SCALAR AND VECTORIAL μ-CALCULUS WITH ATOMS[†]

BARTEK KLIN[∗] AND MATEUSZ ŁEŁYK[∗∗]

University of Warsaw, Dept. of Mathematics, Informatics and Mechanics
e-mail address: klin@mimuw.edu.pl

University of Warsaw, Dept. of Philosophy and Sociology
e-mail address: mlelyk@uw.edu.pl

ABSTRACT. We study an extension of modal μ-calculus to sets with atoms and we study its basic properties. Model checking is decidable on orbit-finite structures, and a correspondence to parity games holds. On the other hand, satisfiability becomes undecidable. We also show expressive limitations of atom-enriched μ-calculi, and explain how their expressive power depends on the structure of atoms used, and on the choice between basic or vectorial syntax.

1. INTRODUCTION

Modal μ-calculus [1,5,6,31] is perhaps the best known formalism for describing properties of labeled transition systems or Kripke models. It combines a simple syntax with a mathematically elegant semantics and it is expressive enough to specify many interesting properties of systems. For example, the property "[in the current state] the predicate p holds, and there exists a transition path where it holds again some time in the future" is defined by a μ-calculus formula:

$$p \wedge \Diamond \mu X.(p \vee \Diamond X).$$

Other similar formalisms, such as the logic CTL* [18], can be encoded in the modal μ-calculus. On the other hand, decision problems such as model checking ("does a given μ-calculus formula hold in a given (finite) model?") or satisfiability ("does a given formula hold in some model?") are decidable.

Formulas of the μ-calculus are built over some fixed vocabulary of basic predicates, such as p above, whose semantics is provided in every model. In principle this vocabulary may be infinite, but this generality is hardly useful: since a formula of the μ-calculus is a finite object, it may only refer to finitely many basic predicates.

In modelling systems, this finiteness may sometimes seem restrictive. Real systems routinely operate on data coming from potentially infinite domains, such as numbers or

---

character strings. Basic predicates observed about a system may reasonably include ones like "a number n was input", denoted here p_n, for every number n. If one considers properties such as "there exists a transition path where the currently input number is input again some time in the future", one is in trouble writing finite formulas to define them. Were infinitary connectives allowed in the formalism, the formula

$$\bigvee_{n\in\mathbb{N}} (p_n \wedge \Diamond\mu X.(p_n \vee \Diamond X))$$

would do the job; however, few good properties of the μ-calculus transport to a naively construed infinitary setting. In particular, an obvious obstruction to any kind of decidability results is that a general infinitary formula is not readily presented as input to an algorithm.

In this paper we introduce μ-calculus with atoms, where infinitary propositional connectives are allowed in a restricted form that includes formulas such as the one above. Roughly speaking, basic properties in formulas, and indeed whole Kripke models, are assumed to be built of *atoms* that come from a fixed infinite structure. Out of many possible structures of atoms, we concentrate on two: *equality atoms*, which come from a pure infinite set without any relations, and *ordered atoms*, where a total ordering on the atoms can be used to build Kripke models and formulas.

Boolean connectives in μ-formulas are indexed by sets that are potentially infinite but are finitely definable in terms of atoms. This makes formulas finitely presentable.

Several basic properties of the standard μ-calculus hold in the atom-based setting. Syntax and semantics of the calculus is defined in a standard way. The model checking problem remains decidable, although this is not trivial, as formulas and models are now, strictly speaking, infinite. A correspondence between μ-formulas and parity games with atoms holds.

Some other properties of the classical calculus fail. The satisfiability problem becomes undecidable, and the atom-based generalization of the finite model property fails. A vectorial extension of the μ-calculus, which in the classical setting is expressively equivalent to the basic "scalar" variant, here becomes strictly more expressive.

Atomic μ-calculi also turn out to be less expressive than might be expected. In particular, the property #Path, which says "there exists a path where no basic predicate holds more than once", although decidable, is not definable. This means that, unlike in the classical setting, an atomic extension of CTL* (where explicit quantification over paths makes such properties easy to define) is not a fragment of the atomic μ-calculus. As it turns out, for atomic CTL* even the model-checking problem is undecidable.

Our approach is a part of a wider programme of extending various computational models to sets with atoms, which are a generalization of nominal sets [37]. For example, in [3] the classical notion of finite automaton was reinterpreted in the universe of sets with atoms, with a result related to register automata [22, 34], an established model of automata over infinite alphabets.

Temporal logics over structures extended with data from infinite domains, and their connections to various types of automata, have been extensively studied in the literature. For example, the linear time logic LTL has been extended with a *freeze quantifier* [16, 17, 21, 38] which, for structures where every position is associated with a single data item, can store the current item for future reference. This can serve as a mechanism for detecting repeated data values (see also [14, 15]). Another known idea is to extend temporal logics with local

constraints over data from a fixed infinite domain, see e.g. [8, 13]. In [20, 26], alternating register automata on data words and trees were studied, closely connected to μ-calculi.

In all these works, the main goal is to study the decidability border for satisfiability (or nonemptiness, in the case of automata). To that end, the authors impose various, sometimes complex restrictions on their logics or automata, and inevitably limit their expressiveness to some extent. In contrast to that, our aim is to lift the classical modal μ-calculus to data-equipped structures in the most syntactically economic way, and to achieve a relatively expressive formalism. As a price for this, the satisfiability problem quickly becomes undecidable. However, we believe that this does not disqualify the atomic μ-calculus as a practical formalism: most applications of temporal logics in system verification rely only on solving the model checking problem, and that remains decidable here on a wide class of structures.

It would be easy to give up even the decidability of model checking. This was done in [36], in a setting very similar to ours, by extending a basic multimodal logic with infinitary boolean connectives subject only to a finite support condition. The resulting logic has few good properties except its huge expressive power (indeed, it can immediately encode our atomic μ-calculus, and much more): its set of formulas is not even countable. Instead, our boolean connectives are subject to a more restrictive condition of orbit finiteness, an idea first used in [4] in the context of first-order logic with atoms.

Another branch of related work is centered around algebras of name-passing processes such as the π-calculus, and variants of the μ-calculus aimed at specific transition systems induced by those algebras. This line of work was started in [11], where a version of the μ-calculus for model checking properties of π-calculus processes was proposed, with a sound-and-complete proof system. Other efforts in this direction, resulting in logics fine-tuned to specific infinite models, include [19, 32], and a more abstract calculus was proposed in [12].

The closest related work in the literature is the *first-order μ-calculus* of Groote et al. [24, 25], where the classical μ-calculus is extended with quantification over data values from arbitrary infinite data domains, and with fixpoints parametrized by data. Our infinitary boolean connectives can be seen as a syntactic variant of quantification over data values, and indeed our atomic μ-calculi in their more expressive, vectorial form can be seen as variants of the first-order μ-calculus specialized to very particular, well-behaved infinite data domains. The main difference is in focus: while [24, 25] aim for an expressive formalism over rich data domains, with little hope for general decidability results and with a focus on pragmatic usability, we insist on keeping model-checking decidable and delineate expressiveness limitations that arise as a result.

This work is a revised and extended version of the conference paper [28]. There, the basic ("scalar") atomic μ-calculus over equality atoms $\mathcal{L}_\mu^=$ was introduced. Here we extend that definition to ordered atoms, and consider a vectorial extension of atomic μ-calculus over both structures of atoms, resulting in four calculi altogether. As it turns out, each of the four has a different expressive power, but (in addition to the obviously similar definitions of syntax and semantics) they share several basic properties, including the decidability of model checking.

The structure of this paper is as follows. In Sect. 2 we briefly recall the modal μ-calculus and related logics in the classical setting. In Sect. 3 we recall the basics of sets with atoms, including the notions of finite support and orbit finiteness, focusing on two important cases: equality atoms and ordered atoms. In Sect. 4 we introduce the syntax and semantics of the atomic μ-calculus both in a basic "scalar" and in a vectorial version, for both these

structures of atoms. In Sects. 5–8 we show various basic properties of atomic μ-calculi, including the decidability of model checking, and a correspondence to parity games and atomic bisimulation games. In Sect. 9 we focus on undecidability results. In Sect. 10 we define properties of Kripke models that separate the four calculi we study in this paper, and in Sect. 11 we formulate a decidable and natural property that is undefinable in all of them. A brief list of future work directions is in Sect. 12.

2. μ-calculus and related logics

To fix the notation and terminology, we begin by recalling basic definitions and properties of the μ-calculus in the classical setting. For a more detailed exposition see e.g. [1, 5–7, 39, 41].

Definition 2.1 (Syntax)**.** Let $\mathbb{P}$ be an infinite set of *basic predicates* and $\mathbb{X}$ an infinite set of *variables*. The set $\mathcal{L}_\mu$ of μ-calculus formulas is generated by the grammar:

$$\phi ::= p \mid X \mid \phi \vee \phi \mid \neg\phi \mid \Diamond\phi \mid \mu X.\phi$$

where p ranges over $\mathbb{P}$ and X over $\mathbb{X}$. We only allow formulas $\mu X.\phi$ where X occurs only positively in ϕ (i.e. under an even number of negations).

We use the following standard abbreviations:

- $\top := p \vee \neg p$,
- $\phi \wedge \psi := \neg(\neg\phi \vee \neg\psi)$,
- $\Box\phi := \neg\Diamond\neg\phi$,
- $\nu X.\phi := \neg\mu X.\neg\phi[X := \neg X]$.

Definition 2.2 (Kripke model)**.** A *Kripke model* is a triple

$$\mathcal{K} = \langle K, \longrightarrow^{\mathcal{K}}, \models^{\mathcal{K}} \rangle$$

such that
- K is a set of *states*,
- $\longrightarrow^{\mathcal{K}} \subseteq K \times K$ is a *transition relation*,
- $\models^{\mathcal{K}} \subseteq K \times \mathbb{P}$ is the *satisfaction relation* for basic predicates.

Superscripts in $\longrightarrow^{\mathcal{K}}$ and $\models^{\mathcal{K}}$ will be often omitted when no risk of confusion arises. As a convention, for a model denoted by a calligraphic letter such as $\mathcal{K}$, its set of states will be denoted by the corresponding italic letter.

Sometimes it will be notationally convenient to speak of the set of basic predicates that hold in a state of a model:

$$\mathrm{pred}^{\mathcal{K}}(x) = \{p \in \mathbb{P} \mid x \models^{\mathcal{K}} p\}.$$

Again, the superscript will usually be omitted.

Definition 2.3 (Semantics)**.** Formulas of μ-calculus are interpreted in the context of a Kripke model $\mathcal{K}$ and an environment, i.e., a partial function $\rho : \mathbb{X} \rightharpoonup \mathcal{P}(K)$. For any formula ϕ, the interpretation $[\![\phi]\!]_\rho \subseteq K$ is defined by induction:
- $[\![p]\!]_\rho = \{x \in K \mid x \models p\}$,
- $[\![X]\!]_\rho = \rho(X)$,

- $\llbracket \neg\phi \rrbracket_\rho = K \setminus \llbracket \phi \rrbracket_\rho$,
- $\llbracket \phi \vee \psi \rrbracket_\rho = \llbracket \phi \rrbracket_\rho \cup \llbracket \psi \rrbracket_\rho$,
- $\llbracket \Diamond\phi \rrbracket_\rho = \{k \in K \mid \exists s \in \llbracket \phi \rrbracket_\rho.\ k \longrightarrow s\}$,
- $\llbracket \mu X.\phi \rrbracket_\rho = \mathrm{lfp}(F)$, where $F(A) = \llbracket \phi \rrbracket_{\rho[X \mapsto A]}$.

In the last clause, the least fixpoint is taken for a function $F : \mathcal{P}(K) \to \mathcal{P}(K)$. Subsets of K form a complete lattice and (thanks to the positivity assumption in Definition 2.1) the function F is monotone, so by Tarski's theorem the least fixpoint exists and it arises as the union of an increasing chain of approximants:

$$A^0 \subseteq A^1 \subseteq A^2 \subseteq \cdots A^\omega \subseteq A^{\omega+1} \subseteq \cdots$$

indexed by ordinal numbers, where

- $A^{\alpha+1} = F(\alpha)$ and
- $A^\beta = \bigcup_{\alpha < \beta} A^\alpha$ for a limit ordinal β.

In particular, $A^0 = \emptyset$.

We write $\llbracket \phi \rrbracket$ (or $\llbracket \phi \rrbracket^{\mathcal{K}}$ if $\mathcal{K}$ is less clear from context) instead of $\llbracket \phi \rrbracket_\rho$ if ρ is empty (i.e., nowhere defined). We say that ϕ *holds* in a state $x \in K$ if $x \in \llbracket \phi \rrbracket$ and denote it $x \models \phi$; this clearly agrees with the satisfaction relation $\models$ for basic predicates.

Example 2.4. The formula $\mu X.(p \vee \Diamond X)$ holds for a state x in a Kripke model $\mathcal{K}$ if and only if there is a finite path from x to a state where p holds. The formula $\nu X.(\Box X \wedge \mu Y.(p \vee \Box Y))$ holds in those states x where p holds infinitely often on every path from x.

For applications in system verification, the following two problems are considered:

Model checking: given a finite Kripke model $\mathcal{K}$, a state $x \in K$ and a formula $\phi \in \mathcal{L}_\mu$, decide if $x \models \phi$.

Satisfiability: given a formula $\phi \in \mathcal{L}_\mu$, decide if there exists a Kripke model $\mathcal{K}$ and a state $x \in K$ such that $x \models \phi$.

It is very easy to see that model checking is decidable: in a finite Kripke model, one can compute the semantics of a formula inductively, directly from the definition. A more efficient procedure can be derived from a correspondence of μ-calculus with parity games.

Some well-known logics used in system verification can be translated into fragments of the μ-calculus. We give two important examples:

Definition 2.5 (CTL*)**.** In the logic CTL* we distinguish state formulas Φ and path formulas ϕ, formed according to the following grammar:

$$\Phi ::= p \mid \Phi \vee \Phi \mid \neg\Phi \mid \exists\phi \qquad\qquad \phi ::= \Phi \mid \phi \vee \phi \mid \neg\phi \mid \phi\mathsf{U}\phi \mid \mathsf{X}\phi$$

where p comes from some fixed set of propositional variables.

Standard notational conventions include:

- $\forall\phi = \neg\exists\neg\phi$,
- $\phi\mathsf{R}\psi = \neg(\neg\phi\mathsf{U}\neg\psi)$,

- $\mathsf{F}\phi = \top\mathsf{U}\phi$,
- $\mathsf{G}\phi = \neg\mathsf{F}\neg\phi$.

CTL* formulas are interpreted in Kripke models $\mathcal{K}$: state formulas are interpreted over states, and path formulas over paths. In the following definition of a satisfaction relation $\models$, for a path $\pi = \langle x_0, x_1, x_2, \ldots \rangle$, by $\pi[n..]$ we denote the subpath starting at x_n.

- $x \models \exists\phi \iff$ for some path π starting at x, $\pi \models \phi$.
- $\pi \models \Phi \iff \pi[0] \models \Phi$.

- $\pi \models \mathsf{X}\phi \iff \pi[1..] \models \phi$
- $\pi \models \phi\mathsf{U}\psi \iff$ there exists $j \geq 0$ s.t. $\pi[j..] \models \psi$ and for all $i < j$, $\pi[i..] \models \phi$.

Boolean connectives are interpreted as expected.

The logic LTL is the fragment of CTL* where the symbol $\exists$ does not occur, with semantics inherited from CTL*. Usually LTL (where the distinction between state and path formulas disappears) is interpreted in models that are infinite paths. Slightly more generally (and more conveniently for our purposes), we will interpret them over pointed *deterministic* Kripke models, i.e., ones where for each $x \in K$ there is exactly one $y \in K$ such that $x \longrightarrow y$. This makes little difference, since in such a model every state uniquely determines an infinite path.

Sometimes it is convenient to consider a syntactic extension of the μ-calculus to a *vectorial* form. There, the fixpoint construct $\mu X.\phi$ is generalized to one of the form

$$\phi = \mu X_i. \left\{ \begin{array}{cc} X_1. & \phi_1 \\ & \vdots \\ X_n. & \phi_n \end{array} \right\} \tag{2.1}$$

where $\{X_1, \ldots, X_n\} \subseteq \mathbb{X}$ (where we assume $X_i \neq X_j$ for $i \neq j$) is any finite set of variables, X_i is some element of that set, and $\phi_1, \ldots, \phi_n$ are formulas where all of $X_1, \ldots, X_n$ are considered bound (and, as before, occur only positively).

There are a few intuitive notions concerning the body of the formula in (2.1), and it will be useful to keep them in mind when we extend the vectorial μ-calculus with atoms in Section 4. An expression

$$X_j.\phi_j$$

may be seen as an *equation* (where the X_j is the left-hand side, and the ϕ_j the right-hand side), and the set

$$\Phi = \{X_j.\phi_j\}_{j=1..n}$$

is a *system of equations* where each variable appears on the left-hand side exactly once. Alternatively, it can be seen as a function from the finite set $\{X_j\}_{j=1..n}$ to the set of formulas. The variable X_i in ϕ, which we call the *entry variable* of the fixpoint formula, must belong to that finite set. Then, striving for syntax that fits in a single line of text, the formula in (2.1) may be rewritten as:

$$\phi = \mu X_i.\Phi = \mu X_i. \{X_j.\phi_j\}_{j=1..n}. \tag{2.2}$$

The set of vectorial μ-calculus formulas will be denoted by $\vec{\mathcal{L}}_\mu$. When we want to distinguish the original calculus $\mathcal{L}_\mu$ from the vectorial one, we will call it the *scalar μ-calculus*.

For the semantics of a formula ϕ as above, in the context of a model $\mathcal{K}$ and an environment $\rho : \mathbb{X} \rightharpoonup \mathcal{P}(K)$, consider a function

$$F : (\mathcal{P}(K))^n \to (\mathcal{P}(K))^n$$

defined by:

$$(F(A_1, \ldots, A_n))_j = [\![\phi_j]\!]_{\rho[X_1 \mapsto A_1, \ldots, X_n \mapsto A_n]} \quad \text{for } j = 1, \ldots, n,$$

then, for $(B_1, \ldots, B_n)$ the least fixed point of F, define

$$[\![\phi]\!]_\rho = B_i.$$

It is not difficult to see that for $n = 1$ this specializes to the original semantics in Definition 2.3.

In the classical setting, vectorial modal μ-calculus is expressively equivalent to the scalar one, i.e., every formula can be rewritten to a semantically equivalent scalar form, using the so-called Bekić principle. The vectorial calculus is therefore mostly considered for convenience, as its formulas seem to be more succint, and they correspond more closely to alternating tree automata. As we shall see in Section 10, in the atomic setting the difference is more significant, and adding vectorial formulas will extend the expressivity of atomic μ-calculus.

3. SETS WITH ATOMS

We now recall the basic notions and results concerning sets with atoms, beginning with the case of *equality atoms*, also known as nominal sets [37]. There are several essentially equivalent ways to introduce these; we follow the set-theoretic presentation of [23], culminating in the notion of orbit-finite sets [3,37] and computable operations on them. Then we recall a more general setting, advocated in [3], where atoms are equipped with a nontrivial relational structure, in particular a total order.

Remark 3.1. When developing a theory of "objects with atoms" (like automata with atoms or modal μ-calculus with atoms) there seems to be two natural ways to proceed: one can either (i) stick to the universe of (sufficiently well-behaved) sets with atoms and work exclusively within this universe (which corresponds to working in a formal system such as ZFA), or (ii) work in a broader universe of all sets and within this universe define what does it mean to be a (sufficiently well-behaved) set with atom. We regard the second strategy as more convenient to our immediate goals so will consistently follow it.

3.1. **Equality atoms.** 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 $\mathrm{Aut}(\mathbb{A})$.

Loosely speaking, 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 hierachy, by transfinite induction on ordinal numbers α:

$$\mathcal{U}_0^{\mathbb{A}} = \emptyset, \qquad \mathcal{U}_{\alpha+1}^{\mathbb{A}} = \mathcal{P}(\mathcal{U}_\alpha^{\mathbb{A}}) + \mathbb{A}, \qquad \mathcal{U}_\beta^{\mathbb{A}} = \bigcup_{\alpha < \beta} \mathcal{U}_\alpha^{\mathbb{A}} \quad \text{for } \beta \text{ a limit ordinal,}$$

where $+$ denotes disjoint union of sets.

We are interested in sets that only depend on a finite number of atoms, in the following sense. Atom automorphisms act on the universe $\mathcal{U}^{\mathbb{A}}$ by consistently renaming all atoms in a given set. Formally, this is again defined by transfinite induction. This defines a group action:

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

For a finite set $S \subset \mathbb{A}$, let $\mathrm{Aut}_S(\mathbb{A})$ 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 $\pi \in \mathrm{Aut}_S(\mathbb{A})$. A set is *equivariant* if it is supported by the empty set. If x has a finite support then it has the least finite support (see [3, Cor. 9.4] for a proof), denoted $\mathrm{supp}(x)$.

