

Decidability Results for Fragments of First-Order Logic via a Symbolic Model Property

Neta Elad 

Tel Aviv University, Israel

Sharon Shoham 

Tel Aviv University, Israel

Abstract

Recently, symbolic structures were proposed as finite representations of potentially infinite first-order structures, where Linear Integer Arithmetic terms and formulas define the domain and interpretations of a structure. We generalize symbolic structures to use any base theory that admits a standard model. Symbolic structures induce a symbolic model property, which holds for a fragment of first-order logic if every satisfiable formula in the fragment has a symbolic model. The symbolic model property implies decidability, since the model-checking problem for symbolic structures is decidable. We use the symbolic model property to prove decidability for several fragments that extend the fragment of stratified formulas, relaxing the quantifier-alternation constraints by allowing one sort to have self-looping functions, under certain restrictions. To establish the symbolic model property for these fragments we construct a symbolic model for a formula from an arbitrary model. The construction and its correctness are proved in a generic fashion, which may be instantiated to other similarly restricted fragments.

2012 ACM Subject Classification Theory of computation → Logic and verification

Keywords and phrases first-order logic, decidability, symbolic structures

Digital Object Identifier 10.4230/LIPIcs.LICS.2026.40

Related Version *Extended Version*: <https://doi.org/10.48550/arXiv.2605.05840> [16]

Funding The research leading to these results has received funding from the European Research Council under the European Union’s Horizon 2020 research and innovation programme (grant agreement No [759102-SVIS]). This research was partially supported by Israel Science Foundation (ISF) grant No. 2117/23.

1 Introduction

In this paper we study a formalism for representing infinite models of formulas in first-order logic (FOL) and use this formalism to prove decidability results for new fragments of FOL. Our investigation is motivated by the use of FOL for verification. First-order logic offers an expressive language for specifying verification conditions, logical formulas that capture the correctness of a computer program or system, as well as means for automation, due to its complete proof system.

In particular, we focus on the use of quantified FOL formulas for modeling systems with unbounded domains, where axiomatically-defined order relations are used to abstract important primitives that are not directly definable in FOL, e.g., reachability in linked lists or tree-shaped data-structures. FOL offers a natural way to express verification conditions of such systems, allowing the user to write specifications that match their mental model of the system, and verify the correctness of the system (see, e.g., [20, 27, 30, 37]).

However, FOL is undecidable in general, and thus the verification process may fail. Completeness of FOL does guarantee that valid formulas admit finite proofs, but there are no such guarantees on counter-models of invalid formulas. An FOL-based verification tool may



© Neta Elad and Sharon Shoham;
licensed under Creative Commons License CC-BY 4.0

41st Annual Symposium on Logic in Computer Science (LICS 2026).

Editors: Claudia Faggian and Joost-Pieter Katoen; Article No. 40; pp. 40:1–40:27



Leibniz International Proceedings in Informatics

LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

get stuck when trying to verify a system due to counter-models it is unable to find, leaving the user with no actionable feedback. This is particularly common during intermediate steps in the verification process, when the user still refines the system’s specifications and fixes bugs in the system.

One approach to handle these issues is by limiting verification conditions to a fragment of FOL that enjoys a finite model property, which means that every satisfiable formula admits a finite model. The finite model property implies decidability, and, importantly, ensures that whenever a counter-model exists, it may be found and presented to the user, providing much-desired feedback.

One very successful fragment in this regard is the fragment of stratified formulas (SF) [31], the many-sorted variant of the Effectively Propositional fragment (EPR) [34, 24]. The finite model property of SF is guaranteed by requiring acyclicity of the quantifier-alternation graph of each formula, where vertices correspond to sorts and edges correspond to $\forall\exists$ quantifier alternations (either explicit or implicit via function symbols). Though useful [31, 28, 26], the fragment of SF is rather limiting in its syntactic constraints, and may require nontrivial amount of work from the user to fit specifications of complex systems into the fragment [36].

In this paper, we pursue an alternative approach that goes beyond a finite model property. We explore effective representations of potentially infinite models, and use them to establish decidability results for fragments of FOL that do not admit a finite model property.

We build on the notion of *symbolic structures*, introduced in [15] as finite representations of potentially infinite models. Symbolic structures use the standard model of the integers and the theory of Linear Integer Arithmetic (LIA) to represent first-order structures. In symbolic structures, the domain is given symbolically by finitely many nodes, each of which represents a potentially infinite set of integers, and interpretations of signature symbols are defined with terms and formulas in the theory of LIA. This is akin to first-order interpretations (over the standard model of the integers) with similarly-behaved elements explicitly grouped together into symbolic nodes.

