetminus T$ such that $|E| = |D|$ (it exists by assumption (ii)), and let π be an atom automorphism such that:

(iv) $D \cdot \pi = E$,

(v) $(\mathrm{supp}(y) \smallsetminus ((H \cap T) \cup \mathrm{supp}(x, \varphi, \rho) \cup D)) \cdot \pi$ is disjoint from $H \cup \mathrm{pred}(x)$,

(vi) $\pi(a) = a$ for $a \in (H \cap T) \cup \mathrm{supp}(x, \varphi, \rho)$.

Such an automorphism exists since conditions (iv)-(vi) restrict its action only on three pairwise disjoint sets of atoms, and they allow mapping them to other pairwise disjoint sets. In particular, we have

$$x \cdot \pi = x, \qquad \psi \cdot \pi = \psi, \qquad \rho \cdot \pi = \rho. \tag{38}$$

The transition relation is equivariant, so $x = x \cdot \pi \longrightarrow y \cdot \pi$ is a valid transition. Moreover, applying π to everything in assumption (iii), by equivariance we obtain

$$y \cdot \pi \in \llparenthesis \psi \rrparenthesis_\rho^{((((H\curvearrowleft T)\cup\mathrm{pred}(x))\curvearrowleft S)\hookrightarrow D)\cdot\pi}. \tag{39}$$

But, because $\mathrm{pred}(x) \subseteq \mathrm{supp}(x) \subseteq T$ and π fixes every atom in $(H \cup \mathrm{pred}(x)) \cap T$ by (vi),

$$\begin{aligned}
&((((H \curvearrowleft T) \cup \mathrm{pred}(x)) \curvearrowleft S) \curvearrowright D) \cdot \pi \\
&= ((((H \cup \mathrm{pred}(x)) \curvearrowleft T) \curvearrowleft S) \curvearrowright D) \cdot \pi \\
&= ((((H \cup \mathrm{pred}(x)) \curvearrowleft T) \curvearrowleft S \cdot \pi) \curvearrowright E) \\
&= ((H \cup \mathrm{pred}(x)) \curvearrowleft (T \cap S \cdot \pi)) \curvearrowright E && \text{by (27)} \\
&= (H \cup \mathrm{pred}(x)) \curvearrowleft ((T \cap S \cdot \pi) \cup E) && \text{by (28).}
\end{aligned}$$

We will now show that this is equal to

$$(H \cup \mathrm{pred}(x)) \curvearrowleft ((T \cap S \cdot \pi) \cup E \cup S \cdot \pi). \qquad (40)$$

By (29), it suffices to show that

$$(H \cup \mathrm{pred}(x)) \cap S \cdot \pi \subseteq (T \cap S \cdot \pi) \cup E.$$

For all $a \in (H \cup \mathrm{pred}(x)) \cap S \cdot \pi$, let us show that $a \in (T \cap S \cdot \pi) \cup E$:

- if $a \in E$, we conclude directly,
- if $a \in (H \cap T) \cup \mathrm{supp}(x, \varphi, \rho)$, then we conclude by $\mathrm{supp}(x, \varphi, \rho) \subseteq T$,
- otherwise $a \in S \cdot \pi \setminus ((H \cap T) \cup \mathrm{supp}(x, \varphi, \rho) \cup E)$, but

  $$\begin{aligned}
  &S \cdot \pi \setminus ((H \cap T) \cup \mathrm{supp}(x, \varphi, \rho) \cup E) \\
  &= S \cdot \pi \setminus ((H \cap T) \cup \mathrm{supp}(x, \varphi, \rho) \cup D) \cdot \pi \\
  &= (\mathrm{supp}(y) \setminus ((H \cap T) \cup \mathrm{supp}(x, \varphi, \rho) \cup D)) \cdot \pi
  \end{aligned}$$

  by (iv), (vi), and the definition of S. But, by (v), this is disjoint from $H \cup \mathrm{pred}(x)$, so there is no such a.