Remark 3.2. In [23, 37] a slightly different variant of equality atoms is developed, where $\mathrm{Aut}(\mathbb{A})$ is taken to be the group of *finite* bijections, i.e. those that fix almost all atoms. As argued in [37, Sec. 6.2], this difference is irrelevant as far as finitely supported sets are concerned: exactly the same finite supports exist under both definitions. In the context of finite bijections as atom automorphisms, existence of least supports was first proved in [23, Prop. 3.4]. We allow all atom bijections as automorphisms since this approach generalizes more naturally to richer atom structures: anticipating ordered atoms (see Section 3.2 below), note that there are no nontrivial finite monotone bijections on the total order of rational numbers.

From now on, we shall only consider sets with atoms that are *hereditarily finitely supported*, i.e., ones that have a finite support, whose every element has some finite support and so on.

Relations, functions etc. are sets in the standard 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

$$\langle x, y \rangle \in R \text{ implies } \langle x \cdot \pi, y \cdot \pi \rangle \in R$$

and a function $f : X \to Y$ is equivariant if

$$f(x \cdot \pi) = f(x) \cdot \pi$$

for every $\pi \in \mathrm{Aut}(\mathbb{A})$.

For any x with atoms, the *S-orbit* of x is the set $\{x \cdot \pi \mid \pi \in \mathrm{Aut}_S(\mathbb{A})\}$. Note that if S supports x then the S-orbit of x is the singleton $\{x\}$. We shall write $x \sim_S y$ to say that x and y belong to the same S-orbit, and we shall omit the subscript and write simply $x \sim y$ for empty S. Similarly, an $\emptyset$-orbit will simply be called an orbit.

For any S, S-orbits form a partition of the universe $\mathcal{U}$. Moreover, for any S-supported set X, the S-orbits of its elements form a partition of X. We call such X *S-orbit-finite* if it is a union of finitely many S-orbits. If $S \subseteq T$ are finite, S supports X and X is S-orbit-finite, then (T supports X and) 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". It is not difficult to check that every finitely supported subset of an orbit-finite set is orbit-finite.

Example 3.3. • Any classical set (without atoms) is an equivariant set with atoms. Since its every element is also equivariant, it forms its own orbit. Therefore, a classical set is orbit-finite if and only if it is finite.

• An atom $a \in \mathbb{A}$ has no elements, and it is supported by $\{a\}$. Every finite set of atoms $S \subseteq \mathbb{A}$ is supported by S, and every element of it forms a singleton S-orbit. Its complement $\mathbb{A} \setminus S$ is also supported by S, and it is a single S-orbit. A subset of $\mathbb{A}$ that is neither finite nor co-finite is not finitely supported, so we do not consider it a legal set with atoms.

• 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 all ordered pairs of atoms, are equivariant sets. The first two have a single orbit each, and the last one is a union of two orbits:

$$\mathbb{A}^2 = \{\langle a, a \rangle \mid a \in \mathbb{A}\} \cup \{\langle a, b \rangle \mid a \neq b \in \mathbb{A}\}.$$

Similarly, $\mathbb{A}^n$ is orbit-finite for every $n \in \mathbb{N}$. The set $\mathbb{A}^*$ of finite sequences of atoms is hereditarily finitely supported, but not orbit-finite.

- The powerset $\mathcal{P}(\mathbb{A})$ is equivariant itself, but it contains elements that are not finitely supported, and therefore it is not considered a legal set with atoms. However, for every legal set with atoms X, the set $\mathcal{P}_{\mathrm{fs}}(X)$ of *finitely supported* subsets of X has the same support as X and is a legal set of atoms.
- There are four equivariant binary relations on $\mathbb{A}$: the empty relation, the equality relation, the inequality relation and the full relation.
- There is no equivariant function from $\binom{\mathbb{A}}{2}$ to $\mathbb{A}$, but $\{\langle\{a,b\},a\rangle \mid a \neq b \in \mathbb{A}\}$ is a legal equivariant relation, and the function constant at an atom a is supported by $\{a\}$. The only equivariant function from $\mathbb{A}$ to $\mathbb{A}$ is identity. The only equivariant functions from $\mathbb{A}^2$ to $\mathbb{A}$ are the two projections, and the only equivariant function from $\mathbb{A}$ to $\mathbb{A}^2$ is the diagonal $a \mapsto \langle a,a\rangle$.

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. In [3] (with the idea going back to [9]), it was observed that every single-orbit equivariant set is in an equivariant bijection with a set of k-tuples of distinct atoms, suitably quotiented by a subgroup $G \leq \mathrm{Sym}(k)$ of the symmetric group on k elements. Thus such a set can be presented by a number k and the finite group G, and an orbit-finite set is a formal union of such presentations. A somewhat more readable and concise scheme was used in [27], where orbit-finite sets are presented by set-builder expressions of the form

$$\{e \mid v_1, \ldots, v_n \in \mathbb{A}, \phi\} \tag{3.1}$$

where e is again an expression, v_i are bound atom variables and ϕ is a first-order formula with equality. We refer to [27] for a precise formulation (and to [35] for a proof that all orbit-finite sets can be presented this way, up to an equivariant bijection); suffice it to say that the expressions in Example 3.3 are of this form, and other similar expressions are allowed. The set defined by such an expression is supported by the atoms that appear freely in the expression.

Remark 3.4. Using this representation, several basic operations on orbit-finite sets are computable, including:

- union and intersection of sets, cartesian product, set difference,
- checking whether two sets are equal, or whether one is an element (or a subset) of the other,
- applying an orbit-finite function to an argument, composing functions or relations,
- finding the direct image of a set along a function, finding the image of a subset along a relation, etc.
- 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 [29, 30].

The following easy but useful fact says that there are not too many orbit-finite sets or finitely supported relations (or functions) between them.

Proposition 3.5. *For any finite $S \subseteq T \subseteq \mathbb{A}$,*

(1) *For any $n \in \mathbb{N}$, there are only finitely many (up to equivariant bijection) n-orbit sets supported by S.*

(2) *For any orbit-finite set X supported by S, there are only finitely many subsets of X supported by T.*
(3) *For any orbit-finite sets X, Y supported by S, there are only finitely many relations between X and Y supported by T.*

Proof. Part (1) follows from a stronger representation result mentioned above [3], which says that every single-orbit set is a quotient of the set of all distinct tuples of atoms of a fixed length. For $n > 1$, every n-orbit set is obviously just an n-tuple of single-orbit sets.

For part (2), notice that X has only finitely many T-orbits, and a subset of X supported by T must arise by selecting a subset of these orbits.

Part (3) follows directly from part (2) as if X and Y are supported by S, then $X \times Y$ is supported by S. $\square$

3.2. **Ordered atoms.** Intuitively, orbit-finite structures with atoms are those that can be built of atoms using equality as the only relation between them. As advocated in [3], it is possible and useful to consider atoms with more structure. Perhaps the most significant case is that of atoms being rational numbers with the ordering relation.

Technically, this amounts to taking $\mathbb{A} = \mathbb{Q}$ and putting $\mathrm{Aut}(\mathbb{A})$ to be the group of automorphism of the total order $\mathbb{Q}$, i.e., the group of monotone bijections on rational numbers. All other definitions follow without further change. The basic results mentioned above for equality atoms hold here as well, although the fact that the least support $\mathrm{supp}(x)$ exists, requires a different proof (see [3, Cor. 9.5]). In particular, operations listed in Remark 3.4 remain computable.

Since $\mathrm{Aut}(\mathbb{A})$ is now a subgroup of the automorphism group considered for equality atoms, it is easy to see that all legal sets with equality atoms remain legal for ordered atoms. However, new legal sets, functions and relations appear.

Example 3.6. • For any $a < b \in \mathbb{A}$, the open interval $(a; b) \subseteq \mathbb{A}$ becomes legal, at it is finitely supported by $S = \{a, b\}$. The interval forms a single S-orbit. The closed interval $[a; b]$ is also supported by S, and it comprises three S-orbits: $\{a\}$, $\{b\}$ and the open interval. In general, finitely supported subsets of $\mathbb{A}$ are exactly those that are finite unions of open intervals and single points.
• $\mathbb{A}$ and $\binom{\mathbb{A}}{2}$ remain equivariant single-orbit sets, but the set of ordered pairs $\mathbb{A}^2$ now decomposes into three orbits:
$$\mathbb{A}^2 = \{\langle a, a \rangle \mid a \in \mathbb{A}\} \cup \{\langle a, b \rangle \mid a < b \in \mathbb{A}\} \cup \{\langle a, b \rangle \mid a > b \in \mathbb{A}\}.$$
Nevertheless, $\mathbb{A}^n$ remains orbit-finite for every n.
• There are now two equivariant functions from $\binom{\mathbb{A}}{2}$ to $\mathbb{A}$: the minimum and maximum. The only equivariant function from $\mathbb{A}$ to $\mathbb{A}$ is still identity. There are four equivariant functions from $\mathbb{A}^2$ to $\mathbb{A}$: the two projections, minimum and maximum, but the only equivariant function from $\mathbb{A}$ to $\mathbb{A}^2$ remains the diagonal $a \mapsto \langle a, a \rangle$.

Remark 3.4 and Proposition 3.5 remain true for ordered atoms.

4. μ-CALCULUS WITH ATOMS

In this section we present our central definition: an extension of the classical μ-calculus to sets with atoms. We begin by considering the scalar calculus; the vectorial calculus is defined towards the end of this section.

4.1. **Syntax.** Syntactically, the (scalar) μ-calculus with atoms (or *atomic μ-calculus*) is simply an extension of the (scalar) classical formalism with orbit-finite propositional connectives.

Definition 4.1. Let $\mathbb{P}$ be an equivariant set with atoms of basic propositions, and let $\mathbb{X}$ be an equivariant set with atoms of variables. The set $\mathcal{L}_\mu^\mathbb{A}$ of formulas of the atomic μ-calculus is generated by the following grammar:

$$\phi ::= p \mid X \mid \bigvee \Phi \mid \neg \phi \mid \Diamond \phi \mid \mu X.\phi$$

where p ranges over $\mathbb{P}$, X ranges over $\mathbb{X}$ and Φ ranges over *orbit-finite* sets of formulas. As usual we only allow $\mu X.\phi$ where X occurs only positively in ϕ.

The above definition is parametrised by a basic structure $\mathbb{A}$ of atoms, and in particular it makes sense for both equality atoms and ordered atoms. When we need to specify and distinguish between these two cases, we will denote the equality-atomic μ-calculus by $\mathcal{L}_\mu^=$, and the ordered-atomic μ-calculus by $\mathcal{L}_\mu^<$.

The "orbit-finite set of formulas" in the definition refers to a canonical action of $\mathrm{Aut}(\mathbb{A})$ on formulas, extending the action on $\mathbb{P}$ (and on $\mathbb{X}$, if it is nontrivial) inductively. Note that, despite ostensibly infinite disjunctions, every formula in $\mathcal{L}_\mu^\mathbb{A}$ has a finite depth, since two formulas in the same orbit necessarily have the same depth. Thanks to this, no need arises for transfinite induction in reasoning about the syntax of $\mathcal{L}_\mu^\mathbb{A}$ formulas.

We use the same syntactic conventions as in the classical case:

- $\top := p \vee \neg p,$
- $\bigwedge \Phi := \neg \bigvee \{\neg \phi \mid \phi \in \Phi\},$
- $\Box \phi := \neg \Diamond \neg \phi,$
- $\nu X.\phi := \neg \mu X.\neg \phi[X := \neg X].$

With these conventions, every formula can be written in the negation normal form, where negation occurs only in front of a basic proposition or a variable.

Often it is notationally convenient to view an orbit-finite set Φ as a family of formulas indexed by a simpler orbit-finite set. For example, we may write

$$\bigvee_{a \in \mathbb{A}} \Diamond a \qquad \text{to mean} \qquad \bigvee \{\Diamond a \mid a \in \mathbb{A}\}.$$

This quantifier-like notation suggests that orbit-finite disjunctions and conjunctions can be used to quantify over atoms existentially and universally. This intuition will be useful in reading our example formulas in what follows.

Example 4.2. Put $\mathbb{P} = \mathbb{A}$. For every $a \in \mathbb{A}$ let $\Phi_a = \{\neg b \mid b \in \mathbb{A} \setminus a\}$. Then Φ_a is supported by $\{a\}$ and orbit-finite, hence $\bigwedge \Phi_a := \bigwedge_{b \neq a} \neg b$ is a formula in $\mathcal{L}_\mu^\mathbb{A}$. The set

$$\Psi = \left\{ \Diamond \left(a \wedge \bigwedge_{b \neq a} \neg b \right) \mid a \in \mathbb{A} \right\}$$

is equivariant and orbit-finite, hence

$$\bigwedge \Psi = \bigwedge_{a \in A} \Diamond \left(a \wedge \bigwedge_{b \neq a} \neg b \right)$$

is also a legal formula in $\mathcal{L}_\mu^\mathbb{A}$.

Remark 4.3. In the classical μ-calculus, one often wants a formula to be *clean*, or *well-named*, meaning that every bound variable is bound only once. This is to ensure that to every variable one can associate a unique binding occurrence of it. In the presence of infinitary connectives, achieving cleanness may seem problematic. For example, in the formula

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

the variable X occurs infinitely many times, and naively replacing each binding occurrence with a completely fresh variable would result in an orbit-infinite conjunction. This is why we will not insist that our formulas be presented in a clean form; instead, we will only require that no variable is bound more than once on any syntactic path from the root of a formula to its subformula. It is easy to rewrite any formula to an equivalent one of this form (for example by annotating each variable with a natural number that is the depth of its binding occurrence in the syntactic tree of a formula), and it will allow us to speak of "the binding occurrence" for any occurrence of a bound variable.

The following syntactic properties can be easily proved by induction on the depth of formulas:

- Every formula ϕ is finitely supported.
- For every formula ϕ, the set of its subformulas is finitely supported and orbit-finite.
- The relation $\trianglelefteq \subseteq \mathcal{L}_\mu^\mathbb{A} \times \mathcal{L}_\mu^\mathbb{A}$ of being a subformula is equivariant, i.e. for every $\pi \in \mathrm{Aut}(\mathbb{A})$ and every $\phi, \psi \in \mathcal{L}_\mu^\mathbb{A}$ we have

$$\phi \trianglelefteq \psi \text{ if and only if } \phi \cdot \pi \trianglelefteq \psi \cdot \pi.$$

4.2. **Semantics.** Semantics of the atomic μ-calculus is a straightforward extension of the classical one: we simply require all sets and relations to be sets with atoms, and replace finite models with orbit-finite ones. The following is a simple rewriting of Definition 2.2:

Definition 4.4 (Kripke model)**.** An *atomic Kripke model* is a triple

$$\mathcal{K} = \langle K, \longrightarrow^\mathcal{K}, \models^\mathcal{K} \rangle$$

such that

- K is a set with atoms,
- $\longrightarrow^\mathcal{K} \subseteq K \times K$ is a finitely supported *transition relation*,
- $\models^\mathcal{K} \subseteq K \times \mathbb{P}$ is a finitely supported *satisfaction relation* for basic predicates.

We shall say that an atomic Kripke model $\mathcal{K}$ is *orbit-finite* if the set K is so. A set S supports $\mathcal{K}$ if and only if it supports K, $\longrightarrow^\mathcal{K}$ and $\models^\mathcal{K}$.

We use the same notational conventions regarding dropping the superscripts as for the classical case. Also as in the classical case, the satisfaction relation can be equivalently defined by the function:

$$\mathrm{pred}(x) = \{p \in \mathbb{P} \mid x \models p\}.$$

which is now a function from K to $\mathcal{P}_{\mathrm{fs}}(\mathbb{P})$, the set of finitely supported subsets of $\mathbb{P}$.

Example 4.5. A model $\mathcal{K}$ with:

- $\{\star\} \cup \mathbb{A}$ as the set of states,
- transitions $\star \longrightarrow a$ for all $a \in \mathbb{A}$,

- satisfaction relation $\models$ such that $\mathrm{pred}(\star) = \emptyset$ and $\mathrm{pred}(a) = \{a\}$ for $a \in \mathbb{A}$,

is an infinite but orbit-finite, equivariant atomic Kripke model. It can be drawn as:

Example 4.6. A model $\mathcal{K}$ with:

- $\mathbb{A}^*$, i.e., finite sequences of atoms as states,
- transitions $w \longrightarrow wa$ for all $w \in \mathbb{A}^*$ and $a \in \mathbb{A}$,
- satisfaction relation $\models$ such that $\mathrm{pred}(\epsilon) = \emptyset$ and $\mathrm{pred}(a_1 \cdots a_n) = \{a_n\}$,

is an equivariant atomic Kripke model with infinitely many orbits.

Definition 4.7. For an atomic Kripke model $\mathcal{K}$, the meaning of a formula $\phi \in \mathcal{L}_\mu^{\mathbb{A}}$ in a variable environment $\rho : \mathbb{X} \rightharpoonup \mathcal{P}(K)$ is defined exactly as in Definition 2.3, with the obvious modification:

$$\left[\!\!\left[\bigvee \Phi \right]\!\!\right]_\rho = \bigcup \{ [\![\phi]\!]_\rho \mid \phi \in \Phi \}. \tag{4.1}$$

Remark 4.8. There is an apparent lack of consistency in literally transporting Definition 2.3 to sets with atoms. In one clause taken from that definition:

$$[\![\mu X.\phi]\!]_\rho = \mathrm{lfp}(F), \text{ where } F(A) = [\![\phi]\!]_{\rho[X \mapsto A]},$$

the least fixpoint of a function $F : \mathcal{P}(K) \to \mathcal{P}(K)$ is considered. However, as mentioned in Example 3.3, $\mathcal{P}(K)$ is not a legal set with atoms in general. The set $\mathcal{P}_{\mathrm{fs}}(K)$ of finitely supported subsets of K is legal, and one could consider a restriction of F to:

$$F : \mathcal{P}_{\mathrm{fs}}(K) \to \mathcal{P}_{\mathrm{fs}}(K).$$

However, $\mathcal{P}_{\mathrm{fs}}(K)$ is not a complete lattice, so Tarski's theorem does not immediately apply to it.

Our approach is to interpret Definition 2.3 and its extending clause in Definition 4.7 literally, initially without regard for the legality of sets with atoms, and then to prove Lemma 4.9 below, which shows by induction that the semantics of each formula, including fixpoint formulas, is in fact a legal, finitely supported set.

A more principled approach would be to restrict functions F and environments ρ to finitely supported sets from the beginning. This would amount to reinterpreting Definition 2.3 in the set theory with atoms ZFA, an approach pursued e.g. in [23]. One then needs to show a suitable version of Tarski's theorem to infer the existence of least fixpoints. Such a theorem indeed holds in a general topos-theoretic setting (see e.g. [2]); for sets with atoms, it guarantees the existence of least fixpoints of monotone, finitely supported functions on lattices where least upper bounds exist for all finitely supported families (see also e.g. [33]).

We choose the more elementary approach to avoid a formal reliance on ZFA. We then need to state the following lemma, which implies that the semantics of every formula is finitely supported.

Lemma 4.9. *For every $\phi \in \mathcal{L}_\mu^{\mathbb{A}}$, an atomic Kripke model $\mathcal{K}$ and a variable environment ρ, the set $[\![\phi]\!]_\rho \subseteq K$ is supported by $S = supp(\phi) \cup supp(\mathcal{K}) \cup B \subseteq \mathbb{A}$, where B is any support of ρ.*

Proof (sketch). By structural induction on ϕ. For example, consider the case of orbit-finite disjunction above, i.e., $\phi = \bigvee \Phi$. By the inductive assumption, the set $\{[\![\phi]\!]_\rho \mid \phi \in \Phi\}$ is

supported by S. Since $\bigcup$ is an equivariant operation on finitely supported families of sets with atoms, the lemma follows.

The most interesting case is that of the fixpoint operator. Recall that

$$[\![\mu X.\phi]\!]_\rho = \mathrm{lfp}(F), \text{ where } F(A) = [\![\phi]\!]_{\rho[X \mapsto A]}$$

is obtained as the union of an increasing chain of approximants:

$$A^0 \subseteq A^1 \subseteq A^2 \subseteq \cdots A^\omega \subseteq A^{\omega+1} \subseteq \cdots$$

defined as in Definition 2.3. By ordinal induction, and using the inductive assumption about ϕ, it is easy to show that A^α is supported by S for every ordinal α. Indeed, if A^α is supported by S then for any $\pi \in \mathrm{Aut}_S(\mathbb{A})$:

$$A^{\alpha+1} \cdot \pi = ([\![\phi]\!]_{\rho[X \mapsto A^\alpha]}) \cdot \pi = [\![\phi]\!]_{\rho[X \mapsto A^\alpha]} = A^{\alpha+1}$$

where the middle equality holds by the inductive assumption about ϕ, using the simple observation that, since S supports $\mu X.\phi$ (hence also ϕ and X), ρ and A^α, then it also supports $\rho[X \mapsto A^\alpha]$.