We start by generalizing the formalism of symbolic structures to other theories, producing a definition that is parametric by the *base theory* underlying the symbolic structure. Our generalization allows any base theory that admits a standard model (i.e., any theory of a structure). Given such a base theory, the nodes in our generalized symbolic structures represent sets of elements of the standard model of the theory, and interpretations are defined using terms and formulas in the language of the theory. This generalization matches how first-order interpretations may be defined over any base structure.

A key property of symbolic structures is that they allow reducing model-checking, which determines whether a given symbolic structure satisfies a given first-order formula, to validity of some formula in the base theory. Thus, model-checking is decidable for symbolic structures over decidable base theories (which is unsurprising due to the connection between symbolic structures and first-order interpretations). Decidability of model-checking induces a procedure for providing counter-models to verification conditions by enumerating and checking symbolic structures. This procedure may not always halt, since undecidability of validity in FOL implies that some invalid formulas do not have symbolic counter-models.

Our main contribution is proving decidability for new fragments of FOL via a *symbolic model property*, which states that every satisfiable formula has a symbolic model. Due to decidability of model-checking for symbolic structures, the symbolic model property trivially implies decidability. However, establishing the symbolic model property may be difficult and is the major problem this paper tackles. Interestingly, and in contrast to the fragment of

stratified formulas, the fragments we consider do not enjoy a finite model property, and some formulas in them may only admit infinite models. Still, the symbolic model property ensures that when verification fails, a symbolic counter-model can be produced.

To define the decidable fragments, we introduce the Ordered Self-Cycle (OSC) family of fragments, where each fragment, $\text{OSC}[\alpha]$, is parameterized by an axiomatization α of an order relation $\prec$ over some sort s^∞ . $\text{OSC}[\alpha]$ extends the fragment of stratified formulas by relaxing the acyclicity requirement of the quantifier-alternation graph, allowing function symbols that introduce self-cycles over sort s^∞ in the graph, under certain restrictions. The OSC family is a generalization of the decidable fragment presented in [15], which corresponds to $\text{OSC}[\text{TOT}]$, where TOT axiomatizes a total order. Decidability of $\text{OSC}[\text{TOT}]$ is proved in [15] via a symbolic model property, using LIA as the base theory.

In this paper, we establish a symbolic model property for additional fragments in the OSC family, using both LIA and a decidable theory of strings as base theories. Most notably, we use the base theory of strings to show the symbolic model property of the Prefix-Ordered Self-Cycle fragment, $\text{OSC}[\text{PREF}]$, where PREF axiomatizes a prefix order, a common construct in verification that arises naturally when modeling reachability in acyclic tree-shaped data-structures or in acyclic linked lists with sharing. We further prove symbolic model properties for several variants of $\text{OSC}[\text{TOT}]$ and $\text{OSC}[\text{PREF}]$.

Our proofs establish a symbolic model property through a generic proof recipe, which extends and simplifies the proof given in [15] for $\text{OSC}[\text{TOT}]$. The proofs work by constructing a symbolic model for a formula from an arbitrary model of the formula. The construction is defined in several steps, some of which are general, while others depend on the specific order axiom α and base theory used.

Finally, we note that we have implemented a proof-of-concept tool for exploring generic symbolic structures by extending the open-source **FEST** [15] Python library. Our implementation generalizes **FEST** to support different base theories in a modular way, in particular enabling model-checking of symbolic structures over the base theory of strings.

In summary, this paper offers the following contributions: **(1)** a general definition of symbolic structures, finite representations of potentially infinite models that can be viewed and acted upon (Section 4); **(2)** the introduction of the family of Ordered Self-Cycle fragments, which are parameterized by an order axiomatization (Section 5); and **(3)** decidability proofs for several OSC fragments, in particular the Prefix-Ordered Self-Cycle fragment $\text{OSC}[\text{PREF}]$, as well as variants of $\text{OSC}[\text{PREF}]$ and $\text{OSC}[\text{TOT}]$ (Section 6). The rest of the paper is organized as follows: related work is discussed in Section 2, preliminaries are given in Section 3, Section 7 gives a brief overview of the implementation of generic symbolic structures, and Section 8 concludes. Due to space considerations, proofs of lesser salience are omitted. See [16] for an extended version with detailed proofs.