By (40), (39) can be rewritten as

$$y \cdot \pi \in \llparenthesis \psi \rrparenthesis_{\rho}^{(H \cup \mathrm{pred}(x)) \curvearrowleft U},$$

for $U = (T \cap S \cdot \pi) \cup E \cup S \cdot \pi$. Then $U \supseteq S \cdot \pi = \mathrm{supp}(y \cdot \pi, \psi, \rho)$ by definition and (38), so we may use the inductive assumption about ψ and $y \cdot \pi$ to conclude

$$y \cdot \pi \in \llbracket \psi \rrbracket_{\xi}^{H \cup \mathrm{pred}(x)}$$

therefore $x \in \llbracket \varphi \rrbracket_{\xi}^{H}$ as required.

We now turn attention to the right-to-left implication of (37). Assume that $x \longrightarrow y$ for some $y \in \llbracket \psi \rrbracket_{\xi}^{H \cup \mathrm{pred}(x)}$. Define $S = \mathrm{supp}(y, \varphi, \rho)$; by the inductive assumption about ψ we have

$$y \in \llparenthesis \psi \rrparenthesis_{\rho}^{(H \cup \mathrm{pred}(x)) \curvearrowleft S}.$$

We want to show that $x \in \llparenthesis \Diamond \psi \rrparenthesis_{\rho}^{H \curvearrowleft T}$. To this end we shall choose a set D as prescribed by the definition of $\llparenthesis \Diamond \psi \rrparenthesis$, so that

$$(H \cup \mathrm{pred}(x)) \curvearrowleft S = (((H \curvearrowleft T) \cup \mathrm{pred}(x)) \curvearrowleft S) \curvearrowright D. \quad (41)$$

Define

$$D = (H \setminus T) \cap S.$$

Obviously $|D| \le |H \setminus T|$. Moreover, since D is disjoint from T, it is also disjoint from

$$(H \cap T) \cup \mathrm{supp}(x, \varphi, \rho),$$

and since it is contained in S and (as we just have said) disjoint from $\mathrm{supp}(\varphi, \rho)$, it is contained in $\mathrm{supp}(y)$. Altogether,

$$D \subseteq \mathrm{supp}(y) \setminus ((H \cap T) \cup \mathrm{supp}(x, \varphi, \rho)),$$

so the first two conditions on D imposed by the definition of $\llparenthesis \Diamond \psi \rrparenthesis$ are satisfied. Finally,

$$\begin{aligned}
&(((H \curvearrowleft T) \cup \mathrm{pred}(x)) \curvearrowleft S) \curvearrowright D \\
&= (((H \cup \mathrm{pred}(x)) \curvearrowleft T) \curvearrowleft S) \curvearrowright D && \mathrm{pred}(x) \subseteq T \\
&= ((H \cup \mathrm{pred}(x)) \curvearrowleft (T \cap S)) \curvearrowright D && \text{by (27)} \\
&= (H \cup \mathrm{pred}(x)) \curvearrowleft ((T \cap S) \cup D) && \text{by (28)} \\
&= (H \cup \mathrm{pred}(x)) \curvearrowleft ((T \cup H) \cap S) && \text{definition of } D \\
&= (H \cup \mathrm{pred}(x)) \curvearrowleft S, && \text{by (29).}
\end{aligned}$$

For the last equation, it suffices to show that

$$(H \cup \mathrm{pred}(x)) \cap S \subseteq T \cup H,$$

which follows directly from $\mathrm{pred}(x) \subseteq T$.

This completes the proof of (41), which implies that $y \in \llparenthesis \psi \rrparenthesis_{\rho}^{(((H \curvearrowleft T) \cup \mathrm{pred}(x)) \curvearrowleft S) \curvearrowright D}$, hence $x \in \llparenthesis \Diamond \psi \rrparenthesis_{\rho}^{H \curvearrowleft T}$ as required.