As a result, the set $[\![\mu X.\phi]\!]_\rho$, being the union of an increasing chain of S-supported sets, it itself S-supported. $\qquad\square$

Atomic CTL* and atomic LTL are defined by analogy to atomic μ-calculus, extending Definition 2.5 with orbit-finite disjunctions, with semantics extended by analogy to (4.1) in Definition 4.7. With CTL* a design decision is to be made: in the semantic clause

• $x \models \exists\phi \iff$ for some path π starting at $x, \pi \models \phi$,

do we require the path π to be finitely supported or not? Note that even an equivariant Kripke model can contain infinite paths that are not finitely supported. As it turns out, this choice matters little: in Theorem 9.3 we shall prove that model checking for atomic CTL* is undecidable and the argument there works whether or not we choose paths as being finitely supported.

4.3. **The vectorial calculus.** A beginner's recipe for moving atom-less mathematical notions to sets with atoms is rather straightforward: replace all functions and relations by equivariant or finitely supported ones, all finite structures by orbit-finite ones, and then see what happens. With this in mind, notice that the classical definition of vectorial μ-calculus includes a finiteness condition that is not explicitly present in the scalar μ-calculus: namely, in a vectorial formula as in (2.1), the set of variables X_j is finite. A natural idea is therefore to replace that set with an orbit-finite one.

Formally, in the *vectorial atomic μ-calculus* $\vec{\mathcal{L}}_\mu^\mathbb{A}$, the fixpoint construction is generalized to the form

$$\phi = \mu \check{X}.\Phi$$

where $\check{X} \in \mathbb{X}$ is the *entry variable* and Φ is an orbit-finite set of *equations* of the form $X.\phi$, $X \in \mathbb{X}$ and ϕ is a formula. We assume that in Φ no variable occurs on the left-hand side of two different equations, and that the entry variable occurs on the left-hand side of some equation.

As with orbit-finite boolean connectives, it is often useful to see the orbit-finite set Φ as a family of equations

$$\{X_j.\phi_j\}_{j \in J}$$

indexed by some simpler orbit-finite set J, with the entry variable pointed to by some $i \in J$. The formula ϕ can be then written as

$$\phi = \mu X_i. \{X_j.\phi_j\}_{j \in J},$$

quite resembling the classical vectorial formula (2.2).

The semantics of such a formula is defined very much like in the classical vectorial calculus, except that here the operator whose fixpoint is calculated runs not on finite tuples $(A_1, \ldots, A_n)$, but tuples $(A_j)_{j \in J}$, indexed by an orbit-finite set J, where $A_j \subseteq K$ (considerations in Remark 4.8 apply here as well). The set of such tuples is (in a classical sense) a complete lattice under pointwise intersections and unions, so least fixpoints of monotone operators exist on it. If the operator defined by:

$$(F(A_j)_{j \in J})_k = [\![\phi_k]\!]_{\rho[X_j \mapsto A_j]_{j \in J}} \quad \text{for } k \in J$$

has a tuple $(B_j)_{j \in J}$ as its least fixpoint, we put

$$[\![\phi]\!] = B_i$$

(where X_i is the entry variable for ϕ) as in the classical setting.

It is easy to check that Lemma 4.9 still holds for this extended semantics, with essentially the same proof.

This syntax and semantics makes sense for any choice of atoms. When we need to specify and distinguish between the two specific cases considered in this paper, we will denote the vectorial equality-atomic μ-calculus by $\vec{\mathcal{L}}_\mu^=$, and the vectorial ordered-atomic μ-calculus by $\vec{\mathcal{L}}_\mu^<$.

Example 4.10. Working over ordered atoms, consider a vocabulary of basic propositions that coincides with the set $\mathbb{A}$ of atoms. Let us analyse the following formula in $\vec{\mathcal{L}}_\mu^<$:

$$\phi = \bigvee_{a \in \mathbb{A}} \nu X_a. \left\{ X_b.(\Diamond b) \wedge \bigvee_{c > b} X_c \right\}_{b \in \mathbb{A}}.$$

Here, the indexing set $J = \mathbb{A}$ is single-orbit, and for each atom b, the formula

$$\phi_b = (\Diamond b) \wedge \bigvee_{c > b} X_c$$

says that:

- some immediate successor of the current state satisfies b, and
- for some $c > b$, the property X_c (which, in the fixpoint construction, will be bound to an analogous formula ϕ_c) holds in the current state.

Altogether, the formula ϕ says that there is an infinite increasing chain of atoms such that each atom in that chain is satisfied in some immediate successor of the current state. This, by the way, is equivalent to saying that there is any infinite *set* of atoms with this property; this is because the set of immediate successors of a particular state in a Kripke model is always finitely supported, and every infinite but finitely supported subset of $\mathbb{A}$ contains an open interval and therefore it contains an infinite increasing chain.

It is instructive to see how the greatest fixpoint in the semantics of ϕ is approximated on an equivariant Kripke model $\mathcal{K}$, by successive applications of a monotone operator on $\mathbb{A}$-indexed tuples of subsets of K:

- at the starting point, every variable X_b is mapped to the full set K;

- after one step, X_b is mapped to the set all states where some successor satisfies b;
- after two steps, X_b is mapped to the set of all states where:
  - some successor satisfies b,
  - some successor satisfies some atom greater than b;
- . . .
- after k steps, X_b is mapped to the set of all states where:
  - some successor satisfies b,
  - there are at least $k-1$ atoms greater than b that are satisfied in some successor.

Note that, after each step, the approximate interpretation of X_b is a subset of K supported by $\{b\}$; moreover, the function that maps each X_b to its approximate interpretation is equivariant. From this it follows that if the model $\mathcal{K}$ is orbit-finite then the fixpoint is reached after finitely many steps.

In the atom-less world one does not gain expressivity by allowing fixpoints over systems of equations. This is a consequence of the following Bekić principle:

Theorem 4.11 (Bekić)**.** *Let D, E be any complete lattices and suppose $F : D \times E \to D$ and $G : D \times E \to E$ are two monotone functions. Then the least fixpoint of $\langle F, G \rangle : D \times E \to D \times E$ is the pair $\langle \widehat{f}, \widehat{g} \rangle$, where*

$$\widehat{f} = \mu f.F(f, \mu g.G(f,g))$$
$$\widehat{g} = \mu g.G(\mu f.F(f,g), g)$$

As we shall see in Section 10, contrary to the classical setting, vectorial *atomic* μ-calculus is strictly more expressive than its scalar counterpart. However, Bekić principle can still be used some extent, to obtain a more regular normal form for vectorial atomic formulas. Recall that when introducing syntax we allowed the fixpoint to be taken over arbitrary orbit-finite sets of equations. Next proposition shows that in fact we can restrict ourselves to just *single-orbit* systems of equations. More precisely: we can demand that in a well-built formula

$$\mu X_i.\{X_j.\phi_j\}_{j \in J}$$

then the set $\{X_j.\phi_j\}_{j \in J}$ is single-orbit.

Proposition 4.12 . *Every formula of vectorial atomic μ-calculus is semantically equivalent to one with fixpoints over single orbit sets of equations.*

Proof. It is sufficient to show that if

$$\mu X_i.\{X_j.\phi_j\}_{j \in J}$$

is a well-built formula and $\{X_j.\phi_j\}_{j \in J}$ has two orbits, then we can split the fixpoint into two nested fixpoints over single-orbit sets of equations. Let

$$\{X_j.\phi_j\}_{j \in J_1} \quad \text{and} \quad \{X_j.\phi_j\}_{j \in J_2}$$

be the two orbits of $\{X_j.\phi_j\}_{j \in J}$. Given an arbitrary Kripke model $\mathcal{K}$ we can see them as monotone functions

$$F_1 : \mathcal{P}(K)^{J_1} \times \mathcal{P}(K)^{J_2} \to \mathcal{P}(K)^{J_1}$$
$$F_2 : \mathcal{P}(K)^{J_1} \times \mathcal{P}(K)^{J_2} \to \mathcal{P}(K)^{J_2}$$

so that the function defined by the whole system $\{X_j.\phi_j\}_{j\in J}$ can be presented as $\langle F_1, F_2 \rangle$ (note that $\mathcal{P}(K)^{J_1} \times \mathcal{P}(K)^{J_2}$ is canonically isomorphic to $\mathcal{P}(K)^J$). By the Bekić principle the least fixpoint of $\langle F_1, F_2 \rangle$ is

$$\mu f.F_1(f, \mu g.F_2(f,g)),$$
$$\mu g.F_2(\mu f.F_1(f,g), g).$$

Without loss of generality suppose that $X_i \in J_1$. Then it follows that $\mu X_i.\{X_j.\phi_j\}_{j\in J}$ is equivalent to

$$\mu X_i.\{X_j.\phi_j'\}_{j\in J_1},$$

where

$$\phi_j' := \phi_j[X_k \mapsto \mu X_k.\{X_l.\phi_l\}_{l\in J_2}]_{k\in J_2}.$$

Note that $\mu X_k.\{X_l.\phi_l\}_{l\in J_2}$ is by definition the k-th projection of the least fixpoint of $\{X_j.\phi_j\}_{j\in J_2}$; $\phi_j[X_k \mapsto \theta_k]_{k\in J_2}$ denotes the simultaneous substitution of all formulas θ_k for the respective variables in ϕ_j. $\qquad\square$

5. Model checking

As we shall see in the following sections, the landscape of modal μ-calculi with atoms is considerably more complex than the classical setting. Some results that hold classically fail here, some others behave differently depending on the chosen structure of atoms. However, some important results from the classical calculus transport to the atomic setting. We begin by proving that this is the case for the decidability of model checking. We state the theorem for the vectorial calculus, and the scalar counterpart follows trivially.

Theorem 5.1. *Model-checking problem for the vectorial atomic μ-calculus $\vec{\mathcal{L}}_\mu^{\mathbb{A}}$ is decidable over orbit-finite atomic Kripke models.*

Proof. Let us fix an orbit-finite atomic Kripke model $\mathcal{K}$. We shall show that the meaning of any formula $\phi \in \vec{\mathcal{L}}_\mu^{\mathbb{A}}$ in $\mathcal{K}$ (under a variable environment ρ) can be computed from ϕ, by structural induction on ϕ and using basic operations listed in Remark 3.4.

In the inductive computation, we shall only ever consider environments ρ that are defined only on an orbit-finite set of variables, and are supported by (the support of $\mathcal{K}$ together with) the support of some subformula of ϕ. Every such environment is presentable by finite means and is amenable to algorithmic manipulation.

The cases of basic propositions, variables, negation and the modality $\Diamond$ are straightforward. For the case of orbit-finite disjunction $\phi = \bigvee \Phi$, first calculate

$$S = \operatorname{supp}(\Phi) \cup \operatorname{supp}(\mathcal{K}) \cup \operatorname{supp}(\rho).$$

Then partition Φ into S-orbits (there are finitely many of them), and select a system of representatives $\phi_1, \ldots, \phi_n$, one from each orbit. Using the inductive assumption, calculate $P_i = \llbracket \phi_i \rrbracket_\rho$ for each i. Then compute the S-orbit $\mathcal{O}_i$ of each P_i; each $\mathcal{O}_i$ is an S-supported family of subsets of K. The union of all $\bigcup \mathcal{O}_i$ is the desired set $\llbracket \phi \rrbracket_\rho$.

The most interesting case is computing

$$\llbracket \mu X_i.\{X_j.\phi_j\}_{j\in J} \rrbracket_\rho.$$

This is done by approximating the least fixpoint by the following standard procedure:

(1) Put $A_j = \emptyset$ for each $j \in J$;

(2) Extend ρ by mapping each variable X_j to A_j;
(3) Using the inductive assumption, calculate $[\![\phi_j]\!]_{\rho[X_k \mapsto A_k]_{k \in J}}$ for each $j \in J$, and assign the resulting tuple as a new value of A;
(4) Repeat steps (2)-(3) until A stabilizes;
(5) Return A_i.

Step (3) above is achieved by choosing an arbitrary representative j from each S-orbit of J, computing the subset $[\![\phi_j]\!]_{\rho[X_k \mapsto A_k]_{k \in J}} \subseteq K$ for each representative, and (uniquely) extending the result to an S-supported relation between J and subsets of K.

The set of all J-indexed tuples of subsets of K is a complete lattice, so by the Knaster-Tarski theorem (see [1]) all we need to show is that this procedure terminates. Notice that, at every stage of computation, the current value of $(A_j)_{j \in J}$ defines a relation between J and K. Moreover, by Lemma 4.9 applied to the vectorial calculus, each such relation is finitely supported by

$$S = \mathrm{supp}(\phi) \cup \mathrm{supp}(\mathcal{K}) \cup \mathrm{supp}(\rho).$$

Since K is orbit-finite, by Proposition 3.5(3) A can take on only finitely many values, therefore the above procedure terminates after finitely many steps. $\qquad\qquad\square$

6. Failure of orbit-finite model property

The classical modal μ-calculus enjoys the so-called finite model property: every satisfiable formula has a finite model. (In fact a stronger *small model property* holds, useful for complexity upper bounds.) There is no chance for this property to hold in the atomic setting, but since orbit-finite sets play the role of finite sets in the universe of sets with atoms, one might hope that an *orbit-finite model property* holds, i.e., that every satisfiable formula in $\mathcal{L}_\mu^{\mathbb{A}}$ has an orbit-finite model.

However, even that weaker property fails even for the scalar μ-calculus $\mathcal{L}_\mu^{\mathbb{A}}$ both for equality and ordered atoms, as we shall now prove. We apply different arguments for the two atom structures.

6.1. **The case of equality atoms.** Over a vocabulary of basic predicates that includes a predicate for each atom a, consider the following two properties:

P1: every reachable state has at least one successor for which some basic predicate holds;
P2: on every path that starts from the current state, no basic predicate holds more than once.

These are definable in the atomic μ-calculus $\mathcal{L}_\mu^=$:

$$\mathbf{P1} \quad \text{is} \quad \nu X. \left(\left(\Diamond \bigvee_{a \in \mathbb{A}} a \right) \wedge \Box X \right),$$

$$\mathbf{P2} \quad \text{is} \quad \neg(\mu X.(\psi \vee \Diamond X)) \qquad \text{where} \qquad \psi = \bigvee_{a \in \mathbb{A}} (a \wedge \Diamond \mu Y.(a \vee \Diamond Y)).$$

The conjunction of **P1** and **P2** is satisfiable. Indeed, it is satisfied in (every state) of the Kripke model from Example 4.6 restricted to those states $w \in \mathbb{A}^*$ where w does not contain any letter more than once. It is easy to see that this is a well-defined equivariant model, and that **P1** and **P2** hold in it. Note that this model still has infinitely many orbits of states, with exactly one orbit for each length of w.

On the other hand, the conjunction of **P1** and **P2** has no orbit-finite models. Indeed, assume that some state x_0 in such a model satisfies both properties. By **P1**, there exists an infinite path in the model:

$$x_0 \longrightarrow x_1 \longrightarrow x_2 \longrightarrow x_3 \longrightarrow x_4 \longrightarrow \cdots .$$

where each state (perhaps except x_0) satisfies some basic predicate. By **P2** no such predicate is satisfied more than once on this path.

Since the model is orbit-finite, there exists a global upper bound on the size of the least supports $\mathrm{supp}(x_i)$. This implies that there exists a number j and distinct atoms a, b such that:
- a holds in some x_i where $i < j$,
- b holds in some x_k where $j < k$, and
- $a, b \notin \mathrm{supp}(x_j)$.

Let $\pi \in \mathrm{Aut}(\mathbb{A})$ be the atom automorphism that swaps a and b and leaves all other atoms untouched; then $x_j \cdot \pi = x_j$, therefore $x_{j-1} \longrightarrow x_j \cdot \pi$ is a valid transition. As a result:

$$x_0 \longrightarrow x_1 \longrightarrow \cdots \longrightarrow x_i \longrightarrow \cdots \longrightarrow x_{j-1} \longrightarrow x_j \cdot \pi \longrightarrow \cdots \longrightarrow x_k \cdot \pi \longrightarrow \cdots$$

is a legal path in the model. But a holds both in x_i and in $x_k \cdot \pi$, so **P2** is violated on this path.

6.2. **The case of ordered atoms.** Properties **P1** and **P2** above make sense also for ordered atoms, and they are defined in $\mathcal{L}_\mu^<$ by the same formulas as before; the ordering relation on atoms is simply not used in those formulas. The orbit-infinite model from before still satisfies both properties.

However, over ordered atoms, the above proof of the lack of an orbit-finite model fails: the bijection π that swaps a and b is not a valid atom automorphism. In fact, over ordered atoms the conjunction of **P1** and **P2** has an orbit-finite model. Indeed, consider a Kripke model with $\mathbb{A}$ as the set of states, the satisfaction relation defined by $a \models a$, and the transition relation by:

$$a \longrightarrow b \text{ if and only if } a < b.$$

It is easy to see that both **P1** and **P2** hold for (every state of) this model.

Consider, however, a stronger version of **P1**, defined with an essential use of the ordering relation between atoms:

P1': for every open interval $(a; b) \subseteq \mathbb{A}$, every reachable state has at least one successor for which some basic proposition $c \in (a; b)$ holds.

This is definable in the atomic μ-calculus $\mathcal{L}_\mu^<$, by the formula

$$\bigwedge_{a < b \in \mathbb{A}} \nu X. \left(\left(\Diamond \bigvee_{c \in (a;b)} c \right) \wedge \Box X \right).$$

The conjunction of **P1'** and **P2** is satisfiable; indeed, it is satisfied in (every state of) the orbit-infinite model that we used to model **P1** and **P2** for equality atoms.

On the other hand, the conjunction of **P1'** and **P2** has no orbit-finite models over ordered atoms. Indeed, assume that some state x_0 in such a model satisfies both properties. By **P1'**, there exists an infinite path in the model:

$$x_0 \longrightarrow x_1 \longrightarrow x_2 \longrightarrow x_3 \longrightarrow x_4 \longrightarrow \cdots .$$

where each state (perhaps except x_0) satisfies some basic predicate. By **P2** no such predicate is satisfied more than once on this path.

Since the model is orbit-finite, there exists a global upper bound on the size of the least supports $\mathrm{supp}(x_i)$. This implies that there exists a number j and an atom d such that:

- d holds in some x_i where $i < j$,
- $d \notin \mathrm{supp}(x_j)$.

Pick some open interval $(a; b)$ that contains d but does not contain any atom from $\mathrm{supp}(x_j)$. By **P1'**, x_j has some successor (call it y) that satisfies some basic proposition $c \in (a; b)$.

Let $\pi \in \mathrm{Aut}(\mathbb{A})$ be an atom automorphism that maps c to d and leaves all atoms from $\mathrm{supp}(x_j)$ untouched; then $x_j \cdot \pi = x_j$, therefore $x_{j-1} \longrightarrow x_j \cdot \pi$ is a valid transition. As a result:

$$x_0 \longrightarrow x_1 \longrightarrow \cdots \longrightarrow x_i \longrightarrow \cdots \longrightarrow x_{j-1} \longrightarrow x_j \cdot \pi \longrightarrow y \cdot \pi$$

is a legal path in the model. But a holds both in x_i and in $y \cdot \pi$, so **P2** is violated on this path.

7. Parity games

An important result in the theory of classical μ-calculus is its correspondence to *parity games*. In this section we show that this correspondence extends, without significant conceptual changes, to the setting with atoms. This has a double purpose. First, since (as we prove) orbit-finite games are decidable, the correspondence gives an alternative route to deciding the model-checking problem for atomic Kripke models. Second, in Section 8 we will use parity games to show that modal μ-formulas are invariant under *atomic bisimulations*, which will be used in Sections 10-11 for proving expressive limitations of atomic μ-calculi.

The following development closely follows analogous definitions and results for the classical, atom-less μ-calculus; see e.g. [7] or [41] for a detailed exposition of that theory.

Definition 7.1. An *atomic parity game* $\mathcal{G}$ is a quadruple $\langle V, V_\exists, R, \Omega \rangle$ such that

(1) V is a set (with atoms) of *nodes*, $V_\exists$ is a finitely supported subset and we define $V_\forall = V \setminus V_\exists$;
(2) $R \subseteq V^2$ is a finitely supported *move* relation;
(3) $\Omega : V \to \mathbb{N}$ is a bounded, finitely supported *rank* function.

The game is called *orbit-finite* if V is orbit-finite.