2 Related Work

Finite model property. A long line of work has identified syntactic fragments of FOL for which satisfiability is shown to be decidable by establishing a finite model property. Classical results include the Ackermann, Gödel, Löb and Bernays-Schönfinkel-Ramsey fragments, as well as later refinements and extensions studied extensively in logic and finite model theory [2, 17, 25, 19, 29, 9, 1, 21, 38]. Though some of these fragments admit specialized decision procedures, they all enjoy a finite model property. For a detailed survey, see [38]. Related to this line of work are decidability results for satisfiability over finite structures (e.g., [35, 13, 5]). All of these works are similar in spirit to our symbolic-model-property approach, but distinct as the fragments we consider do not admit a finite model property.

Decidability and infinite models. The seminal works on monadic second-order logic (MSO) [10, 11, 33] prove decidability for fragments that do not admit a finite model property. There, the specific structures of lines (S1S) and trees (S2S) are considered, and finite-state automata are used to represent infinite models. A complementary line of work studies the construction of infinite models via proof-theoretic methods [3, 32], using saturated sets of clauses to witness the existence of models. Using the symbolic model property to prove decidability of FOL fragments was introduced in [15], which this work generalizes to other fragments of FOL and symbolic structures over different base theories.

Representing infinite structures. Symbolic structures can also be expressed as first-order interpretations (see, e.g., [18]) over the standard structure of the base theory, by indirectly encoding the nodes in the symbolic domain using the dimensionality of the FO interpretation. The different nodes in a symbolic structure provide a natural way to group and define similarly-behaved elements. An alternative approach to representing infinite structures is by using automatic structures [7, 22, 4, 8], where automata and regular languages are used. However, these works do not provide a method to automatically find a satisfying structure for a given formula, nor identify a decidable fragment where representable satisfying structures are guaranteed to exist.

3 Preliminaries

First-order logic. We consider many-sorted first-order logic with equality. We use the usual definitions of a signature Σ , terms over signature TERMS_Σ , formulas $\mathcal{L}_\Sigma$, first-order structures and the satisfaction relation. For a set of variables X , we denote by $\text{TERMS}_\Sigma(X)$ and $\mathcal{L}_\Sigma(X)$ the sets of terms and, respectively, formulas over Σ with free variables in X . In particular, we write $\text{TERMS}_\Sigma(\emptyset)$ for the set of ground terms and $\mathcal{L}_\Sigma(\emptyset)$ for the set of sentences (formulas without free variables) over Σ .

Theories of structures. In this paper we consider first-order theories of structures. Given a first-order structure $\mathcal{M}$ for signature Σ , the theory of $\mathcal{M}$, denoted $\text{Th}(\mathcal{M})$, is the set of all sentences satisfied by $\mathcal{M}$, i.e., $\text{Th}(\mathcal{M}) = \{\varphi \in \mathcal{L}_\Sigma(\emptyset) \mid \mathcal{M} \models \varphi\}$. For $\mathcal{T} = \text{Th}(\mathcal{M})$, we say that a sentence $\varphi \in \mathcal{L}_\Sigma(\emptyset)$ is *valid* in $\mathcal{T}$ (or $\mathcal{T}$ -valid), denoted $\models^{\mathcal{T}} \varphi$, if $\varphi \in \mathcal{T}$, i.e., $\mathcal{M} \models \varphi$. We say that a formula is *satisfiable* in $\mathcal{T}$ (or $\mathcal{T}$ -satisfiable) if its existential closure is $\mathcal{T}$ -valid. We say that $\mathcal{T}$ is *decidable* when there exists an algorithm that decides whether a sentence is $\mathcal{T}$ -valid.

The theory of Linear Integer Arithmetic (LIA). The theory of LIA is the theory of the structure $\mathcal{M}^{\text{LIA}}$ for the signature

$$\Sigma^{\text{LIA}} = \{c_z \mid z \in \mathbb{Z}\} \cup \{+(\cdot, \cdot)\} \cup \{<(\cdot, \cdot)\}$$

over the single sort **int**. The domain of $\mathcal{M}^{\text{LIA}}$ consists of the integers $\mathbb{Z}$, and the interpretations of the constant symbols c_z , function symbol $+$ and relation symbol $<$ are defined in the usual way. With abuse of notation, we sometimes write z for c_z . Recall that the theory of LIA is decidable (see, e.g., [12]).