This completes the proof of Theorem 4. □

Appendix C
Proof of Lemmas from Section VII-D

Proof of Lemma 6

It is enough to show that for every relevant tuple $(\psi, \rho, (n, H))$, and for every free variable X in ψ, the function $\rho(X) : (\mathbb{N} \times \mathcal{P}_{\mathrm{fin}}\mathbb{A}) \to \mathcal{P}K$ has a support of size at most

$$M \cdot (\textit{max-var-supp}(\varphi) + \textit{max-subf-supp}(\varphi)),$$

where M is the number of fixpoint operators that surround ψ in φ. This is proved by induction on the definition of the set of relevant tuples. All cases are evident except the one for $\mu X.\psi$. For this case, making the inductive assumption about $(\mu X.\psi, \rho, (n, H))$, prove by induction on α that every Φ^{α} is supported by the set $\mathrm{supp}(\rho, X, \psi)$. As a result, also each $\rho[X \mapsto \Phi^{\alpha}]$ is supported by that set, which is of size at most

$$(M + 1) \cdot (\textit{max-var-supp}(\varphi) + \textit{max-subf-supp}(\varphi))$$

as required. □

Proof of Lemma 7

By induction on the definition of the set of relevant tuples. All clauses except $\Diamond \psi$ and $\bigvee_{a \in \mathbb{A}} \psi_a$ are trivial, since they do not change the history H.

For the case of $\Diamond \psi$, it is easy to check that in the resulting tuple $(\psi, \rho, (n', H'))$ there is

$$H' \subseteq \mathrm{supp}(y, \psi, \rho),$$

and the conclusion follows since

- $|\mathrm{supp}(y)| \le \textit{dim}(\mathcal{K})$,
- $|\mathrm{supp}(\psi)| \le \textit{max-subf-supp}(\varphi)$, and
- $|\mathrm{supp}(\rho)| \le \textit{max-ctx-supp}(\varphi)$.

For the case of $\psi = \bigvee_{a \in \mathbb{A}} \psi_a$, either $H' = H$ or

$$H' \subseteq \mathrm{supp}(x, \psi, \rho) \cup \{a\}$$

and the bound follows as before. □

Proof of Lemma 8

It is enough to show the inequality for all clauses in the definition of the set of relevant tuples. In all clauses except $\bigvee_{a \in A} \psi_a$ and $\Diamond\psi$ the inequality holds trivially as equality, as then $n' = n$ and $H' = H$.

For the case of $\psi = \bigvee_{a \in \mathbb{A}} \psi_a$, we have

$$(n', H') = ((n, H) \curvearrowleft S) \curvearrowright \{a\},$$

so we conclude by (30). Similarly, for the case of $\Diamond\psi$, we have

$$(n', H') = ((n, H \cup \mathrm{pred}(x)) \curvearrowleft S) \curvearrowright D$$

and conclude from (30) that

$$n' + |H'| = n + |H \cup \mathrm{pred}(x)| \geq n + |H|. \qquad \square$$

Proof of Lemma 9

If ψ is of the form $\top$, a or $\sharp a$, the conclusion holds trivially because the semantics of those formulas does not depend on m_1 or m_2. For other cases, the proof follows by fixpoint induction on the definition of the set of relevant tuples. The two interesting cases are $\bigvee_{a \in \mathbb{A}} \psi_a$ and $\Diamond\psi$, where a nontrivial condition on the size of the anonymous part of the history appears. For both these cases it is important that, by (22) and by Lemma 7, we have

$$m_1, m_2 > \mathit{dim}(\mathcal{K}) \geq 0.$$

As a result, the condition $n > 0$ in the definition of $\llparenthesis \bigvee_{a \in \mathbb{A}} \psi_a \rrparenthesis$ holds both for $n = m_1$ and $n = m_2$, and the condition $|D| \leq n$ in the definition of $\llparenthesis \Diamond\psi \rrparenthesis$ does not restrict the choice of the set D neither for $n = m_1$ nor for $n = m_2$; indeed, the condition $D \subseteq \mathrm{supp}(y)$ becomes a stronger one since $|\mathrm{supp}(y)| \leq \mathit{dim}(\mathcal{K})$.

Finally, by Lemma 8, the inequality (22) is preserved in any tuple relevant for $(\psi, \rho, (m_1, H))$ and $(\psi, \rho, (m_2, H))$. □