In a context of a parity game, a *match* (starting at a node v_0) is a sequence, finite or infinite, of nodes:

$$\vec{v} = v_0, v_1, v_2, \ldots$$

such that $\langle v_n, v_{n+1} \rangle \in R$ for each $n \in \mathbb{N}$. A match is *complete* if it cannot be extended to a longer match, i.e., if it is infinite or if it ends with a node that has no successors in R. Otherwise the match is *partial*. A complete match $\vec{v}$ is *won* by $\exists$ if it is infinite and $\limsup_n \Omega(v_n)$ is an even number, or if it is finite and its last node is in $V_\forall$. Otherwise $\vec{v}$ is won by $\forall$.

For a node $v_0 \in V$, let $\mathrm{PM}_\exists(v_0)$ be the set of partial matches that begin in v_0 and end in a node from $V_\exists$. A *strategy* for $\exists$ is a function $\rho : \mathrm{PM}_\exists(v_0) \to V$ such that for each partial match $\vec{v} = \langle v_0, \ldots, v_n \rangle \in \mathrm{PM}_\exists(v_0)$ it holds that $\langle v_n, \rho(\vec{v}) \rangle \in R$. A strategy is *positional* if $\rho(v_0, \ldots, v_n)$ only depends on v_n; positional strategies can be seen as functions from $V_\exists$ to V. A match $\vec{v} = v_0, v_1, v_2, \ldots$ *conforms* to a strategy ρ if for every n such that $v_n \in V_\exists$

there is $v_{n+1} = \rho(v_0, \ldots, v_n)$ ($v_{n+1} = \rho(v_n)$ if ρ is positional). A strategy ρ is *winning* for $\exists$ if every match that conforms to ρ is won by $\exists$. If a positional winning strategy exists for v_0, we say that v_0 is a *winning node* for $\exists$. The set of such nodes is denoted $\mathrm{Win}_\exists(\mathcal{G})$. Strategies and winning strategies for $\forall$, and the set $\mathrm{Win}_\forall(\mathcal{G})$ of nodes winning for $\forall$, are defined analogously.

It is a standard result that every parity game $\mathcal{G}$ is *positionally determined*, i.e., for every node in it either there is a positional winning strategy for $\exists$ or for $\forall$. Atomic parity games are obviously (forgetting about the action of atom automorphisms) parity games in the classical sense, so they are positionally determined. However, it might happen that in an atomic parity game no winning strategy is finitely supported.

Example 7.2. Consider an atomic parity game (over equality atoms) where:

$$V = \binom{\mathbb{A}}{2} \cup \mathbb{A} \qquad V_\exists = \binom{\mathbb{A}}{2} \qquad R = \{\langle \{a,b\}, a \rangle \mid a, b \in \mathbb{A}\} \cup \{\langle a, \{b,c\} \rangle \mid a, b, c \in \mathbb{A}\}$$

$$\Omega(v) = 0 \text{ for all } v \in V.$$

Since $\exists$ wins every infinite play and every state has a successor with respect to R, it is clear that every state is winning for $\exists$. However, no winning strategy for $\exists$ is finitely supported. Indeed, such a strategy would determine a finitely supported function from $\binom{\mathbb{A}}{2}$ to $\mathbb{A}$ such as $f(C) \in C$ for all $C \in \binom{\mathbb{A}}{2}$, and it is easy to see that over equality atoms no such function exists.

In spite of this, winning regions in orbit-finite parity games are computable. Indeed, every orbit-finite game can be effectively transformed into a finite game in the following way. For an orbit-finite parity game $\mathcal{G} = \langle V, V_\exists, R, \Omega \rangle$, let S be any finite set of atoms that supports $\mathcal{G}$. Let V/S, $V_\exists/S$ be the sets of S-orbits of V and $V_\exists$, respectively; let $[v]$ denote the S-orbit of $v \in V$. Obviously $V_\exists/S \subseteq V/S$. Define $R/S \subseteq V/S \times V/S$ and $\Omega/S : V/S \to \mathbb{N}$ by:

$$\langle [v], [w] \rangle \in R/S \quad \text{if} \quad \langle x, y \rangle \in R \text{ for some } x \in [v], y \in [w]$$
$$\Omega/S([v]) = n \quad \text{if} \quad \Omega(x) = n \text{ for some } x \in [v]$$

This is well defined since S supports both R and Ω. In particular, Ω/S is a function. We call $\mathcal{G}/S = \langle V/S, V_\exists/S, R/S, \Omega/S \rangle$ the *orbit game* of $\mathcal{G}$.

Games $\mathcal{G}$ and $\mathcal{G}/S$ are very similar. Indeed, forgetting about the winning condition, the game $\mathcal{G}$ can be seen as a Kripke model over $\mathbb{N}$ as the set of basic predicates: put V as the set of states, R as the transition relation and the satisfaction relation defined by

$$v \models n \text{ if and only if } \Omega(v) = n.$$

The same can be said for $\mathcal{G}/S$. As it turns out, $\mathcal{G}$ and $\mathcal{G}/S$ are bisimilar in the classical sense:

Lemma 7.3. *The quotient function Π defined for every $v \in V$ by $\Pi(v) = [v]$ is a bisimulation between $\mathcal{G}$ and $\mathcal{G}/S$ understood as Kripke models.*

Proof. First, if $\Pi(w) = [v]$, then $w \in [v]$ and consequently all the elements of $[v]$ have label $\Omega(w)$. So, by definition $\Omega/S([v]) = \Omega(w)$.

Now suppose $\Pi(w) = [v]$ and $\langle [v], [z] \rangle \in R/S$. It means that there are $x \in [v]$, $y \in [z]$ such that $\langle x, y \rangle \in R$. Since w and x are in the same S-orbit, pick a $\pi \in \mathrm{Aut}_S(\mathbb{A})$ such that $x \cdot \pi = w$. Since S supports R, we get $\langle w, y \cdot \pi \rangle \in R$. But $y \cdot \pi \in [z]$, so $\Pi(y \cdot \pi) = [z]$.

For the opposite direction, let $\Pi(w) = [v]$ and for some x, $\langle w, x \rangle \in R$. Then by the definition of R/S, $\langle [v], [x] \rangle \in R/S$ and obviously $\Pi(x) = [x]$. $\square$

Moreover, for every $v \in V$, $v \in V_\exists$ if and only if $[v] \in V_\exists/S$. As a result, $\exists$ has a (positional) winning strategy from v in $\mathcal{G}$ if and only if she has a (positional) winning strategy from $[v]$ in $\mathcal{G}/S$. This implies that one can effectively decide whether a player has a winning strategy in an orbit-finite atomic parity game $\mathcal{G}$ by calculating first $S = \mathrm{supp}(\mathcal{G})$, then $\mathcal{G}/S$, and finally solving the analogous problem in the finite parity game obtained, using standard methods.

A correspondence of the atomic μ-calculus with atomic parity games relies on defining, for an atomic modal μ-formula and a Kripke model $\mathcal{K}$, a parity game with atoms $\mathcal{G}_{\phi,\mathcal{K}}$ whose nodes are pairs consisting of (occurrences of) subformulas of ϕ and states of $\mathcal{K}$, such that $x \models^\mathcal{K} \phi$ if and only if $\langle \phi, k \rangle$ is a winning node for $\exists$ in $\mathcal{G}_{\phi,\mathcal{K}}$. We shall show how to do this for the *vectorial* μ-calculus $\vec{\mathcal{L}}_\mu^\mathbb{A}$; the analogous result for the scalar calculus $\mathcal{L}_\mu^\mathbb{A}$ is a special case. Both results are uncomplicated translations of the classical theorem (see e.g. [7, 41]); the main difference is that the classical proof deals only with the scalar calculus (which in the atom-less world is sufficient, because the vectorial calculus can be translated into the scalar one).

First, notice that alternation depth of a formula can be defined exactly as in the classical case (see e.g. [7]). In particular, if

$$\{X_j.\psi_j\}_{j \in J}$$

is an orbite-finite system of equations, then ψ_j are of finitely many syntactical shapes, hence $\mu X_i.\{X_j.\psi_j\}_{j \in J}$ can be assigned a depth as in the atom-less case. The alternation depth of ϕ will be denoted by $\alpha(\phi)$.

We work with a fixed model $\mathcal{K}$, context ρ and a fixed formula ϕ in negation normal form, possibly containing free fixpoint variables. The nodes of $\mathcal{G}_{\phi,\mathcal{K}}$ are pairs of the form $\langle \psi, x \rangle$ where x is a state in $\mathcal{K}$ and ψ is an *occurrence* of a subformula of ϕ (we do not count equations and systems of equations as subformulas; for the sake of simplicity we shall be talking about formulas and not their occurrences, but it must be noted that distinguishing between the two is important for the parity game to be well-defined). This set of nodes is orbit-finite and finitely supported by $\mathrm{supp}(\phi) \cup \mathrm{supp}(\mathcal{K})$.

Nodes *controlled by* $\exists$ are of the following types:

- $\langle X, x \rangle$, where X is a free variable in ϕ and $x \notin \rho(X)$;
- $\langle p, x \rangle$, where p is a basic predicate, and $x \notin \llbracket p \rrbracket_\rho$;
- $\langle \neg p, x \rangle$, where p is a basic predicate and $x \in \llbracket p \rrbracket_\rho$;
- $\langle \bigvee \Phi, x \rangle$;
- $\langle \Diamond \psi, x \rangle$.

Dually, nodes *controlled by* $\forall$ are of the following form

- $\langle X, x \rangle$, where X is a free variable in ϕ and $x \in \rho(X)$;
- $\langle p, x \rangle$, where p is a basic predicate, and $x \in \llbracket p \rrbracket_\rho$;
- $\langle \neg p, x \rangle$, where p is a basic predicate and $x \notin \llbracket p \rrbracket_\rho$;
- $\langle \bigwedge \Phi, x \rangle$;
- $\langle \Box \psi, x \rangle$.

Nodes of the form $\langle \mu X.\Phi, x \rangle$, $\langle \nu X.\Phi, x \rangle$ and $\langle X, x \rangle$ where X is a bound variable, are not controlled by either player. This is not a problem since from each such node there will be only one move available.

Depending on a type of a node, the following *moves* are available:

No move: from a node of type $\langle X, x \rangle$, where X is a free variable, or nodes of type $\langle p, x \rangle$ or $\langle \neg p, x \rangle$ where p is a basic predicate, no moves are available.

Modal move: from a node of type $\langle \Diamond\phi, x\rangle$ or $\langle \Box\phi, x\rangle$ the player that controls the node can move to any node $\langle \phi, y\rangle$ such that $x \longrightarrow^{\mathcal{K}} y$;

Boolean move: from a node of type $\langle \bigwedge \Phi, x\rangle$ or $\langle \bigvee \Phi, x\rangle$, the controlling player can move to any node $\langle \phi, x\rangle$ such that $\phi \in \Phi$;

Automatic move: there are two types of automatic moves:
- from a node of type $\langle \mu X.\Phi, x\rangle$ or $\langle \nu X.\Phi, x\rangle$ the game moves to the node $\langle \psi, x\rangle$ where ψ is the unique formula such that $X.\psi \in \Phi$;
- from a node of type $\langle X, x\rangle$ where X is a variable bound by a (unique) subformula $\mu X'.\Phi$ (or $\nu X'.\Phi$) of ϕ, the game moves to the node $\langle \psi, x\rangle$ where ψ is the unique formula such that the equation $X.\psi$ is in Φ.

Modal and boolean moves above are essentially as in the classical setting. The only substantial difference is for automatic moves: the idea is that players are allowed to (and indeed are forced to) travel across various equations from a given system. When a node with a bound variable X is reached, the game automatically transfers to the X-th equation of the system that binds it.

Nonzero *ranks* are assigned only to nodes of bound variables, and they depend on the alternation depth of the subformulas of ϕ which bind those variables. More precisely, consider a node $\langle X, x\rangle$, where X is bound in ϕ by $\eta X'.\Phi$ (i.e. it occurs on the left-hand side of one of the equations from Φ), for $\eta \in \{\mu, \nu\}$. Then

- if $\eta = \mu$, then the rank of $\langle X, x\rangle$ is $2 \cdot \lfloor \frac{\alpha(\mu X'.\Phi)}{2}\rfloor + 1$ and
- if $\eta = \nu$, then the rank of $\langle X, x\rangle$ is $2 \cdot \lfloor \frac{\alpha(\nu X'.\Phi)}{2}\rfloor$.

Now we are ready to prove

Theorem 7.4 (Adequacy theorem). *For every state x in a Kripke model $\mathcal{K}$ and every formula ϕ in $\vec{\mathcal{L}}_\mu^{\mathbb{A}}$,*

$$x \models^{\mathcal{K}} \phi \iff \langle \phi, x\rangle \in \mathrm{Win}_\exists(\mathcal{G}_{\phi,\mathcal{K}}).$$

Proof. We closely follow the lines of [41, Thm. 3.27]. That proof relies on an auxiliary notion of an *unfolding game*, which needs to be generalized a little to adapt it to the present context, where fixpoints are taken over families of sets indexed by atoms. This requires only a very small adjustment of the original conctruction. Given a monotone operator

$$F : \mathcal{P}(A)^B \to \mathcal{P}(A)^B,$$

we can see it as a monotone operator F' of type

$$\mathcal{P}(B \times A) \to \mathcal{P}(B \times A).$$

With the above definition one quickly checks that

$$\langle b, a\rangle \in \mathrm{lfp}(F') \iff a \in (\mathrm{lfp}(F))_b.$$
$$\langle b, a\rangle \in \mathrm{gfp}(F') \iff a \in (\mathrm{gfp}(F))_b.$$

Now, the definition the unfolding game of F is as in [41] (we sketch it for the reader's convenience): it is a simple parity game with the set of nodes

$$(B \times A) \cup \mathcal{P}(B \times A).$$

Nodes in $B \times A$ are controlled by $\exists$, the remaining nodes are controlled by $\forall$. From a node $\langle b, a\rangle \in B \times A$, the player $\exists$ is allowed to move to any set $C \subseteq B \times A$ such that

$$\langle b, a\rangle \in F(C).$$

Conversely, given a set $C \subseteq B \times A$, the player $\forall$ chooses any $\langle b, a \rangle$ such that

$$\langle b, a \rangle \in C.$$

The winning condition depends on whether we want to characterize the greatest or the least fixpoint of F. In the former case, every infinite game is won by $\exists$ (i.e. all positions are assigned rank 0), in the latter, such games are won by $\forall$ (i.e. all positions are assigned rank 1). In both cases the player who got stuck, loses.

The unfolding game for the least fixpoint of F will be denoted by $\mathcal{U}_F^\mu$; the game for the greatest fixpoint by $\mathcal{U}_F^\nu$.

The following proposition is proved by a mix of routine fixpoint arguments; see the proof of [41, Thm. 3.14].

Proposition 7.5 . *For a monotone $F : \mathcal{P}(A)^B \to \mathcal{P}(A)^B$, let lfp(F) and gfp(F) denote the least and the greatest fixpoints of F. For any $a \in A$ and $b \in B$,*
- *$\langle a, b \rangle \in \mathrm{Win}_\exists(\mathcal{U}_F^\mu)$ if and only if $a \in (lfp(F))_b$, and*
- *$\langle a, b \rangle \in \mathrm{Win}_\exists(\mathcal{U}_F^\nu)$ if and only if $a \in (gfp(F))_b$.* $\square$

As in [41, Thm. 3.27], a proof of Theorem 7.4 follows by induction on the structure of the formula ϕ. At each step, allowing a nonempty but orbit-finite set $\{X_j\}_{j \in J}$ of free variables and assuming their interpretations $\{A_j\}_{j \in J}$ as subsets $A_j \subseteq K$, one proves that:

$$x \in \llbracket \phi \rrbracket_{\{X_j \mapsto A_j\}_{j \in J}} \iff \langle \phi, x \rangle \in \mathrm{Win}_\exists(\mathcal{G}_{\phi,\mathcal{K}[X_j \mapsto A_j]_{j \in J}}). \tag{7.1}$$

The only case that differs slightly from [41] is the fixpoint operator: once again we have to adapt the proof to the case of vectorial calculus. Assume that $\phi = \eta X_i.\Phi$ where $\eta \in \{\mu, \nu\}$ and

$$\Phi := \{X_j.\psi_j\}_{j \in J}$$

is an orbit-finite set of equations with $i \in J$. The monotone operator induced by Φ on $\mathcal{P}(K)^J$ will be denoted by $\llbracket \Phi \rrbracket$. For every $j \in J$ we put

$$\phi^j := \eta X_j.\Phi.$$

(Note that $\phi^i = \phi$.) By our inductive assumption, for every j the condition (7.1) holds for ψ_j. By the definition of $\llbracket \eta X.\Phi \rrbracket$ and by Proposition 7.5 it is sufficient to demonstrate that for every $x \in K$ and $j \in J$:

$$\langle x, j \rangle \in \mathrm{Win}_\exists(\mathcal{U}_{\llbracket \Phi \rrbracket}^\eta) \iff \langle \phi^j, x \rangle \in \mathrm{Win}_\exists(\mathcal{G}_{\phi^j,\mathcal{K}}).$$

The proof of the above fact (for the scalar case) given in [41, Thm. 3.27] proceeds by translating winning strategies for $\exists$ between the two games. This proof adapts straightforwardly to our case. $\square$

We should remark that in the above proof, the changes that we introduced were driven by the vectorial character of our formalism rather than directly by the use of atoms. One would have to make essentially the same changes in the original proof (i.e. [41, Thm. 3.27]) to make it work for the standard (i.e. atom-less) vectorial μ-calculus. As we already mentioned, in the atom-less setting this can be avoided since the atom-less vectorial calculus can be encoded into the scalar one. That encoding fails in the presence of atoms, as we shall show in Section 10.

8. ATOMIC BISIMULATION GAMES

In Sections 10-11 we shall show that atomic μ-calculi cannot define various properties of Kripke models. A general strategy for proving such results is using suitable notions of bisimulation: first prove that bisimilar states cannot be distinguished by any formula, then show a Kripke model and two bisimilar states p, q in it such that p has property P but q does not, to conclude that the property P is not definable by any formula.

In this section, to prepare the ground for such undefinability results, we define two notions of bisimulation (or, more precisely, two hierarchies of bisimulations parametrized by natural numbers). The first, which we call *stack bisimulation*, is suitable for scalar μ-calculi with atoms. The second, simply called *bisimulation*, is suitable for the vector calculi.

Denote by $\mathbb{A}^{(\leq k)}$ the set of ordered tuples of atoms of length at most k. Elements of such sets will be denoted with vector notation: $\vec{a}, \vec{b}$ etc. If $|\vec{a}| > 0$, let $\mathrm{pop}(\vec{a})$ denote the tuple $\vec{a}$ with the last element removed. Recall that we write $x \sim y$ to say that x and y are in the same orbit.

Definition 8.1. For a number $k \in \mathbb{N}$, a *k-stack-bisimulation* on a Kripke model $\mathcal{K}$ is a symmetric relation B on $K \times \mathbb{A}^{(\leq k)}$ such that, whenever $\langle x, \vec{a} \rangle B \langle y, \vec{b} \rangle$ then:

   (i) $\langle \mathrm{pred}(x), \vec{a} \rangle \sim \langle \mathrm{pred}(y), \vec{b} \rangle$ (and in particular $|\vec{a}| = |\vec{b}|$),
  (ii) for every x' such that $x \longrightarrow x'$ there is a y' such that $y \longrightarrow y'$ and $\langle x', \vec{a} \rangle B \langle y', \vec{b} \rangle$,
 (iii) if $|\vec{a}| > 0$ then $\langle x, \mathrm{pop}(\vec{a}) \rangle B \langle y, \mathrm{pop}(\vec{b}) \rangle$,
 (iv) if $|\vec{a}| < k$ then for every $c \in \mathbb{A}$ there exists a $d \in \mathbb{A}$ such that $\langle x, \vec{a}c \rangle B \langle y, \vec{b}d \rangle$.

Two states are called *k-stack-bisimilar* if they are related by a k-stack-bisimulation.

Formal definitions of particular bisimulations are often rather dry, and it is convenient to prove their existence using simple bisimulation games. A *k-stack-bisimulation game* on a model $\mathcal{K}$ is played between two players called Spoiler and Duplicator, who make their moves alternately. A legal position in the game is a tuple

$$\langle x, \vec{a}, y, \vec{b} \rangle \in K \times \mathbb{A}^{(\leq k)} \times K \times \mathbb{A}^{(\leq k)}$$

such that $\langle \mathrm{pred}(x), \vec{a} \rangle \sim \langle \mathrm{pred}(y), \vec{b} \rangle$ (and in particular $|\vec{a}| = |\vec{b}|$). In such a position, Spoiler can make three kinds of moves. He can either

- make a *model move*:
  - choose a state $x' \in K$ such that $x \longrightarrow x'$, to which Duplicator must respond by choosing a state $y' \in K$ such that $y \longrightarrow y'$; or
  - choose a state $y' \in K$ such that $y \longrightarrow y'$, to which Duplicator must respond by choosing a state $x' \in K$ such that $x \longrightarrow x'$;

  the game then proceeds from $\langle x', \vec{a}, y', \vec{b} \rangle$ provided that it is a legal position; or
- make a *pop move*, assuming that $|\vec{a}| > 0$:
  - remove the last element from $\vec{a}$, to which Duplicator must respond by removing the last element of $\vec{b}$; or
  - remove the last element from $\vec{b}$, to which Duplicator must respond by removing the last element of $\vec{a}$;

  the game then proceeds from $\langle x, \mathrm{pop}(\vec{a}), y, \mathrm{pop}(\vec{b}) \rangle$, which is always a legal position; or
- make a *push move*, assuming that $|\vec{a}| < k$:
  - extend $\vec{a}$ with some atom c, to which Duplicator must respond by extending $\vec{b}$ with some atom d; or

– extend $\vec{b}$ with some atom d, to which Duplicator must respond by extending $\vec{a}$ with some atom c;

the game then proceeds from $\langle x, \vec{a}c, y, \vec{b}d \rangle$, provided that it is a legal position.