A decidable theory of strings. Given a finite alphabet Δ , the language of regular expressions over Δ , denoted $\mathcal{R}_\Delta$, is defined by the following grammar:

$$r \in \mathcal{R}_\Delta ::= \epsilon \mid \delta \in \Delta \mid (r_1 \cdot r_2) \mid (r_1 \mid r_2) \mid (r^*).$$

We further denote by $(r?)$ the regular expression $(r \mid \epsilon)$ and by (r^+) the regular expression $(r \cdot (r^*))$. We consider the decidable theory of strings over Δ , denoted STR_Δ , which is the theory of the structure $\mathcal{M}_\Delta^{\text{STR}}$ (also known as $\mathbf{S}_{\text{reg}}$ [6]) for the signature

$$\Sigma_\Delta^{\text{STR}} = \{c_\epsilon\} \cup \{\ell_\delta(\cdot) \mid \delta \in \Delta\} \cup \{\sqsubset(\cdot, \cdot)\} \cup \{ /r/(\cdot) \mid r \in \mathcal{R}_\Delta \}$$

over the single sort **string**. The domain of $\mathcal{M}_\Delta^{\text{STR}}$ is the set of finite strings, also known as *words*, over Δ , the constant symbol c_ϵ is interpreted as the empty string ϵ , the unary function symbols ℓ_δ (“last”) are interpreted as appending the letter δ to a string, the binary relation symbol $\sqsubset$ is interpreted as the strict prefix relation over strings, and the unary relation symbols $/r/$ are interpreted as the languages (i.e., sets of words) of the regular expressions r . With abuse of notation we sometimes use ϵ as a constant symbol and $t \cdot \delta$ (for $\delta \in \Delta$) as shorthand for $\ell_\delta(t)$. As shown in [6], the theory STR_Δ is decidable for any finite Δ .

4 Symbolic Structures

In this section we introduce a generalization of the symbolic structures formalism [15]. Throughout the section we fix a theory $\mathcal{T} = \text{Th}(\mathcal{M}^\mathcal{T})$ of a structure $\mathcal{M}^\mathcal{T} = (\mathcal{D}^\mathcal{T}, \mathcal{I}^\mathcal{T})$ for a single-sorted signature $\Sigma^\mathcal{T}$, and define symbolic structures over $\mathcal{T}$. We use the shorthand $\mathcal{L}^\mathcal{T}$ and $\text{TERMS}^\mathcal{T}$ for $\mathcal{L}_{\Sigma^\mathcal{T}}$ and $\text{TERMS}_{\Sigma^\mathcal{T}}$ respectively.

Given a signature Σ (not to be confused with $\Sigma^\mathcal{T}$), symbolic structures for Σ are finite representations of (explicit) first-order structures for Σ , where the domain of each sort is stitched up piecewise from subsets of the domain of the standard model $\mathcal{M}^\mathcal{T}$ of the underlying base theory $\mathcal{T}$; each such subset is summarized by a node in the symbolic structure. Interpretations of constant, function and relation symbols from Σ are given symbolically in symbolic structures, using terms and formulas in the language of $\Sigma^\mathcal{T}$.

The symbolic representation allows to check if the structure represented by a symbolic structure satisfies a formula φ over Σ by essentially substituting symbols in Σ appearing in φ by their symbolic definitions, transforming φ into a formula over $\Sigma^\mathcal{T}$ that is $\mathcal{T}$ -valid iff the explicit structure satisfies φ . Formally, symbolic structures and their explications are defined as follows.

► **Definition 4.1** (Symbolic structure). *A symbolic structure over base theory $\mathcal{T}$ for a first-order signature Σ with sorts $\mathcal{S}$ is a triple $S = (\mathcal{D}^S, \mathcal{B}^S, \mathcal{I}^S)$, defined as follows.*

- $\mathcal{D}^S$ is a function from $\mathcal{S}$ to finite, non-empty sets of nodes, where $\mathcal{D}^S(s_1) \cap \mathcal{D}^S(s_2) = \emptyset$ for every $s_1 \neq s_2 \in \mathcal{S}$. We refer to $\bigcup_{s \in \mathcal{S}} \mathcal{D}^S(s)$ as the symbolic domain of S .
- $\mathcal{B}^S$ is a function from the symbolic domain to $\mathcal{T}$ -satisfiable formulas in $\mathcal{L}^\mathcal{T}(\{x\})$ (i.e., formulas over $\Sigma^\mathcal{T}$ with at most one free variable x). For each node n in the symbolic domain, $\mathcal{B}^S(n)$ is called the bound formula of n .
- $\mathcal{I}^S$ is a function that maps each symbol $s \in \Sigma$ to its symbolic interpretation $\mathcal{I}^S(s)$, also denoted as s^S .

The symbolic interpretation of a constant symbol $c: s$ is a pair $c^S = \langle n, t \rangle \in \mathcal{D}^S(s) \times \text{TERMS}^\mathcal{T}(\emptyset)$ such that

$$\models^\mathcal{T} \mathcal{B}(n)[x \mapsto t]. \quad (1)$$

The symbolic interpretation of a function symbol $f: s_1 \times \dots \times s_m \rightarrow s$ is a function $f^S: \mathcal{D}^S(s_1) \times \dots \times \mathcal{D}^S(s_m) \rightarrow \mathcal{D}^S(s) \times \text{TERMS}^\mathcal{T}(\{x_1, \dots, x_m\})$ such that whenever $f^S(n_1, \dots, n_m) = \langle n, t \rangle$,

$$\models^\mathcal{T} \forall x_1, \dots, x_m. \left(\bigwedge_{j=1}^m \mathcal{B}(n_j)[x \mapsto x_j] \right) \rightarrow \mathcal{B}(n)[x \mapsto t]. \quad (2)$$

The symbolic interpretation of a relation symbol $R: \mathbf{s}_1 \times \dots \times \mathbf{s}_m$ is a function $R^S: \mathcal{D}^S(\mathbf{s}_1) \times \dots \times \mathcal{D}^S(\mathbf{s}_m) \rightarrow \mathcal{L}^\mathcal{T}(\{x_1, \dots, x_m\})$.

We call conditions (1) and (2) the well-definedness conditions of S .

A symbolic structure S is a finite representation of an *explicit* structure $\mathcal{E}(S)$. Next we define the explicit structure represented by a symbolic structure.

► **Definition 4.2** (Explication). Let $S = (\mathcal{D}^S, \mathcal{B}^S, \mathcal{I}^S)$ be a symbolic structure over base theory $\mathcal{T}$ for a first-order signature Σ with sorts $\mathcal{S}$. The explication of S , denoted $\mathcal{E}(S)$ is the first-order structure $\mathcal{E}(S) = (\mathcal{D}^{\mathcal{E}(S)}, \mathcal{I}^{\mathcal{E}(S)})$ for Σ , defined as follows.

- The explicit domains $\mathcal{D}^{\mathcal{E}(S)}$ of sorts are given by explicating each node in the domain of a sort into a set of explicit elements from the domain of $\mathcal{M}^\mathcal{T}$ and taking the disjoint union of these sets. For a symbolic node n we denote its explicit elements by

$$\mathcal{E}(n) \triangleq \{ \langle n, d \rangle \mid d \in \mathcal{D}^\mathcal{T} \text{ and } \mathcal{M}^\mathcal{T}, [x \mapsto d] \models \mathcal{B}(n) \},$$

and we define the explicit domain as

$$\mathcal{D}^{\mathcal{E}(S)}(\mathbf{s}) \triangleq \bigcup_{n \in \mathcal{D}^S(\mathbf{s})} \mathcal{E}(n)$$

- The explicit interpretations $\mathcal{I}^{\mathcal{E}(S)}$ of symbols are given by explicating the symbolic interpretations according to $\mathcal{M}^\mathcal{T}$.

The interpretation of a constant symbol $c: \mathbf{s} \in \Sigma$ where $c^S = \langle n, t \rangle$ is the explicit element

$$c^{\mathcal{E}(S)} \triangleq \langle n, t^{\mathcal{M}^\mathcal{T}} \rangle,$$

where $t^{\mathcal{M}^\mathcal{T}}$ is defined recursively in the usual way.

The interpretation of a function symbol $f: \mathbf{s}_1 \times \dots \times \mathbf{s}_m \rightarrow \mathbf{s} \in \Sigma$ is the function $f^{\mathcal{E}(S)}: \mathcal{D}^{\mathcal{E}(S)}(\mathbf{s}_1) \times \dots \times \mathcal{D}^{\mathcal{E}(S)}(\mathbf{s}_m) \rightarrow \mathcal{D}^{\mathcal{E}(S)}(\mathbf{s})$, defined such that if $f^S(n_1, \dots, n_m) = \langle n, t \rangle$ then