Duplicator loses if she fails to move the game to a legal position in response to a Spoiler move. Since Spoiler can always make a pop move or a push move, the only way for Duplicator to win is to achieve an infinite play.

By a standard argument, $\langle x, \vec{a} \rangle$ and $\langle y, \vec{b} \rangle$ are k-stack-bisimilar if and only if Duplicator has a winning strategy in the k-stack-game from the position $\langle x, \vec{a}, y, \vec{b} \rangle$.

We will now show that k-stack-bisimilar states cannot be distinguished by formulas from a certain fragment of the atomic μ-calculus $\mathcal{L}_\mu^{\mathbb{A}}$. To describe this fragment, it is convenient to introduce a notion of an "ordered" global support of a subformula.

Given a fixed equivariant formula ϕ, we define a relation between finite sequences of atoms and (occurrences of) subformulas of ϕ. For $\vec{a} \in \mathbb{A}^*$ and $\psi \trianglelefteq \phi$, we write $\vec{a} \blacktriangleright_\phi \psi$ to read "$\vec{a}$ globally supports ψ within ϕ". This relation is defined by induction on the depth of the occurrence of ψ within ϕ:

- $\vec{a} \blacktriangleright_\phi \phi$ if and only if $\vec{a} = \epsilon$,
- if $\psi \trianglelefteq \phi$ such that $\psi = \Diamond\psi'$, $\psi = \Box\psi'$, $\psi = \mu X.\psi'$ or $\psi = \nu X.\psi'$ then $\vec{a} \blacktriangleright_\phi \psi'$ if and only if $\vec{a} \blacktriangleright_\phi \psi$,
- if $\psi \trianglelefteq \phi$ such that $\psi = \bigvee \Psi$ or $\psi = \bigwedge \Psi$ and $\psi' \in \Psi$, then $\vec{e} \blacktriangleright_\phi \psi'$ if and only if there is a (necessarily unique, for a given $\vec{e}$) decomposition $\vec{e} = \vec{a}^\frown \vec{c}$ such that:
  - $\vec{a} \blacktriangleright_\phi \psi$ and
  - $\vec{c}$ is some ordering of the set $\mathrm{supp}(\psi')$.

Some basic properties of global supports follow directly from the definition. Every (occurrence of) subformula $\psi \trianglelefteq \phi$ is globally supported by some sequence, possibly more than one, since the least supports of each formula on the path trom ψ to the root of ϕ may be ordered in many ways. However, all global supports of any given ψ within ϕ have the same length. Moreover, for subformulas $\psi \trianglelefteq \theta \trianglelefteq \phi$,

- every global support of ψ within ϕ has a (unique) prefix that globally supports θ within ϕ, and
- every global support of θ within ϕ extends (not necessarily uniquely) to a global support of ψ within ϕ.

It is also easy to see that if $\vec{a} \blacktriangleright_\phi \psi$ then the set of all atoms in $\vec{a}$ supports ψ, not only as a formula but also as its occurrence within ϕ.

Call a formula ϕ *globally k-supported* if every subformula $\psi \trianglelefteq \phi$ has a global support of length at most k. It is easy to see that every formula ϕ in $\mathcal{L}_\mu^{\mathbb{A}}$, or even in $\vec{\mathcal{L}}_\mu^{\mathbb{A}}$, is globally k-supported for some number k, bounded from above by the height of ϕ multiplied by the maximal size of the least support of a subformula of ϕ.

The following theorem is formulated for equivariant models and formulas, but it is not difficult to generalize it to ones supported by some fixed set S.

Theorem 8.2. *Assume that $\langle x, \epsilon \rangle$ and $\langle y, \epsilon \rangle$ are k-stack-bisimilar in an equivariant model $\mathcal{K}$. For every equivariant, globally k-supported formula ϕ in $\mathcal{L}_\mu^{\mathbb{A}}$, $x \models \phi$ if and only if $y \models \phi$.*

Proof. Fix ϕ, x and y as in the assumptions. We will define a bisimulation relation B (in the classical sense) on the graph of the game $\mathcal{G}_{\phi,\mathcal{K}}$ such that

$$\langle \phi, x \rangle B \langle \phi, y \rangle. \tag{8.1}$$

This theorem will then follow by Theorem 7.4. Note that the graph of $\mathcal{G}_{\phi,\mathcal{K}}$ can be seen as a Kripke model where labels correspond to ranks of nodes together with the information about which player controls which node. Since $\mathcal{K}$ is fixed we will skip the reference to it.

The bisimulation B is defined by: $\langle \psi, x' \rangle B \langle \theta, y' \rangle$ if and only if

(1) there exists $\pi \in \mathrm{Aut}(\mathbb{A})$ such that (the occurrence) ψ in ϕ is mapped by π to (the occurrence) θ in ϕ (note that $\phi \cdot \pi = \phi$ since ϕ is equivariant), and
(2) there exist $\vec{a}, \vec{b} \in \mathbb{A}^{(\leq k)}$ such that:
   - $\vec{a} \blacktriangleright_{\phi} \psi$,
   - $\vec{b} \blacktriangleright_{\phi} \theta$,
   - $\langle x', \vec{a} \rangle$ and $\langle y', \vec{b} \rangle$ are k-stack bisimilar.

Under this definition (8.1) indeed holds: it is enough to put the identity automorphism as π and $\vec{a} = \vec{b} = \epsilon$.

We need to verify that B is a bisimulation on $\mathcal{G}_{\phi}$. Note that if $\langle \psi, x' \rangle B \langle \theta, y' \rangle$ then both pairs get the same label (i.e. rank) in $\mathcal{G}_{\phi}$, since the label of a formula depends only on its syntactical properties (i.e. alternation depth), which are preserved by application of atom automorphims.

Now we verify the zig-zag bisimulation condition for B. Since B is clearly symmetric, it is enough to verify one direction only. Assume that

$$\langle \psi, x' \rangle B \langle \theta, y' \rangle \qquad \text{and} \qquad \langle \psi, x' \rangle \longrightarrow^{\mathcal{G}_{\phi}} \langle \psi', x'' \rangle.$$

We are looking for $\langle \theta', y'' \rangle$ such that

$$\langle \psi', x'' \rangle B \langle \theta', y'' \rangle \qquad \text{and} \qquad \langle \theta, y' \rangle \longrightarrow^{\mathcal{G}_{\phi}} \langle \theta', y'' \rangle.$$

We cover three non-trivial cases:

(1) $\psi = \Diamond \xi$ (the case $\psi = \Box \xi$ is similar),
(2) $\psi = \bigvee \Psi$ (the case $\psi = \bigwedge \Psi$ is similar), or
(3) $\psi = X$ (for a bound X).

Case 1. In such a situation $\psi' = \xi$ and $x' \longrightarrow^{\mathcal{K}} x''$. Moreover, $\theta = \Diamond \theta'$ for some formula θ'. Since $\langle \psi, x' \rangle B \langle \theta, y' \rangle$, there are some sequences $\vec{a}, \vec{b} \in \mathbb{A}^{(\leq k)}$ such that $\vec{a} \blacktriangleright_{\phi} \psi$, $\vec{b} \blacktriangleright_{\phi} \theta$ and $\langle x', \vec{a} \rangle$ and $\langle y', \vec{b} \rangle$ are k-stack bisimilar. By condition (ii) in Definition 8.1, there is some state y'' in $\mathcal{K}$ such that $y' \longrightarrow^{\mathcal{K}} y''$ (hence $\langle \theta, y' \rangle \longrightarrow^{\mathcal{G}_{\phi}} \langle \theta', y'' \rangle$ as required) and $\langle x'', \vec{a} \rangle$ and $\langle y'', \vec{b} \rangle$ are k-stack bisimilar. By definition of the global support relation we have $\vec{a} \blacktriangleright_{\phi} \psi'$, $\vec{b} \blacktriangleright_{\phi} \theta'$, hence $\langle \psi', x'' \rangle B \langle \theta', y'' \rangle$ (the automorphism π that was good for the pair ψ, θ remains good for ψ', θ').

Case 2. In such a situation $x'' = x'$ and $\psi' \in \Psi$. Moreover, $\theta = \bigvee \Theta$ for some orbit-finite set Θ of formulas. Since $\langle \psi, x' \rangle B \langle \theta, y' \rangle$, there are some sequences $\vec{a}, \vec{b} \in \mathbb{A}^{(\leq k)}$ such that $\vec{a} \blacktriangleright_{\phi} \psi$, $\vec{b} \blacktriangleright_{\phi} \theta$ and $\langle x', \vec{a} \rangle$ and $\langle y', \vec{b} \rangle$ are k-stack bisimilar. Let $\vec{c}$ be any ordering of the finite set $\mathrm{supp}(\psi')$. Then $\vec{a} ^\frown \vec{c} \blacktriangleright_{\phi} \psi'$, and since by our assumption ϕ is globally k-supported, the length of $\vec{a} ^\frown \vec{c}$ is at most k.

Since $\langle x', \vec{a} \rangle$ and $\langle y', \vec{b} \rangle$ are k-stack bisimilar, by repeated application of condition (iv) in Definition 8.1, there is a sequence $\vec{d} \in \mathbb{A}^{*}$ such that

$$\langle x', \vec{a} ^\frown \vec{c} \rangle \text{ and } \langle y', \vec{b} ^\frown \vec{d} \rangle \text{ are } k\text{-stack bisimilar.}$$

This means in particular that $\vec{a}^\frown \vec{c} \sim \vec{b}^\frown \vec{d}$. This implies that there exists some $\pi' \in \mathrm{Aut}(\mathbb{A})$ such that:

$$\psi \cdot \pi' = \theta \quad (\text{hence } \Psi \cdot \pi' = \Theta), \qquad \vec{a} \cdot \pi' = \vec{b} \qquad \text{and} \qquad \vec{c} \cdot \pi' = \vec{d}.$$

Put $\theta' = \psi' \cdot \pi'$ and $y'' = y'$. Then

$$\theta' \in \Theta \qquad \text{and} \qquad \vec{b}^\frown \vec{d} \blacktriangleright_\phi \theta'.$$

As a result,

$$\langle \psi', x'' \rangle B \langle \theta', y'' \rangle \qquad \text{and} \qquad \langle \theta, y' \rangle \longrightarrow^{\mathcal{G}_\phi} \langle \theta', y'' \rangle$$

as required.

Case 3. In this situation $x'' = x'$ and $\psi \trianglelefteq \psi' = \mu X.\zeta$ is the subformula of ϕ that binds X. Since π maps ψ to θ as an occurrence of a subformula, we get that

$$\theta = X \cdot \pi \trianglelefteq \theta' = \mu(X \cdot \pi).(\zeta \cdot \pi) \trianglelefteq \phi.$$

Clearly, $\psi' \cdot \pi = \theta'$, hence θ' is the subformula of ϕ that binds $X \cdot \pi$ and $\langle \theta, y' \rangle \longrightarrow^{\mathcal{G}_\phi} \langle \theta', y' \rangle$.

Since $\langle \psi, x' \rangle B \langle \theta, y' \rangle$, there are some sequences $\vec{a}, \vec{b} \in \mathbb{A}^{(\leq k)}$ such that $\vec{a} \blacktriangleright_\phi \psi$, $\vec{b} \blacktriangleright_\phi \theta$ and $\langle x', \vec{a} \rangle$ and $\langle y', \vec{b} \rangle$ are k-stack bisimilar. Let $\vec{c}$ be the (unique) prefix of $\vec{a}$ such that $\vec{c} \blacktriangleright_\phi \psi'$, and let $\vec{d}$ be the prefix of $\vec{b}$ with the same length as $\vec{c}$. Then $\vec{d} \blacktriangleright_\phi \theta'$. Indeed, by equivariance we have $(\vec{c} \cdot \pi) \blacktriangleright_\phi \theta'$, so all sequences that globally support θ' within ϕ must have the same length as $\vec{c}$; moreover, since $\vec{b} \blacktriangleright_\phi \theta$ we know that some prefix of $\vec{b}$ must globally support θ', and $\vec{d}$ is the prefix of $\vec{b}$ of the same length as $\vec{c}$.

By repeated use of condition (iii) in Definition 8.1, putting $y'' = y'$, we get that $\langle x'', \vec{c} \rangle$ and $\langle y'', \vec{d} \rangle$ are k-stack bisimilar. From this we obtain

$$\langle \psi', x'' \rangle B \langle \theta', y'' \rangle \qquad \text{and} \qquad \langle \theta, y' \rangle \longrightarrow^{\mathcal{G}_\phi} \langle \theta', y'' \rangle$$

as required. $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\square$

Note that every formula in $\mathcal{L}_\mu^{\mathbb{A}}$ is globally k-supported for some number k. Therefore, by Theorem 8.2, to prove that a property P is not definable it is enough to construct, for every number k, a Kripke model $\mathcal{K}$ over $\mathbb{A}$ as the language of basic predicate symbols, and two states $x, y \in K$ such that $\langle x, \epsilon \rangle$ and $\langle y, \epsilon \rangle$ are k-stack-bisimilar and P fails for x but holds for y.

As we shall see in Section 10, formulas of vectorial atomic μ-calculus $\vec{\mathcal{L}}_\mu^{\mathbb{A}}$ can distinguish between stack-bisimilar states in a Kripke model. Therefore, to show undefinability results for vector calculi, we shall now introduce a finer notion of bisimulation, where in the corresponding bisimulation games Spoiler can replace atoms in the current position at will, not respecting the stack regime of Definition 8.1.

Definition 8.3. For a number $k \in \mathbb{N}$, a *k-bisimulation* on a Kripke model $\mathcal{K}$ is a symmetric relation B on $K \times \mathbb{A}^{(\leq k)}$ such that, whenever $\langle x, \vec{a} \rangle B \langle y, \vec{b} \rangle$ then:

  (i) $\langle \mathrm{pred}(x), \vec{a} \rangle \sim \langle \mathrm{pred}(y), \vec{b} \rangle$,

  (ii) for every x' such that $x \longrightarrow x'$ there is a y' such that $y \longrightarrow y'$ and $\langle x', \vec{a} \rangle B \langle y', \vec{b} \rangle$, and

  (iii) for every $\vec{c} \in \mathbb{A}^{(\leq k)}$ there exists a $\vec{d} \in \mathbb{A}^{(\leq k)}$ such that $\langle \vec{a}, \vec{c} \rangle \sim \langle \vec{b}, \vec{d} \rangle$ and $\langle x, \vec{c} \rangle B \langle y, \vec{d} \rangle$.

Two states are called *k-bisimilar* if they are related by a k-bisimulation.

As for stack bisimulations above, k-bisimilarity on a Kripke model is an equivalence relation. If shorter tuples $\vec{a}'$ and $\vec{b}'$ arise from $\vec{a}$ and $\vec{b}$ respectively by selecting the same subset of positions, then $\langle x, \vec{a}' \rangle$ and $\langle y, \vec{b}' \rangle$ are k-bisimilar as well. Finally, for $l < k$, the restriction of a k-bisimulation to the set $K \times \mathbb{A}^{(\leq l)}$ is an l-bisimulation.

As before, a *k-bisimulation game* on a model $\mathcal{K}$ is played between Spoiler and Duplicator, and a legal position in the game is a tuple

$$\langle x, \vec{a}, y, \vec{b} \rangle \in K \times \mathbb{A}^{(\leq k)} \times K \times \mathbb{A}^{(\leq k)}$$

such that $\langle \mathrm{pred}(x), \vec{a} \rangle \sim \langle \mathrm{pred}(y), \vec{b} \rangle$. In such a position, Spoiler can make two kinds of moves. He can either

- make a *model move*:
  - choose a state $x' \in K$ such that $x \longrightarrow x'$, to which Duplicator must respond by choosing a state $y' \in K$ such that $y \longrightarrow y'$; or
  - choose a state $y' \in K$ such that $y \longrightarrow y'$, to which Duplicator must respond by choosing a state $x' \in K$ such that $x \longrightarrow x'$;

  the game then proceeds from $\langle x', \vec{a}, y', \vec{b} \rangle$ provided that it is a legal position; or
- make an *atom replacement move*:
  - choose any tuple $\vec{c} \in \mathbb{A}^{(\leq k)}$, to which Duplicator must respond by choosing a tuple $\vec{d} \in \mathbb{A}^{(\leq k)}$ such that $\langle \vec{a}, \vec{c} \rangle \sim \langle \vec{b}, \vec{d} \rangle$; or
  - choose any tuple $\vec{d} \in \mathbb{A}^{(\leq k)}$, to which Duplicator must respond by choosing a tuple $\vec{c} \in \mathbb{A}^{(\leq k)}$ such that $\langle \vec{a}, \vec{c} \rangle \sim \langle \vec{b}, \vec{d} \rangle$;

  the game then proceeds from $\langle x, \vec{c}, y, \vec{d} \rangle$ provided that it is a legal position.

Duplicator loses if she fails to move the game to a legal position in response to a Spoiler move. Since Spoiler can always make an atom replacement move, the only way for Duplicator to win is to achieve an infinite play.

By a standard argument, $\langle x, \vec{a} \rangle$ and $\langle y, \vec{b} \rangle$ are k-bisimilar if and only if Duplicator has a winning strategy from the position $\langle x, \vec{a}, y, \vec{b} \rangle$.

The following result shows that k-bisimilar states cannot be distinguished by formulas from the globally k-supported fragment of the vectorial atomic μ-calculus $\vec{\mathcal{L}}_\mu^{\mathbb{A}}$. As for Theorem 8.2, is not difficult to generalize it to models and formulas supported by some fixed set S.

Theorem 8.4. *Assume that $\langle x, \epsilon \rangle$ and $\langle y, \epsilon \rangle$ are k-bisimilar in an equivariant model $\mathcal{K}$. For every equivariant, globally k-supported formula ϕ, $x \models \phi$ if and only if $y \models \phi$.*

Proof. The idea is the same as in the proof of Theorem 8.2: for any globally k-supported formula ϕ in $\vec{\mathcal{L}}_\mu^{\mathbb{A}}$, we define a bisimulation B on the graph of the game $\mathcal{G}_{\phi,\mathcal{K}}$ such that $\langle \phi, x \rangle$ and $\langle \phi, y \rangle$ are related by b.

The relation B is defined exactly as before, with k-stack bisimulation replaced by k-bisimulation. In checking the bisimulation condition, the only difference is the case of bound variables (i.e. Case 3), arising from the fact that the binding fixpoint constructions are now richer than those in the scalar calculus. Assume that

$$\langle X, x' \rangle B \langle \theta, y' \rangle$$

and X is a variable bound in $\mu Y.\Phi \trianglelefteq \phi$. By definition of B, there is some $\pi \in \mathrm{Aut}(\mathbb{A})$ such that $X \cdot \pi = \theta$, so θ is a variable bound in some $\mu(Y \cdot \pi).(\Phi \cdot \pi) \trianglelefteq \phi$.

The game $\mathcal{G}_{\phi,\mathcal{K}}$ makes an automatic move from the node $\langle X, x' \rangle$ to the node $\langle \psi', x' \rangle$, where ψ' is the unique formula such that $X.\psi' \in \Phi$. Note, however, that X is not necessarily a subformula of ψ', so there is no guarantee that the automatic move from $\langle \theta, y' \rangle$ goes to the node $\langle \psi' \cdot \pi, y' \rangle$.

Since $\langle X, x' \rangle B \langle \theta, y' \rangle$, there are some sequences $\vec{a}, \vec{b} \in \mathbb{A}^{(\leq k)}$ such that $\vec{a} \blacktriangleright_\phi X$, $\vec{b} \blacktriangleright_\phi \theta$ and $\langle x', \vec{a} \rangle$ and $\langle y', \vec{b} \rangle$ are k-bisimilar. Moreover, let $\vec{c} \in A^{(\leq k)}$ be such that $\vec{c} \blacktriangleright_\phi \psi'$. Note that it may not be possible to choose $\vec{c}$ which is a prefix of $\vec{a}$; this is the crucial difference from the proof of Theorem 8.2.