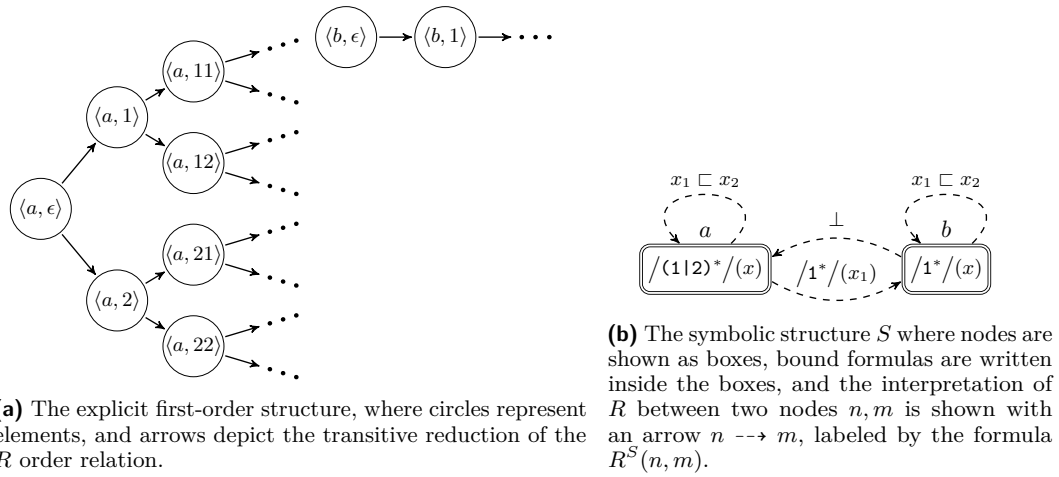
$$f^{\mathcal{E}(S)}(\langle n_1, d_1 \rangle, \dots, \langle n_m, d_m \rangle) \triangleq \langle n, t^{\mathcal{M}^\mathcal{T}}_{[x_1 \mapsto d_1, \dots, x_m \mapsto d_m]} \rangle.$$

The interpretation of a relation symbol $R: \mathbf{s}_1 \times \dots \times \mathbf{s}_m \in \Sigma$ is the relation

$$R^{\mathcal{E}(S)} \triangleq \left\{ (\langle n_1, d_1 \rangle, \dots, \langle n_m, d_m \rangle) \left| \begin{array}{l} \langle n_1, d_1 \rangle \in \mathcal{E}(n_1), \\ \dots, \\ \langle n_m, d_m \rangle \in \mathcal{E}(n_m), \text{ and} \\ \mathcal{M}^\mathcal{T}, [x_1 \mapsto d_1, \dots, x_m \mapsto d_m] \\ \models R^S(n_1, \dots, n_m) \end{array} \right. \right\}$$

► **Example 4.3.** Consider the infinite first-order structure depicted in Figure 1a over the single-sorted signature $\Sigma = \{R(\cdot, \cdot)\}$ with sort $\mathbf{s}$ and a binary relation R , where we have infinitely many elements ordered in a prefix order, forming a full binary tree that has an infinite tail “after” one of its branches. The figure depicts the transitive reduction of the binary relation R by arrows.

Figure 1b depicts a symbolic structure S over the base theory of $\text{STR}_{\{1,2\}}$ for Σ , whose explication is the structure from Figure 1a, where the a -elements are grouped together into one node and the b -elements are grouped into a second node. Accordingly, the symbolic structure has two nodes in its symbolic domain, $\mathcal{D}^S(\mathbf{s}) = \{a, b\}$, where a represents an



■ **Figure 1** An example of an infinite first-order structure for the signature $\{R(\cdot, \cdot)\}$, presented explicitly and symbolically. The structure contains infinitely many elements, ordered in a prefix order encoded by R . The elements are of two “kinds”: a -elements and b -elements. The a -elements are ordered in a full, infinite binary tree, represented by binary strings over alphabet 1, 2, whereas the b -elements form an infinite tail for the “top” branch of 1-strings of the a -binary tree.

infinite set of binary words, $\mathcal{B}^S(a) = / (1|2)^* /$, and b represents an infinite set of unary words, $\mathcal{B}^S(b) = / 1^* /$. The relation symbol R is defined symbolically as:

$$R^S(a, a) \triangleq x_1 \sqsubset x_2, \quad R^S(b, a) \triangleq \perp, \quad R^S(a, b) \triangleq / 1^* / (x_1), \quad R^S(b, b) \triangleq x_1 \sqsubset x_2,$$

expressing that for any two a -elements or b -elements, their order (according to R) matches the natural prefix order on their respective words, that no b -element comes before an a -element, and that all b -elements come after a -elements with words matching the $/ 1^* /$ language.

► **Theorem 4.4** (Model-checking for symbolic structures). *Given a symbolic structure S over base theory $\mathcal{T}$ for signature Σ and sentence $\varphi \in \mathcal{L}_\Sigma(\emptyset)$, there exists a computable sentence $\varphi^S \in \mathcal{L}^\mathcal{T}(\emptyset)$ (given by the model-checking transformation described in [15]) such that $\mathcal{E}(S) \models \varphi \iff \models^\mathcal{T} \varphi^S$.*

► **Remark 4.5.** For symbolic structures that only use quantifier-free formulas in bound formulas and interpretations, the model-checking transformation preserves the quantification structure of the original formula. Thus, for a base theory $\mathcal{T}$ that is only decidable for the quantifier-free fragment of $\Sigma^\mathcal{T}$ model-checking would still be decidable, but only for the quantifier-free fragment of FOL.