Since $\langle x', \vec{a} \rangle$ and $\langle y', \vec{b} \rangle$ are k-bisimilar, thanks to condition (iii) in Definition 8.3 we can choose $\vec{d}$ such that $\langle \vec{a}, \vec{c} \rangle \sim \langle \vec{b}, \vec{d} \rangle$ and such that $\langle x', \vec{c} \rangle$ and $\langle y', \vec{d} \rangle$ are k-bisimilar. So pick some $\tau \in \mathrm{Aut}(\mathbb{A})$ such that $\vec{a} \cdot \tau = \vec{b}$ and $\vec{c} \cdot \tau = \vec{d}$. By definition of $\vec{a}$ we have that

$$X \cdot \tau = X \cdot \pi = \theta \qquad \text{and} \quad \Phi \cdot \tau = \Phi \cdot \pi.$$

As a result, $\psi' \cdot \tau$ is the formula such that the equation

$$\theta.(\psi' \cdot \tau)$$

is in $\Phi \cdot \tau$. Also, since $\vec{c} \blacktriangleright_\phi \psi'$, hence $\vec{d} \blacktriangleright_\phi (\psi' \cdot \tau)$. Hence putting $\theta' = \psi' \cdot \tau$ and $y'' = y'$ we are done. $\qquad\square$

As we already observed, every formula of the vectorial atomic μ-calculus $\vec{\mathcal{L}}_\mu^{\mathbb{A}}$ is globally k-supported for some number k. Therefore, by Theorem 8.4, to prove that a property P is not definable it is enough to construct, for every number k, an orbit-finite Kripke model $\mathcal{K}$ over $\mathbb{A}$ as the language of basic predicate symbols, and two states $x, y \in K$ such that $\langle x, \epsilon \rangle$ and $\langle y, \epsilon \rangle$ are k-bisimilar and P fails for x but holds for y.

9. Undecidability results

The main purpose of this section is to show that the satisfiability of atomic μ-calculus formulas is undecidable, even for the scalar calculus $\mathcal{L}_\mu^{\mathbb{A}}$. We start by showing that satisfiability of atomic LTL formulas is undecidable and end with the same conclusion about the model checking problem for CTL*. All proofs in this section work the same both for equality and ordered atoms.

Theorem 9.1. *It is undecidable whether a given atomic LTL formula is satisfiable.*

Proof. The proof follows the lines of the proof from [34] of the undecidability of the universality problem for register automata.

Given a Turing machine $\mathcal{M}$ with a (finite) set of states Q, a (finite) working alphabet Γ (including a blank symbol $\flat$), initial and accepting states $q_{init}, q_{acc} \in Q$ and a transition relation δ, we shall build an atomic LTL formula that is satisfiable if and only if $\mathcal{M}$ accepts the empty word. Without loss of generality assume that $\mathcal{M}$, upon reaching an accepting configuration, enters an infinite loop in that configuration.

Consider the following language of basic predicate symbols:
- a special symbol \$,
- $\mathtt{atom}_a$ for each $a \in \mathbb{A}$,
- $\mathtt{tape}_\gamma$ for each $\gamma \in \Gamma$,
- $\mathtt{head}_q$ for each $q \in Q$.

Our formula shall be a conjunction of several properties. First, we enforce that every state in a model either satisfies \$ (and no other basic predicates) or satisfies exactly one predicate $\texttt{atom}_a$, exactly one predicate $\texttt{tape}_\gamma$ and at most one predicate $\texttt{head}_q$. This is ensured by a conjunction of formulas such as:

$$\mathsf{G}\left(\$ \vee \bigvee_{a \in \mathbb{A}} \texttt{atom}_a\right) \quad \text{and} \quad \bigwedge_{a \neq b \in \mathbb{A}} \mathsf{G}(\texttt{atom}_a \to \neg\texttt{atom}_b)$$

and so on. We also ensure that the initial state of the model satisfies \$.

Furthermore, we ensure that as far as predicates \$ and $\texttt{atom}_a$ are concerned, the model can be presented as an infinite word:

$$\$w\$w\$w \cdots$$

where w is a finite sequence of predicate symbols of the form $\texttt{atom}_\mathtt{a}$ such that no single predicate appears in w more than once. This is achieved by a conjunction of formulas such as:

- $\bigwedge_{a,b \in \mathbb{A}} \mathsf{G}(\texttt{atom}_a \wedge \mathsf{X}\texttt{atom}_b \to \mathsf{G}(\texttt{atom}_a \to \mathsf{X}\texttt{atom}_b))$
- $\bigwedge_{a \in \mathbb{A}} \mathsf{G}(\texttt{atom}_a \to (\neg\texttt{atom}_a\mathsf{U}\$))$.

Portions of the model between two consecutive occurrences of \$ will store configurations of $\mathcal{M}$. To this end, we ensure that each portion contains exactly one state that satisfies some $\texttt{head}$ predicate:

$$\mathsf{G}(\$ \to \mathsf{X}((\neg\psi \wedge \neg\$)\mathsf{U}(\psi \wedge \mathsf{X}(\neg\psi\mathsf{U}\$))))$$

where $\psi = \bigvee_{q \in Q} \texttt{head}_q$.

We then ensure that every two consecutive configurations encode a legal step of $\mathcal{M}$. Predicates $\texttt{atom}$ are useful for this, as they trace single tape cells in subsequent configurations. To ensure that the letters on the tape do not change unless the machine head is directly over them, we state for each $\gamma \in \Gamma$:

$$\bigwedge_{a \in \mathbb{A}} \mathsf{G}(\texttt{atom}_a \wedge \texttt{tape}_\gamma \wedge \neg\psi \to \mathsf{X}(\neg\texttt{atom}_a\mathsf{U}(\texttt{atom}_a \wedge \texttt{tape}_\gamma)))$$

where ψ is as before.

Furthermore, for every "head to the right" rule

$$\langle q, \gamma, q', \gamma', \Rightarrow\rangle \in \delta$$

we add a formula

$$\bigwedge_{a \in \mathbb{A}} \mathsf{G}(\texttt{atom}_a \wedge \texttt{head}_q \wedge \texttt{tape}_\gamma \to \mathsf{X}(\neg\texttt{atom}_a\mathsf{U}(\texttt{atom}_a \wedge \texttt{tape}_{\gamma'} \wedge \mathsf{X}\texttt{head}_{q'})))$$

and similarly for "head to the left" transition rules.

Finally we ensure the correct form of the initial configuration and that an accepting state is reached, by adding formulas $\mathsf{X}\texttt{head}_{\mathtt{q_{init}}}$, $\mathsf{X}(\texttt{tape}_\flat\mathsf{U}\$)$ and $\mathsf{F}\texttt{head}_{q_{acc}}$.

Models of the conjunction of all these formulas correspond to accepting runs of $\mathcal{M}$ on the empty input word. As a result, it is undecidable whether an LTL formula has a model. $\qquad\square$

Properties used in the above proof can be formulated in atomic μ-calculus as well. As an immediate result, it is undecidable whether a given atomic μ-calculus formula has a *deterministic* model. Some more care is needed to prove that general satisfiability is undecidable, but:

Theorem 9.2. *It is undecidable whether a given formula of the atomic μ-calculus is satisfiable.*

Proof. For any LTL formula ϕ in the negation normal form (i.e., one where negations occur only in front of basic predicates, and where conjunction, disjunction, X, U and R modal operators are used), construct an atomic μ-calculus formula $M(\phi)$ by induction as follows:

$$M(\top) = \top, \qquad M(\bot) = \bot$$
$$M(p) = p, \qquad M(\neg p) = \neg p$$
$$M(\mathsf{X}\phi) = \Box M(\phi)$$
$$M(\phi \vee \psi) = M(\phi) \vee M(\psi)$$
$$M(\phi \wedge \psi) = M(\phi) \wedge M(\psi)$$
$$M(\phi\mathsf{U}\psi) = \mu Y.(M(\psi) \vee (M(\phi) \wedge \Box Y))$$
$$M(\phi\mathsf{R}\psi) = \nu Y.(M(\psi) \wedge (M(\phi) \vee \Box Y))$$

This translation does not have all properties that may be desired (see e.g. [10]) but it is sufficient for our purposes:

(i) In every word model, if a state satisfies ϕ then it satisfies $M(\phi)$,
(ii) In every Kripke model $\mathcal{K}$, if a state x satisfies $M(\phi)$ then every infinite path in $\mathcal{K}$ that starts from x, considered as a word model, satisfies ϕ.

Note that the converse to the implication in (ii) does not hold in general (consider e.g. $\phi = \mathsf{X}p \vee \mathsf{X}q$ and its translation $M(\phi) = \Box p \vee \Box q$). Both properties (i) and (ii) are proved by induction, for example for (ii):

- Assume $x \models M(\phi \vee \psi)$. Without loss of generality, assume $x \models M(\phi)$. By the inductive assumption, every path starting from x satisfies ϕ, so it satisfies $\phi \vee \psi$.
- Assume $x \models M(\phi\mathsf{U}\psi)$. By definition of M, on every path π starting from x the formula $M(\phi)$ holds in every state y until at some point $M(\psi)$ holds. Now, for each state y the path π has a sub-path that starts at y, and by the inductive assumption ϕ holds for all these subpaths until at some point ψ holds. As a result, $\phi\mathsf{U}\psi$ holds for the path π.

Properties (i) and (ii) immediately imply that ϕ is satisfiable if and only if $M(\phi)$ is satisfiable in a state where some infinite path begins. Putting $\psi = \nu X.\Diamond X$, a formula which holds exactly in those states where an infinite path begins, we get that ϕ is satisfiable if and only if $M(\phi) \wedge \psi$ is satisfiable. By Theorem 9.1, satisfiability of atomic μ-calculus formulas is undecidable. $\qquad \Box$

Finally, as a corollary of Theorem 9.1 we obtain:

Theorem 9.3. *The model checking problem for atomic CTL* is undecidable.*

Proof. We reduce the satisfiability problem for atomic LTL, with some insight into the proof of Theorem 9.1. Given any Turing machine $\mathcal{M}$ as considered there, consider a Kripke model with the set of states:

$$(\mathbb{A} \times \Gamma \times (Q \cup \{\mathtt{nohead}\})) \cup \{\$\}$$

with basic predicates defined in the obvious way, and with transitions going both ways between every two states. This model is orbit-finite and equivariant.

Now, for the atomic LTL formula ϕ obtained from $\mathcal{M}$ as in the proof of Theorem 9.1, the formula $\exists\phi$ holds in this model in the state \$ if and only if ϕ is satisfiable.

This works regardless of whether we interpret CTL* path formulas over arbitrary paths or over finitely supported ones, because the formula ϕ in the proof of Theorem 9.1 forces its model to be finitely supported. ☐

As a corollary we conclude that, unlike in the classical, atom-less setting, atomic CTL* is *not* a fragment of the atomic μ-calculus.

10. Separation results

In previous sections we introduced four atomic μ-calculi altogether: the scalar $\mathcal{L}_\mu^=$ and the vectorial $\vec{\mathcal{L}}_\mu^=$ calculus for equality atoms, and their corresponding versions $\mathcal{L}_\mu^<$ and $\vec{\mathcal{L}}_\mu^<$ for ordered atoms. It is clear that the vectorial versions are at least as expressive as the scalar ones, and that ordered-atoms calculi can express all properties that equality-atoms calculi can. In this section we show that all four calculi are in fact different, and we show three properties of Kripke models, called INFSUCC, CHAIN and EVENSUCC, that separate the four calculi as shown:
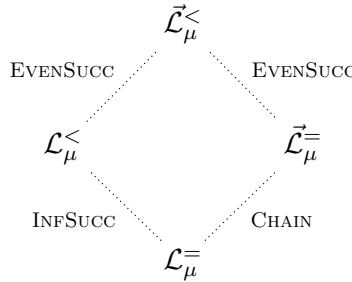
Later, in Section 11, we shall show a further property that is not definable even in $\vec{\mathcal{L}}_\mu^<$.

10.1. **The InfSucc property.**
It is obvious that atomic μ-calculus over ordered atoms is able to express many properties that are not definable over equality atoms simply because they refer to the ordering of atoms. However, there are properties that do not rely on that order *per se*, but are still definable in $\mathcal{L}_\mu^<$ and not even in the vectorial calculus $\vec{\mathcal{L}}_\mu^=$.

Consider a vocabulary of basic predicates that coincides with the set $\mathbb{A}$ of atoms. Let INFSUCC denote the property "there are infinitely many atoms a such that some immediate successor of the current state satisfies a".

Atom ordering is not mentioned anywhere in the definition on INFSUCC, and indeed it is an equivariant property of (states in) Kripke models over equality atoms. However, it is not definable in $\vec{\mathcal{L}}_\mu^=$. To see this, we rely on Theorem 8.4.

For any number k, we construct a model $\mathcal{K}$ over equality atoms and two states $p, q \in K$ that are k-bisimilar in k, but such that INFSUCC holds for p and fails for q. To this end, fix some finite set $S \subseteq \mathbb{A}$ with $|S| = 2k$. The model will be supported by S (see below how to achieve a similar result with an equivariant model). Let the set of states of $\mathcal{K}$ be:

$$K = \{p, q\} + \{r_a \mid a \in \mathbb{A}\}$$

with the obvious action of $\mathrm{Aut}(\mathbb{A})$ that makes each of p and q a singleton orbit. The satisfaction relation of $\mathcal{K}$ is such that

$$\mathrm{pred}(p) = \mathrm{pred}(q) = \emptyset \qquad \text{and} \qquad \mathrm{pred}(r_a) = \{a\} \quad \text{for } a \in \mathbb{A},$$

and the transition relation is defined by

$$p \longrightarrow r_a \text{ for } a \notin S \qquad \text{and} \qquad q \longrightarrow r_a \text{ for } a \in S.$$

It is clear that the property INFSUCC holds for p but fails for q. However, $\langle p, \epsilon \rangle$ and $\langle q, \epsilon \rangle$ are k-bisimilar according to Definition 8.3. To see this we describe a winning strategy for Duplicator in the corresponding k-bisimulation game.

Starting from the initial position $\langle p, \epsilon, q, \epsilon \rangle$, Spoiler may begin by making a series of atom replacement moves. Duplicator responds to each of these, ensuring that every position reached:

$$\langle p, \vec{a}, q, \vec{b} \rangle$$

is legal, i.e. that $\vec{a} \sim \vec{b}$ (hence in particular $|\vec{a}| = |\vec{b}|$), and that the condition

$$a_i \notin S \iff b_i \in S$$

holds for each $i = 1, \ldots, |\vec{a}|$. It is easy to see that this is possible thanks to the assumption that $|\vec{a}| \leq k$ and $|S| = 2k$.

Afterwords, when Spoiler decides to make a model move from (say) p to some r_a such that $a = a_i$ for some $i \leq |\vec{a}|$, Duplicator responds by moving from q to $b_i \in \vec{b}$. Thanks to Duplicator's previous effort, this is always a legal move. Other model moves from Spoiler are dealt with similarly. Afterwords, Spoiler is reduced to atom replacement moves that Duplicator can parry easily.

Our model $\mathcal{K}$ is supported by S, but it is easy to achieve a similar results in an equivariant model that includes a copy of our $\mathcal{K}$ for each $S \subseteq \mathbb{A}$ with $|S| = 2n$, with two additional equivariant initial states that can make transitions to the states p and q, respectively, in each of the copies.

As a result, by Theorem 8.4, INFSUCC is not definable in $\vec{\mathcal{L}}_\mu^=$.

Over ordered atoms, however, the situation is quite different. Notice that in a Kripke model with atoms, the set of successors of any particular state (and therefore the set of basic predicates that hold in those successors) is finitely supported. Recall that over ordered atoms, finitely supported subsets of $\mathbb{A}$ are exactly those that are finite unions of open intervals and single points. As a result, a finitely supported set of atoms is infinite if and only if it contains an open interval. Using this observation, it is easy to see that INFSUCC is definable in $\mathcal{L}_\mu^<$ by a formula:

$$\bigvee_{a < b \in \mathbb{A}} \bigwedge_{c \in (a;b)} \Diamond c.$$
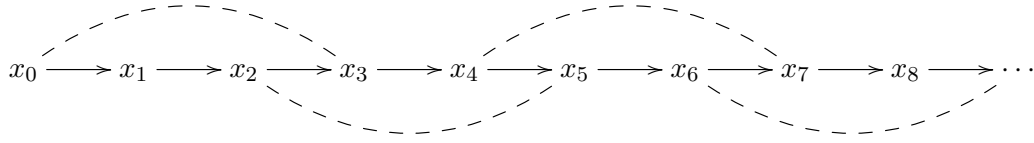
10.2. **The Chain property.**

As before, consider a vocabulary of basic propositions that coincides with the set of $\mathbb{A}$ of atoms (in this section we consider equality atoms only). Given a Kripke model $\mathcal{K}$, an infinite path:

$$x_0 \longrightarrow x_1 \longrightarrow x_2 \longrightarrow \cdots$$

in $\mathcal{K}$ is called a *chain* if

$$|\text{pred}(x_i)| = 1 \qquad \text{and} \qquad \text{pred}(x_{2i}) = \text{pred}(x_{2i+3})$$

for each $i \in \mathbb{N}$. The following diagram, where required equality of basic propositions satisfied in states is marked with dashed lines, justifies the name *chain*:

$$x_0 \longrightarrow x_1 \longrightarrow x_2 \longrightarrow x_3 \longrightarrow x_4 \longrightarrow x_5 \longrightarrow x_6 \longrightarrow x_7 \longrightarrow x_8 \longrightarrow \cdots$$

Let CHAIN denote the property "there exists a chain path starting from the current state". To show that it is not definable in $\mathcal{L}_\mu^=$, we rely on Theorem 8.2. We construct, for every number k, an orbit-finite Kripke model $\mathcal{K}$ over $\mathbb{A}$ as the language of basic predicate symbols, and two states $p, q \in K$ such that $\langle p, \epsilon \rangle$ and $\langle q, \epsilon \rangle$ are k-stack-bisimilar and CHAIN holds for p but fails for q.

Given a positive number k, pick some large n ($n > 2^k$ will be enough) and fix $2n$ distinct atoms

$$S = \{a_0, a_1, \ldots, a_{n-1}, a_n\} \cup \{b_1, \ldots, b_{n-1}\}.$$

We shall build a (finite) model $\mathcal{K}$ supported by S. Its set K of states will contain:

- p_i and q_i for $i = 1, \ldots, 2n$,
- r_i and s_i for $i = 1, \ldots, n - 1$,
- two special states $\top$ and $\bot$.

The satisfaction relation of $\mathcal{K}$ is defined so that:

- $\mathrm{pred}(p_i) = \mathrm{pred}(q_i) = \{a_{\frac{i}{2}-1}\}$ if i is even,
- $\mathrm{pred}(p_i) = \mathrm{pred}(q_i) = \{a_{\frac{i+1}{2}}\}$ if i is odd,
- $\mathrm{pred}(r_i) = \mathrm{pred}(s_i) = \{b_i\}$,
- $\mathrm{pred}(\top) = a_n$ and $\mathrm{pred}(\bot) = \emptyset$.

The transition relation of $\mathcal{K}$ contains the following:

- $p_i \longrightarrow p_{i+1}$ and $q_i \longrightarrow q_{i+1}$ for $1 \leq i < 2n$,
- $p_{2n} \longrightarrow \top$ and $q_{2n} \longrightarrow \bot$,
- $p_{2i+1} \longrightarrow r_i$ and $q_{2i+1} \longrightarrow s_i$ for $1 \leq i < n$,
- $r_i \longrightarrow q_{2i+3}$ and $s_i \longrightarrow p_{2i+3}$ for $1 \leq i < n - 1$,
- $r_{n-1} \longrightarrow \bot$ and $s_{n-1} \longrightarrow \top$,
- $\top \longrightarrow \top$.

These definitions look rather depressing, but the following diagram for $n = 5$ should make the idea clear. Here, atomic predicates satisfied in a state are marked over its outgoing transitions:
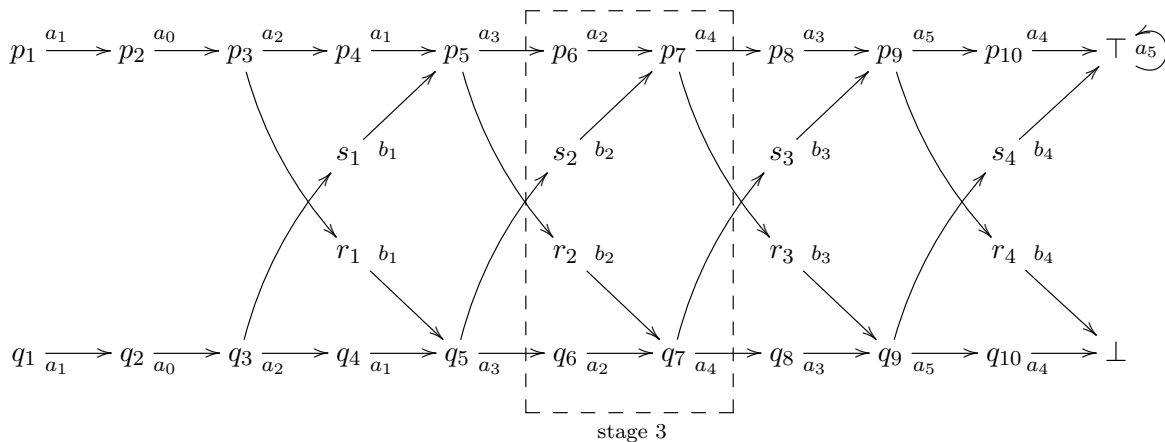
stage 3

(see below for an explanation of what *stage* 3 means).

It is important to note that the top-most path starting in p_1 and eventually entering an infinite loop at state $\top$, is a chain path. The state p_1 therefore satisfies the property CHAIN. On the other hand, the state q_1 does not satisfy the property: to build an infinite path from q_1 one has to visit a state s_i at some point, and when that happens the chain property is immediately invalidated, since the basic predicate b_i satisfied in s_i does not match the predicate a_i that held three steps earlier.

We shall show that p_1 and q_1 are k-stack-bisimilar. We begin by showing a strategy for Duplicator in the k-stack-bisimulation game that, although not successful, will be the basis for constructing a better, winning strategy.

Starting from the initial position $\langle p_1, \epsilon, q_1, \epsilon \rangle$, Duplicator may try to copy Spoiler's moves verbatim, that is:

- respond to a model move $p_i \longrightarrow p_{i+1}$ with $q_i \longrightarrow q_{i+1}$, to a model move $p_{2i+1} \longrightarrow r_i$ with $q_{2i+1} \longrightarrow s_i$, etc.,
- respond to a push move with an atom c with a push move with the same atom c.

(Spoiler's pop moves do not leave any choice for Duplicator, so there is no need to specify Duplicator's response to them.)

This strategy is losing: Spoiler can make a sequence of model moves from p_1 to $p_2, \ldots, p_{2n}$ and finally to $\top$ and then loop there, to which Duplicator responds by moving from q_1 all the way to $\bot$ and then she is stuck.

However, along the way, Duplicator may look for an opportunity to "cheat". To describe such opportunities, let us divide states in our model into *stages*, where stage i (for $i = 1, \ldots, n-1$) consists of states:

$$p_{2i}, q_{2i}, p_{2i+1}, q_{2i+1}, r_{i-1} \text{ and } s_{i-1}$$

(the latter two are included only if $i > 1$). For example, stage 3 is marked in the diagram above. Moreover, we say that an atom is *remembered* in a position if it is present in a sequence of atoms $\vec{a}$ that is a component of that position. Note that, according to Duplicator's strategy so far, the two sequences of atoms in every reachable position are the same, so at most k atoms are remembered in any given position.

Since model moves in the game must be matched by a model moves, the two states in a game position always belong to the same stage; we can therefore meaningfully say that a position belongs to a stage. An opportunity for Duplicator to cheat occurs when the play reaches a position in a stage i, such that neither the atom a_i nor b_i is remembered.

If such a position is reached, Duplicator starts to respond to Spoiler's moves in a way that arises from applying the swap permutation $(a_i \ b_i)$, leaving all other atoms intact. More specifically,

- if Spoiler makes a push move with a_i (or b_i), Duplicator responds with a push move with b_i (resp. a_i). This results in a legal position, since neither a_i nor b_i is satisfied in any state that belongs to stage i;
- if Spoiler makes a model move $p_{2i+1} \longrightarrow p_{2i+2}$, Duplicator responds by $q_{2i+1} \longrightarrow s_i$. The resulting position

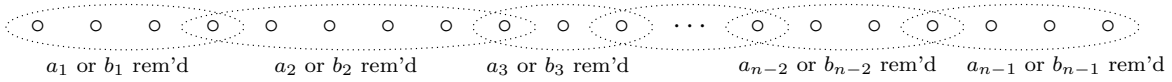$$\langle p_{2i+2}, \vec{a}, s_i, \vec{a} \cdot (a_i \ b_i) \rangle$$

is legal. Moreover, after the next round of model moves (possibly preceded by a sequence of push and pop moves) the game will necessarily reach the position

$$\langle p_{2i+3}, \vec{c}, p_{2i+3}, \vec{c} \cdot (a_i \ b_i) \rangle$$

for some sequence $\vec{c}$. This position is winning for Duplicator, since neither a_i nor b_i are relevant for the remainder of the game.

Other model moves for Spoiler are dealt with in a similar way.

It remains to be shown that an opportunity to cheat must inevitably happen for Duplicator. Assume to the contrary that during a play, either a_i or b_i is remembered throughout the stage i, for each $i = 1, \ldots, n-1$. Since model moves do not change the set of atoms remembered, it follows that a_i or b_i is remembered also in the last position before entering stage i, and in the first position after leaving stage i. As a result, positions in the play are covered by overlapping regions throughout which particular atoms must be remembered, as depicted here:

$$a_1 \text{ or } b_1 \text{ rem'd} \qquad a_2 \text{ or } b_2 \text{ rem'd} \qquad a_3 \text{ or } b_3 \text{ rem'd} \qquad a_{n-2} \text{ or } b_{n-2} \text{ rem'd} \quad a_{n-1} \text{ or } b_{n-1} \text{ rem'd}$$

(The regions correspond to stages of the game, but note that the regions need not be of the same size, since the players may make different numbers of push-or-pop moves in each stage.) Since only k atoms may be remembered at any given time, and since atoms are remembered in a stack-like regime, this is impossible if $n > 2^k$. To prove this, proceed by induction on k:

- for $k = 1$ we have $n - 1 \geq 2$ so at some point two atoms must be remembered but there is no space for it,
- for $k > 1$, notice that throughout the overlapping regions, the stack that remembers atoms may never get empty, even for a moment. This means that there is some atom a_i (or b_i) that sits at the bottom of the stack (formally, remains the first element of the atom sequence that is a component of game positions) throughout this part of the play. One of the intervals

$$1, \ldots, i-1 \qquad \text{or} \qquad i+1, \ldots, n-1$$

  is of the size at least half of the interval $1, \ldots, n-1$. In the part of the play defined by the longer of the two intervals of stages, the atom a_i (or b_i) remains useless, which effectively leaves enough space for remembering $k-1$ atoms. It is now enough to invoke the inductive assumption.

A winning strategy for Duplicator is to first copy Spoiler's moves, then use the first opportunity to cheat to force the game into a winning position as described above. This shows that Duplicator wins from the starting position $\langle p_1, \epsilon, q_1, \epsilon \rangle$ as required.

This means that $\langle p_1, \epsilon \rangle$ and $\langle q_1, \epsilon \rangle$ are k-stack-bisimilar, hence by Theorem 8.2 they satisfy the same globally k-supported formulas. Since every formula of the atomic μ-calculus is globally k-supported for some k, this implies that no formula in $\mathcal{L}_\mu^=$ defines CHAIN.

However, it is not hard to define CHAIN in the vectorial calculus $\vec{\mathcal{L}}_\mu^=$. Indeed, for any atom $a \in \mathbb{A}$, the property *there exists a chain*

$$x_0 \longrightarrow x_1 \longrightarrow x_2 \longrightarrow x_3 \longrightarrow \cdots$$

starting in the current state such that $pred(x_1) = \{a\}$, is defined by a formula:

$$\phi_a = \nu X_a. \left\{ X_b. \bigvee_{c \in \mathbb{A}} (\theta_c \wedge \Diamond (\theta_b \wedge \Diamond X_c)) \right\}_{b \in \mathbb{A}},$$

where for any $b \in \mathbb{A}$ the formula

$$\theta_b = b \wedge \bigwedge_{d \neq b} \neg d$$

ensures that b, and no other basic proposition, holds in the current state. Then CHAIN is defined by the alternative $\bigvee_{a \in \mathbb{A}} \phi_a$.

10.3. The EvenSucc property.

As before, consider a vocabulary of basic propositions that coincides with the set of $\mathbb{A}$ of atoms. Let EVENSUCC denote the property "the number of atoms a such that some immediate successor of the current state satisfies a, is finite and even". We shall show that this property is not definable in $\vec{\mathcal{L}}_\mu^=$ or $\mathcal{L}_\mu^<$, but it is definable in $\vec{\mathcal{L}}_\mu^<$.

The argument for $\vec{\mathcal{L}}_\mu^=$ relies on the same model $\mathcal{K}$ as considered in Section 10.1. Notice that $|S|$ there is even, hence the state $q \in K$ satisfies EVENSUCC, but the state $p \in K$ does not, and they are k-bisimilar over equality atoms as shown before.

For $\mathcal{L}_\mu^<$, we rely on Theorem 8.2. For any number k, we construct a model $\mathcal{K}$ over ordered atoms and two states $p, q \in K$ that are k-stack-bisimilar in k, but such that EVENSUCC holds for p and fails for q. To this end, put $n = 2^{k+1}$ and fix some finite sets $S, T \subseteq \mathbb{A}$ with $|S| = n$, $|T| = 2n + 1$ and such that every element of T is greater than every element of S. Let the set of states of $\mathcal{K}$ be:

$$K = \{p, q\} + \{r_a \mid a \in \mathbb{A}\}$$

with the obvious action of $\mathrm{Aut}(\mathbb{A})$ that makes each of p and q a singleton orbit. The satisfaction relation of $\mathcal{K}$ is such that

$$\mathrm{pred}(p) = \mathrm{pred}(q) = \emptyset \qquad \text{and} \qquad \mathrm{pred}(r_a) = \{a\} \quad \text{for } a \in \mathbb{A},$$

and the transition relation is defined by

$$p \longrightarrow r_a \text{ for } a \in S \qquad \text{and} \qquad q \longrightarrow r_a \text{ for } a \in T.$$

Obviously the property EVENSUCC holds for p but fails for q. However, $\langle p, \epsilon \rangle$ and $\langle q, \epsilon \rangle$ are k-stack-bisimilar according to Definition 8.3. To see this we describe a winning strategy for Duplicator in the corresponding k-stack-bisimulation game.

Starting from the initial position $\langle p, \epsilon, q, \epsilon \rangle$, Spoiler may begin by making a series of push moves. An important observation is that at this stage, before any model move has been made, pop moves make no sense for Spoiler: they just bring the game to a position that has already been visited. This means that this first phase of the game takes at most k moves. In this phase, Duplicator responds to Spoiler's moves much as she would in an k-round Ehrenfeucht-Fraïssé game on two large total orders, one of them of even size, the other of odd size. As in that classical scenario, Duplicator is able to survive k rounds of the game, and ensure that any model move by the Spoiler afterwards can be met with a legal response.

As a result, by Theorem 8.2, EVENSUCC is not definable in $\mathcal{L}_\mu^<$.

On the other hand, the property is definable in $\vec{\mathcal{L}}_\mu^<$. To see this, for fixed atoms $a < b$, consider the property "b is the least atom greater than a such that some successor of the current state satisfies b", defined by:

$$\phi_{a,b}^1 = \Diamond b \wedge \bigwedge_{c \in (a;b)} \neg \Diamond c.$$

Then define the property "b is the *second* least atom greater than a such that some successor of the current state satisfies b" by:

$$\phi_{a,b}^2 = \bigvee_{c \in (a;b)} (\phi_{a,c}^1 \wedge \phi_{c,b}^1).$$

Then consider the property "the number of atoms $c > a$ such that some immediate successor of the current state satisfies c, is finite and even", defined by:

$$\psi_a = \mu X_a. \left\{ X_b. \left(\bigwedge_{c>b} \neg \Diamond c \right) \vee \left(\bigvee_{c>b} (\phi_{b,c}^2 \wedge X_c) \right) \right\}_{b \geq a}.$$

Finally, EVENSUCC is defined by:

$$\bigvee_{a \in \mathbb{A}} \bigwedge_{b < a} \psi_b.$$

11. EXPRESIVENESS LIMITATIONS

As before, consider Kripke models over a vocabulary of basic predicates that coincides with the set $\mathbb{A}$ of atoms. Denote the property "there exists an infinite path where no a holds more than once", by #PATH. Such properties of states in Kripke models have potentially significant practical importance. For example, one may imagine a system equipped with a token (e.g. password) generator where one needs to verify that, on every path where no token is generated more than once, the security of the system is never breached.

In a previous version of this paper [28], we proved that #PATH is not definable in the $\mathcal{L}_\mu^=$, the scalar μ-calculus over equality atoms. This is in contrast to the similar but definable property **P2** from Section 6. Our proof there did not work for ordered atoms, and in [28] we left is as an open problem whether the property is definable in the calculus that we here call $\mathcal{L}_\mu^<$. We shall now show that, in fact, #PATH is not definable even in the vectorial calculus $\vec{\mathcal{L}}_\mu^<$ over ordered atoms.

But first notice that there would be no hope of defining #PATH in some calculus with decidable model checking if the property itself was undecidable. So we begin by showing that the property is indeed decidable on Kripke models over equality atoms, which makes its undefinability in our μ-calculi all the more disappointing.

Theorem 11.1. *#PATH is decidable on orbit-finite Kripke models over equality atoms.*

Proof. For simplicity, assume that a given orbit-finite Kripke model $\mathcal{K}$ is equivariant; a generalization to finitely supported models is straightforward.

Notice that for every state $x \in K$, the set $\text{pred}(x)$ is either finite (and contained in $\text{supp}(x)$) or co-finite. Moreover, a single orbit of K only contains states of one of these two kinds. It is not difficult to decide the existence of a desired path where at least one state is of the second kind. Indeed, two such states cannot occur on the path at all, and even if exactly one of them occurs, almost all other states on the path must satisfy none of the basic predicates. The existence of such a path from a given state x is straightforward to decide.

Once the existence of such paths is excluded, all (orbits of) states x with co-finite $\text{pred}(x)$ may be safely deleted from the model. From now on, assume that $\text{pred}(x) \subseteq \text{supp}(x)$ for each $x \in K$.

Derived from $\mathcal{K}$, construct a new orbit-finite Kripke model $\hat{\mathcal{K}}$, over the empty set of basic predicates, with the set of states defined by:

$$\hat{K} = \{\langle x, S \rangle \mid x \in K, \ S \subseteq \text{supp}(x) \setminus \text{pred}(x)\}$$

and the transition relation by:

$$\langle x, S \rangle \longrightarrow^{\hat{\mathcal{K}}} \langle y, T \rangle \text{ if and only if } x \longrightarrow^{\mathcal{K}} y \text{ and } (S \cup \text{pred}(x)) \cap \text{supp}(y) \subseteq T$$

(since there are no basic predicates, the satisfaction relation in $\hat{\mathcal{K}}$ is trivial). The intuition is that in a state $\langle x, S \rangle$, atoms in S are marked as having had occurred previously on a path, and are forbidden from occurring in the future. Note that this marking is restricted to atoms from the support of the current state x only.

Note that $\hat{\mathcal{K}}$ is indeed equivariant and orbit-finite: every orbit of K gives rise to at most 2^k orbits in $\hat{K}$, where k is the size of the least support of any (equivalently, every) element in the orbit. Moreover, (a representation of) $\hat{\mathcal{K}}$ is computable from $\mathcal{K}$: for each orbit in K one can enumerate all corresponding orbits in $\hat{K}$, and orbits of transitions are also easy to enumerate.

We shall now prove that a state $x \in K$ admits an infinite path where no a holds more than once, if and only if $\langle x, \emptyset \rangle \in \hat{K}$ admits any infinite path. (Note that the theorem follows from this, since the latter property is decidable, indeed, it is definable in the atomic μ-calculus $\mathcal{L}_\mu^=$, whose model-checking problem is decidable by Theorem 5.1).

For the left-to-right implication, assume an infinite path in $\mathcal{K}$, i.e., a sequence

$$x = x_0 \longrightarrow^{\mathcal{K}} x_1 \longrightarrow^{\mathcal{K}} x_2 \longrightarrow^{\mathcal{K}} x_3 \longrightarrow^{\mathcal{K}} \cdots$$

such that $\text{pred}(x_i) \cap \text{pred}(x_j) = \emptyset$ for each $i \neq j \in \mathbb{N}$. Define

$$y_i = \langle x_i, S_i \rangle, \quad \text{where} \quad S_i = \text{supp}(x_i) \cap \bigcup_{j=0}^{i-1} \text{pred}(x_j).$$

In particular, $y_0 = \langle x, \emptyset \rangle$. Then

$$\langle x_i, S_i \rangle \longrightarrow^{\hat{\mathcal{K}}} \langle x_{i+1}, S_{i+1} \rangle \tag{11.1}$$

for each $i \in \mathbb{N}$. Indeed, calculate

$$(S_i \cup \text{pred}(x_i)) \cap \text{supp}(x_{i+1}) = \left(\left(\text{supp}(x_i) \cap \bigcup_{j=0}^{i-1} \text{pred}(x_j) \right) \cup \text{pred}(x_i) \right) \cap \text{supp}(x_{i+1})$$

$$\subseteq \left(\bigcup_{j=0}^{i} \text{pred}(x_j) \right) \cap \text{supp}(x_{i+1}) = S_{i+1}.$$

As a result

$$\langle x, \emptyset \rangle = \langle x_0, S_0 \rangle \longrightarrow^{\hat{\mathcal{K}}} \langle x_1, S_1 \rangle \longrightarrow^{\hat{\mathcal{K}}} \langle x_2, S_2 \rangle \longrightarrow^{\hat{\mathcal{K}}} \cdots \tag{11.2}$$

form an infinite path in $\hat{\mathcal{K}}$.

For the right-to-left implication, assume any infinite sequence as in (11.2), for some x_i and S_i such that the condition (11.1) holds for every $i \in \mathbb{N}$. We construct sequences

$$y_0, y_1, \ldots \in K \quad T_0, T_1, \ldots \subseteq \mathbb{A} \quad \pi_1, \pi_2, \ldots \in \text{Aut}(\mathbb{A})$$

by simultaneous induction as follows:

- $y_0 = x_0$ and $T_0 = S_0$,
- π_{i+1} is an atom automorphism such that:
  - $\pi_{i+1}(a) = a$ for $a \in \mathrm{supp}(y_i)$, and
  - $\pi_{i+1}(a) \notin \bigcup_{j=0}^{i} \mathrm{supp}(y_j)$ for $a \in \mathrm{supp}(x_{i+1} \cdot \pi_1 \pi_2 \cdots \pi_i) \setminus \mathrm{supp}(y_i)$,

  and acting in an arbitrary way on the remaining atoms (such an automorphism always exists since the union of supports in the second clause is finite, and therefore some atoms exists outside of it),
- $y_{i+1} = x_{i+1} \cdot \pi_1 \pi_2 \cdots \pi_i \pi_{i+1}$,
- $T_{i+1} = S_{i+1} \cdot \pi_1 \pi_2 \cdots \pi_i \pi_{i+1}$.

Notice that, since $\langle x_{i+1}, S_{i+1} \rangle$ is a legal state in $\hat{K}$, by equivariance so is $\langle y_{i+1}, T_{i+1} \rangle$. Moreover,

$$\langle y_i, T_i \rangle \longrightarrow^{\hat{\mathcal{K}}} \langle y_{i+1}, T_{i+1} \rangle$$

for each $i \in \mathbb{N}$, therefore the sequence

$$\langle y_0, T_0 \rangle \longrightarrow^{\hat{\mathcal{K}}} \langle y_1, T_1 \rangle \longrightarrow^{\hat{\mathcal{K}}} \langle y_2, T_2 \rangle \longrightarrow^{\hat{\mathcal{K}}} \cdots$$

is an infinite path in $\hat{K}$. To see this, note that

$$\langle y_i \cdot \pi_{i+1}, T_i \cdot \pi_{i+1} \rangle \longrightarrow^{\hat{\mathcal{K}}} \langle y_{i+1}, T_{i+1} \rangle,$$

(by equivariance of the transition relation $\longrightarrow^{\hat{\mathcal{K}}}$, applying the automorphism $\pi_1 \cdots \pi_{i+1}$ to the transition $\langle x_i, S_i \rangle \longrightarrow^{\hat{\mathcal{K}}} \langle x_{i+1}, S_{i+1} \rangle$), and

$$y_i \cdot \pi_{i+1} = y_i \quad \text{and} \quad T_i \cdot \pi_{i+1} = T_i$$

since π_{i+1} by definition fixes $\mathrm{supp}(y_i)$ and $T_i \subseteq \mathrm{supp}(y_i)$.