In the sequel, we focus on symbolic structures over decidable base theories, which ensures that model-checking is decidable.

5 The Ordered Self-Cycle (OSC) Family of Fragments

In this section we define the OSC family of fragments and state our decidability results. We start with the definition of the $\text{OSC}[\alpha]$ fragments, which are parameterized by a sentence α that axiomatizes an order relation. We then present the various kinds of orders considered in this paper. Finally we define the simplified $\text{OSC}^*[\alpha]$ fragments, which are used in our decidability proofs.

5.1 Fragments Definition

The Ordered Self-Cycle (OSC) family of first-order logical fragments is a set of extensions of the fragment of stratified formulas (SF) [31]. We start by recalling the properties of SF. The SF fragment is the many-sorted variant of the Effectively PROpositional fragment (EPR) [34]. SF establishes a finite model property by requiring the quantifier-alternation graph of each formula in SF to be acyclic. The vertices of the quantifier-alternation graph correspond to sorts in a formula, and edges correspond to $\forall\exists$ quantifier alternations, either explicit in the formula or implicitly given by function symbols. In SF, the finite model property stems from the fact that after Skolemization, there are only finitely many ground terms, which allows complete instantiation of the universal Skolemized formula, as in EPR.

The OSC family of fragments relaxes the quantifier-alternation restrictions of SF for a single sort, s^∞ , equipped with some axiomatically defined strict order relation, $\prec: s^\infty \times s^\infty$. We denote by STRICT the axiomatization of $\prec$ as a strict order relation, given by the conjunction of the following formulas:

$$\begin{aligned} \forall x. \neg(x \prec x) & \quad (\text{irreflexivity}) \\ \forall x, y, z. (x \prec y \wedge y \prec z) \rightarrow x \prec z & \quad (\text{transitivity}) \end{aligned}$$

Formally, the OSC family is defined as follows.

► **Definition 5.1 (OSC).** *The OSC family of fragments considers first-order signatures that include a distinguished sort s^∞ and a binary relation symbol $\prec: s^\infty \times s^\infty$. Given such a signature Σ and a sentence α such that $\alpha \models \text{STRICT}$ and α is a single-sorted formula over the s^∞ sort, the $\text{OSC}[\alpha]$ fragment consists of sentences of the form $\alpha \wedge \varphi$, where:*

1. *The formula φ uses a single variable of sort s^∞ .*
2. *Function and relation symbols other than $\prec$ used in φ have at most one argument of sort s^∞ .*
3. *In the quantifier-alternation graph of φ , the only cycles are self-loops at s^∞ , and the only outgoing edges from s^∞ are to s^∞ , resulting from function symbols whose range is s^∞ .*
4. *All nested function-application terms of sort s^∞ appearing in φ are ground.*

5.2 Decidability Results for Fragments of OSC

In this section we give an overview of the OSC fragments for which we prove decidability. First, we consider a strict total order, denoted TOT, axiomatized as the conjunction of the following formulas:

$$\begin{aligned} \forall x. \neg(x \prec x) & \quad (\text{irreflexivity}) \\ \forall x, y, z. (x \prec y \wedge y \prec z) \rightarrow x \prec z & \quad (\text{transitivity}) \\ \forall x, y. x = y \vee x \prec y \vee y \prec x & \quad (\text{totality}) \end{aligned}$$

Next, we consider a strict prefix order, denoted PREF, which is axiomatized as the conjunction of the following:

$$\begin{aligned} \forall x. \neg(x \prec x) & \quad (\text{irreflexivity}) \\ \forall x, y, z. (x \prec y \wedge y \prec z) \rightarrow x \prec z & \quad (\text{transitivity}) \\ \forall x, y, z. (x \prec z \wedge y \prec z) \rightarrow (x = y \vee x \prec y \vee y \prec x) & \quad (\text{downwards totality}) \end{aligned}$$

Finally, we consider two variants of each of the above orders. The “progressive-successor” variant, denoted PROSUCC, where all functions are progressive and there exists a successor to every element, and the “regressive-predecessor” variant, denoted REGPRED, where all

```

action send(n: node, v: value):
  assume n == root or msg(n, v);
  sent(n, v)      := true;
  msg(left(n), v) := true;
  msg(right(n), v) := true;

```

■ **Figure 2** Simple message broadcast protocol for a network of nodes with a binary-tree topology.