As a consequence, the sequence

$$y_0 \longrightarrow^{\mathcal{K}} y_1 \longrightarrow^{\mathcal{K}} y_2 \longrightarrow^{\mathcal{K}} \cdots$$

forms a path in $\mathcal{K}$. A useful property of this path, easy to infer from the definition of y_i, is that:

$$(\mathrm{supp}(y_{i+1}) \setminus \mathrm{supp}(y_i)) \cap \bigcup_{j=0}^{i} \mathrm{supp}(y_j) = \emptyset. \tag{11.3}$$

In words, whenever a locally fresh atom appears in some y_i, then it does not appear anywhere earlier in the path.

We shall show that no basic predicate holds on this path more than once. Assume towards a contradiction that

$$a \in \mathrm{pred}(y_i) \cap \mathrm{pred}(y_j)$$

for some $a \in \mathbb{A}$ and $i < j$. Then obviously $a \in \mathrm{supp}(y_i)$ and $a \in \mathrm{supp}(y_j)$, and by induction on the difference $j - i$, using (11.3), we get that $a \in \mathrm{supp}(y_k)$ for all k between i and j. Again by induction, and by definition of the transition relation in $\hat{\mathcal{K}}$, a belongs to all sets $T_{i+1}, T_{i+2}, \ldots, T_j$. But this means that $a \in T_j \cap \mathrm{pred}(y_j)$, which contradicts the fact that $\langle y_j, T_j \rangle$ is a legal state in $\hat{K}$. This completes the proof of the right-to-left implication, and of the entire theorem. $\square$

We shall now show that, in spite of its decidability and intuitive simplicity, #PATH is not definable in $\vec{\mathcal{L}}_{\mu}^{<}$, the vectorial atomic μ-calculus over ordered atoms. To this end, we rely on Theorem 8.4. We construct, for every number k, an orbit-finite Kripke model $\mathcal{K}$ over

$\mathbb{A}$ as the language of basic predicate symbols, and two states $x, y \in K$ such that $\langle x, \epsilon \rangle$ and $\langle y, \epsilon \rangle$ are k-bisimilar and #PATH fails for x but holds for y.

Given $k > 0$, fix some $n > k$ and fix any set S of $2n$ atoms:

$$a_1 < b_1 < a_2 < b_2 < \cdots < a_n < b_n.$$

We shall build a Kripke model $\mathcal{K}$ and its sub-model $\check{\mathcal{K}}$, both supported by S. Both models operate in three phases:

- Phase I, both in $\mathcal{K}$ and in $\check{\mathcal{K}}$, consists of n states $p_1, \ldots, p_n \in K$ such that $\mathrm{pred}(p_i) = \{a_i\}$, with transitions:

$$p_i \longrightarrow p_{i+1} \qquad \text{for } 1 \le i < n.$$

  Note that the transition relation is entirely deterministic in this phase.

- Phase II in $\mathcal{K}$ consists of $n^2 + n$ states

$$q_1^0, q_2^0, \ldots, q_n^0, q_1^1, q_2^1, \ldots, q_n^1, \ldots, q_1^n, q_2^n, \ldots, q_n^n \in K$$

  with

$$\mathrm{pred}(p_j^i) = \begin{cases} \{a_j\} & \text{if } i = j \\ \{b_j\} & \text{otherwise.} \end{cases}$$

  A transition from Phase I to Phase II is realised by adding $n + 1$ transitions from p_n:

$$p_n \longrightarrow q_1^i \qquad \text{for } 0 \le i \le n.$$

  Note that this introduces nondeterminism to the model. Further transitions in Phase II are:

$$q_j^i \longrightarrow q_{j+1}^i \qquad \text{for } 0 \le i \le n \text{ and } 1 \le j < n.$$

  These transitions, again, are deterministic.

  In $\check{\mathcal{K}}$, Phase II is the same but with the states $q_1^0, q_2^0, \ldots, q_n^0$ excluded (so only n^2 states are present).

- Phase III, both in $\mathcal{K}$ and in $\check{\mathcal{K}}$, is an infinite (but single-orbit) clique of states:

$$\{r_c \mid c \in \mathbb{A}\} \subseteq K$$

  with $\mathrm{pred}(r_c) = \{c\}$ and the full relation as the transition relation. Additionally, a transition from Phase II to Phase III is realised by adding a transition from each state p_n^i (for $0 \le i \le n$) to each state r_c. This, obviously, is highly nondeterministic.

  Note that every state in K satisfies exactly one basic predicate. The following diagram shows the general shape of $\mathcal{K}$, with basic predicates satisfied in a state marked over its

outgoing transitions:
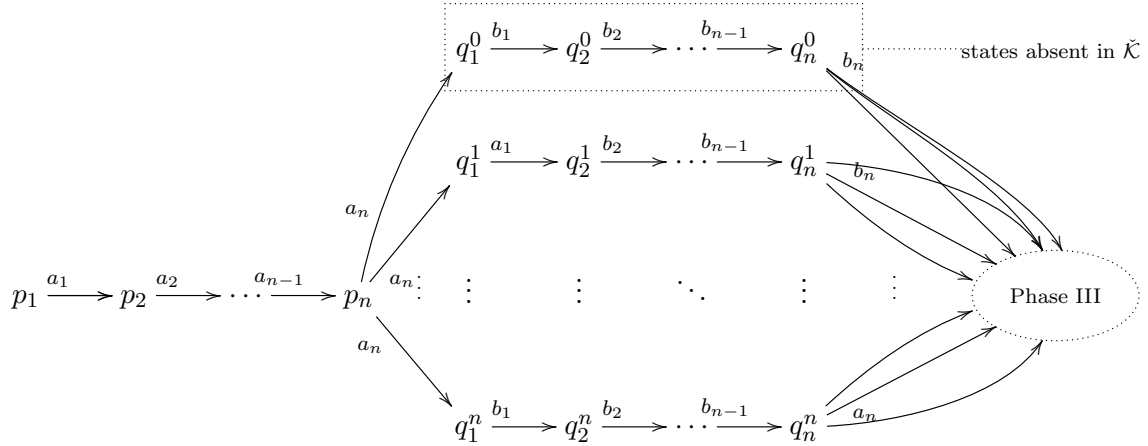
Both $\mathcal{K}$ and $\check{\mathcal{K}}$ are orbit-finite. Indeed, there are only finitely many states in Phases I and II, each of them forming a singleton orbit, and the single infinite orbit of states is formed by Phase III. The models are not equivariant, but with a small additional effort they can be made so: keeping n fixed, consider a disjoint union of the models $\mathcal{K}$ taken for all sets of atoms $S = \{a_1, b_1, \cdots, a_n, b_n\}$. Then add a single equivariant state $\star$ where no basic predicate holds, with transitions from $\star$ to states p_1 in all the disjoint components. The resulting models remain orbit-finite, they are equivariant, and the following reasoning applies to them with little change.

It is easy to see that the state p_1 (and, indeed, every state) in $\mathcal{K}$ satisfies #PATH: in the finite path

$$p_1, \ldots, p_n, q_1^0, \ldots, q_n^0$$

no basic predicate holds more than once, and in Phase III it is easy to extend it to an infinite path so that this remains true. On the other hand, in the smaller model $\check{\mathcal{K}}$ the state p_1 does not satisfy #PATH, since in Phase II one of the a_i from Phase I is necessarily repeated there.

We shall show that $\langle p_1, \epsilon \rangle$ in $\mathcal{K}$ is k-bisimilar to $\langle p_1, \epsilon \rangle$ in $\check{\mathcal{K}}$ by providing a winning strategy for Duplicator in the corresponding k-bisimulation game. Formally, since we defined both k-bisimulations and k-bisimulation games as operating on single Kripke models, we need to consider the bisimulation game on a disjoint union of $\mathcal{K}$ and $\check{\mathcal{K}}$. To avoid confusion we will denote the states in the $\check{\mathcal{K}}$-component of this union by $\check{p}_i$, $\check{q}_j^i$ etc., and will describe a winning strategy for Duplicator from the position

$$\langle p_1, \epsilon, \check{p}_1, \epsilon \rangle$$

in the k-bisimulation game.

In Phase I, Duplicator copies Spoiler's moves verbatim. That is, whenever Spoiler makes a model move from p_i to p_{i+1} (or from $\check{p}_i$ to $\check{p}_{i+1}$), Duplicator responds by moving from $\check{p}_i$ to $\check{p}_{i+1}$ (respectively, from p_i to p_{i+1}; Duplicator has no choice here anyway, since in Phase I the transition relation is deterministic in both models). Moreover, whenever Spoiler makes an atom replacement move by choosing some new tuple of atoms, Duplicator responds by picking the same tuple of atoms. In this way, Duplicator ensures that in Phase I the game visits only positions of the form $\langle p_i, \vec{a}, \check{p}_i, \vec{a} \rangle$ for $\vec{a} \in \mathbb{A}^{(\leq k)}$.

Duplicator fares well this way until at some point Spoiler decides to make a model move from a configuration

$$\langle p_n, \vec{a}, \check{p}_n, \vec{a} \rangle.$$

Note that at this point Duplicator has no control over how atoms in $\vec{a}$ relate to the atoms $a_1, b_1, \cdots, a_n, b_n$.

If Spoiler makes a model move from $\check{p}_n$ to some $\check{q}_1^i$, Duplicator responds by moving from p_n to q_1^i. The game is then essentially over: $\check{q}_1^i$ and q_1^i have the same futures in $\check{\mathcal{K}}$ and $\mathcal{K}$, so Duplicator can proceed by forever copying Spoiler's moves. Similarly, if Spoiler makes a move from p_n to some q_1^i for $i > 0$, Duplicator responds by moving from $\check{p}_n$ to $\check{q}_1^i$ and wins.

The only remaining case is when Spoiler moves from p_n to q_1^0; the only move that does not have an obvious counterpart in $\check{\mathcal{K}}$. Since $n > k$ and $|\vec{a}| \leq k$, there exists an $i \in \{1, \ldots, n\}$ such that $\vec{a}$ contains no atom from the closed interval $[a_i, b_i]$. Then there exists an atom automorphism π (i.e., a monotone bijection on the total order of rational numbers) such that:

- $\vec{a} \cdot \pi = \vec{a}$, that is, $\pi(c) = c$ for each $c \in \vec{a}$,
- $\pi(b_j) = b_j$ for $j \neq i$, and
- $\pi(b_i) = a_i$.

Having fixed such a π, Duplicator responds to Spoiler's move by moving from $\check{p}_n$ to $\check{q}_1^i$. Note that the target position:

$$\langle q_1^0, \vec{a}, \check{q}_1^i, \vec{a} \rangle$$

is legal. For this we need to check that:

$$\langle \text{pred}(q_1^0), \vec{a} \rangle \sim \langle \text{pred}(\check{q}_1^i), \vec{a} \rangle.$$

By definition, $\text{pred}(q_1^0) = \{b_1\}$. If $i \neq 1$ then $\text{pred}(\check{q}_1^i) = \{b_1\}$ and the condition holds trivially. If $i = 1$ then $\text{pred}(\check{q}_1^1) = \{a_1\}$ and the condition holds since $\vec{a}$ does not contain any atom from the interval $[a_1, b_1]$.

Afterwards, Duplicator copies Spoiler's moves translating them along the automorphism π. More specifically, throughout Phase II:

- if Spoiler makes a model move from q_j^0 to q_{j+1}^0, Duplicator responds by moving from $\check{q}_j^i$ to $\check{q}_{j+1}^i$ (again, no real choice here) and *vice versa*,
- if Spoiler makes an atom replacement move by choosing a new tuple of atoms $\vec{c}$ in the $\mathcal{K}$-component of the game position, Duplicator responds by choosing $\vec{c} \cdot \pi$. If, on the other hand, Spoiler picks a new tuple of atoms $\vec{d}$ in the $\check{\mathcal{K}}$-component, Duplicator responds by choosing $\vec{d} \cdot \pi^{-1}$.

In Phase III the strategy remains similar, except that a Spoiler's model move to a state r_c is matched by Duplicator's move to $\check{r}_{\pi(c)}$, and a Spoiler's move to $\check{r}_c$ is matched by a move to $r_{\pi^{-1}(c)}$. This strategy ensures that only legal positions in the game are visited, and since every move by Spoiler is met with a response, the strategy is winning for Duplicator.

This means that $\langle p_0, \epsilon \rangle$ and $\langle \check{p}_0, \epsilon \rangle$ are k-bisimilar, hence by Theorem 8.4 they satisfy the same globally k-supported formulas. Since every formula of the atomic μ-calculus is globally k-supported for some k, this implies that no formula defines #PATH.

12. Future work

We list some interesting aspects of the atomic μ-calculus that we leave for future work.

Complexity issues. For the decidability results we presented, in particular for the model checking problem over orbit-finite structures, one immediately asks about the complexity of the algorithms proposed. The answer depends on the way one measures the size of input structures. One obvious option is to consider the length of their representation with set-builder expressions and first-order formulas. With this view, most basic operations listed in Remark 3.4 become Pspace-hard, because the first-order theory of pure equality is Pspace-complete. As a result, the complexity of basic operations dwarfs the distinction between the two algorithmic approaches to model checking based on direct fixpoint computation and on parity games.

Another approach is to measure orbit-finite structures by the number of their orbits, and the (hereditary) size of their least support. Note that the number of orbits of a set can be exponentially bigger than the size of its logical representation; for example, the number of orbits of the set $\mathbb{A}^n$, whose representation has size linear in n, is equal to the n-th Bell number. In this view the difference between the two approaches to model checking becomes more prominent.

We defer precise complexity analyses until we have a better general understanding of various time and space complexity models on atomic structures.

Other atoms. To simplify the presentation, in this paper we focus on equality atoms and ordered atoms only. However, the definition of atomic μ-calculus and its basic properties could be transported without much difficulty to other relational structures of atoms subject to some model-theoretic assumptions such as oligomorphicity (i.e. the assumption that $\mathbb{A}^n$ is orbit-finite for every n), homogeneity (i.e. the assumption that every isomorphism between two finite substructures of $\mathbb{A}$ extends to an automorphism of $\mathbb{A}$), and decidability of the first-order theory of $\mathbb{A}$.

This, however, is with some exceptions. Some of our proofs in this paper, in particular the one for undefinability of #Path, rely on particular atom structures and it is not clear how far they generalize. As a concrete open problem, we leave the question whether #Path is perhaps definable over atoms from the universal undirected graph, also known as the *random graph* or *Rado graph*, where vertices are natural numbers, and an undirected edge $\{n, m\}$ is present if and only if the n-th bit in the binary representation of m is 1 (for $n < m$).

Defining #Path. The fact that #Path is not definable in atomic μ-calculi is disappointing, since it looks like a property of potential practical importance in system verification. Since we know that #Path is decidable, it is desirable to extend atomic μ-calculus in some well-structured and syntactically economic way that would allow one to define such properties while preserving the decidability of model checking. The property of "global freshness" has been studied in the context of automata with atoms [40], and one may look for inspiration there. Our proof of Theorem 11.1 also suggests some promising options. We leave this for future work.

References

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

[2] A. Bauer and P. L. Lumsdaine. On the bourbaki–witt principle in toposes. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 155, pages 87–99. Cambridge University Press, 2013.

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

[4] M. Bojańczyk and T. Place. Toward model theory with data values. In *Procs. ICALP 2012 Part II*, volume 7392 of *Lecture Notes in Computer Science*, pages 116–127, 2012.

[5] J. Bradfield and C.Stirling. Modal mu-calculi. In *Handbook of Modal Logic*, volume 3, pages 721 – 756. Elsevier, 2007.

[6] J. Bradfield and C. Stirling. Modal logics and mu-calculi: an introduction. In *Handbook of Process Algebra*, pages 293–330. North-Holland, 2001.

[7] J. Bradfield and I. Walukiewicz. The mu-calculus and model-checking. In H. Veith E. Clarke, T. Henzinger, editor, *Handbook of Model Checking*. Springer-Verlag, 2015.

[8] C. Carapelle and M. Lohrey. Temporal logics with local constraints (invited talk). In *Procs. CSL 2015*, volume 41 of *LIPIcs*, pages 2–13, 2015.

[9] V. Ciancia and U. Montanari. Symmetries, local names and dynamic (de)-allocation of names. *Inf. Comput.*, 208(12):1349–1367, 2010.

[10] S. Cranen, J. F. Groote, and M. Reniers. A linear translation from CTL* to the first-order modal μ-calculus. *Theoretical Computer Science*, 412(28):3129 – 3139, 2011.

[11] M. Dam. Model checking mobile processes. *Information and Computation*, 129(1):35–51, 1996.

[12] R. De Nicola and M. Loreti. Multiple-labelled transition systems for nominal calculi and their logics. *Mathematical Structures in Computer Science*, 18(01):107–143, 2008.

[13] S. Demri and D. D'Souza. An automata-theoretic approach to constraint LTL. *Inf. Comput.*, 205(3):380–415, 2007.

[14] S. Demri, D. D'Souza, and R. Gascon. Temporal logics of repeating values. *J. Log. Comput.*, 22(5):1059–1096, 2012.

[15] S. Demri, D. Figueira, and M. Praveen. Reasoning about data repetitions with counter systems. *Logical Methods in Computer Science*, 12(3), 2016.

[16] S. Demri and R. Lazic. LTL with the freeze quantifier and register automata. *ACM Trans. Comput. Log.*, 10(3):16:1–16:30, 2009.

[17] S. Demri, R. Lazic, and D. Nowak. On the freeze quantifier in constraint LTL: decidability and complexity. *Inf. Comput.*, 205(1):2–24, 2007.

[18] E. A. Emerson and J. Y. Halpern. "sometimes" and "not never" revisited: On branching versus linear time temporal logic. *J. ACM*, 33(1):151–178, 1986.

[19] G.-L. Ferrari, S. Gnesi, U. Montanari, and M. Pistore. A model-checking verification environment for mobile processes. *ACM Transactions on Software Engineering and Methodology*, 12(4):440–473, 2003.

[20] D. Figueira. Alternating register automata on finite words and trees. *Logical Methods in Computer Science*, 8(1), 2012.

[21] D. Figueira and L. Segoufin. Future-looking logics on data words and trees. In *Procs. MFCS 2009*, volume 5734 of *Lecture Notes in Computer Science*, pages 331–343, 2009.

[22] N. Francez and M. Kaminski. Finite-memory automata. *TCS*, 134(2):329–363, 1994.

[23] M. Gabbay and A. M. Pitts. A new approach to abstract syntax with variable binding. *Formal Asp. Comput.*, 13(3-5):341–363, 2002.

[24] J. F. Groote and R. Mateescu. Verification of temporal properties of processes in a setting with data. In *Procs. AMAST'99*, pages 74–90, 1999.

[25] J. F. Groote and T. AC Willemse. Model-checking processes with data. *Science of Computer Programming*, 56(3):251–273, 2005.

[26] M. Jurdzinski and R. Lazic. Alternating automata on data trees and xpath satisfiability. *ACM Trans. Comput. Log.*, 12(3):19:1–19:21, 2011.

[27] B. Klin, E. Kopczyński, J. Ochremiak, and S. Toruńczyk. Locally finite constraint satisfaction problems. In *Procs. LICS 2015*, pages 475–486, 2015.

[28] B. Klin and M. Łełyk. Modal μ-Calculus with Atoms. In *Procs. CSL 2017*, volume 82 of *LIPIcs*, pages 30:1–30:21, 2017.

[29] B. Klin and M. Szynwelski. SMT solving for functional programming over infinite structures. In *MFSP*, volume 207, pages 57–75, 2016.

[30] E. Kopczyński and S. Toruńczyk. Lois: Syntax and semantics. In *Procs. of POPL 2017*, pages 586–598, 2017.

[31] D. Kozen. Results on the propositional μ-calculus. *Theor. Comp. Sci.*, 27(3):333 – 354, 1983.

[32] Hui-Min Lin. Predicate μ-calculus for mobile ambients. *Journal of Computer Science and Technology*, 20(1):95–104, 2005.

[33] S. Lösch and A. M. Pitts. Denotational semantics with nominal scott domains. *J. ACM*, 61(4):27:1–27:46, 2014.

[34] F. Neven, T. Schwentick, and V. Vianu. Towards regular languages over infinite alphabets. In *MFCS*, pages 560–572, 2001.

[35] J. Ochremiak. *Extended constraint satisfaction problems*. PhD thesis, University of Warsaw, 2016.

[36] J. Parrow, J. Borgström, L.-H. Eriksson, R. Gutkovas, and T. Weber. Modal logics for nominal transition systems. In *Procs. CONCUR 2015*, volume 42 of *LIPIcs*, pages 198–211, 2015.

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

[38] L. Segoufin. Automata and logics for words and trees over an infinite alphabet. In *Procs. CSL 2006*, volume 4207 of *Lecture Notes in Computer Science*, pages 41–57, 2006.

[39] W. Thomas and T. Wilke. Automata, logics, and infinite games: A guide to current research. *Bulletin of Symbolic Logic*, 10(1):114–115, 2004.

[40] N. Tzevelekos. Fresh-register automata. *SIGPLAN Not.*, 46(1):295–306, 2011.

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