functions are regressive and there exists a predecessor to every element. We consider every combination of each of the TOT and PREF axioms, together with each of the PROSUCC and REGPRED variants. Formally, we define PROSUCC to be the conjunction of

$$\bigwedge_{f \in \Sigma} \forall x. x \prec f(x) \quad (\text{progressivity})$$

$$\forall x. \exists y. x \prec y \wedge \forall z. x \prec z \rightarrow (z = y \vee y \prec z), \quad (\text{successor existence})$$

and REGPRED to be the conjunction of

$$\bigwedge_{f \in \Sigma} \forall x. f(x) \prec x \quad (\text{regressivity})$$

$$\forall x. \exists y. y \prec x \wedge \forall z. z \prec x \rightarrow (z = y \vee z \prec y). \quad (\text{predecessor existence})$$

► **Example 5.2.** For an example of the applicability of OSC[PREF] let us consider a distributed message broadcast protocol, in a network with a topology of a binary tree. The network has a root node and each node has two children. Each node maintains a set of values it received by messages. The root node may send any value to its children, while any other node may only send values it already received, as expressed in Figure 2. We encode the topology of the network by a prefix order $\prec$, axiomatized by PREF, having **root** be minimal according to PREF, and the functions **left** and **right**, encoding the children of a node respect the order. We further encode the **send** action by a formula over two copies of the signature that specifies how the **sent** and **msg** relations are modified by the action. We wish to prove the safety property that for any node n that received value v , there exists an ancestor n' (according to the order in tree topology) that sent that value: $\forall n: \mathbf{node}, v: \mathbf{value}. \exists n': \mathbf{node}. \mathbf{msg}(n, v) \rightarrow n' \prec n \wedge \mathbf{sent}(n', v)$. This property, along with the encoding of the the protocol is in OSC[PREF], but not in the more restrictive fragment of SF.

► **Theorem 5.3.** *The OSC[α] fragment is decidable for*

$$\alpha \in \{\text{TOT}, \text{TOT} \wedge \text{PROSUCC}, \text{TOT} \wedge \text{REGPRED}, \text{PREF}, \text{PREF} \wedge \text{PROSUCC}, \text{PREF} \wedge \text{REGPRED}\}.$$

The result for OSC[TOT] is a restatement of [15], while the other 5 results are new.

5.3 The Simplified OSC* Family of Fragments

In this section we define a simplified family of fragments, denoted OSC*, such that satisfiability in OSC is reducible to satisfiability in OSC*, a property that our decidability proofs for fragments of OSC build on. Roughly, formulas in OSC* are single-sorted and only contain simple $c, x, f(x), g(x)$ terms.

► **Definition 5.4 (OSC*).** *The OSC* family of fragments considers single-sorted first-order signatures with a binary relation symbol $\prec(\cdot, \cdot)$, where all other function and relation symbols are unary. The single sort is denoted s^∞ . Given such a signature Σ and a sentence α over Σ such that $\alpha \models \text{STRICT}$, the OSC*[α] fragment consists of sentences of the form $\alpha \wedge \varphi$, where:*

1. φ is a positive Boolean combination of formulas of the form $\forall x.\phi(x)$, where ϕ is quantifier-free; and
2. all terms appearing in ϕ are of the form c (for some constant symbol $c \in \Sigma$), x or $f(x)$ (for some unary $f \in \Sigma$).

Note that Σ may include multiple function symbols, all of which are unary and cause self-loops.

► **Theorem 5.5.** *There exists an algorithm that translates every formula $\alpha \wedge \varphi \in \text{OSC}[\alpha]$ to some formula $\alpha \wedge \varphi^* \in \text{OSC}^*[\alpha]$, such that there exists a model $M \models \alpha \wedge \varphi$ iff there exists a model $M^* \models \alpha \wedge \varphi^*$. Moreover, M is computable from M^* .*

The model-preserving translation leverages key properties of OSC. First, the restrictions of OSC guarantee that after Skolemization, the signature of OSC formulas can generate only finitely many ground terms for sorts other than s^∞ , similarly to SF. This fact allows us to Skolemize and then “instantiate” formulas in OSC with all ground terms of non- s^∞ sorts, and arrive at an equi-satisfiable universal formula where the only variables are of sort s^∞ . Next, we replace applications of function and relation symbols by fresh unary symbols for each tuple of non- s^∞ ground terms appearing in them, considering all possible equalities between ground terms. Finally, we perform a sort of flattening of nested ground terms of s^∞ , iteratively replacing each $f(t)$ term with some fresh constant c^* and adding the axiom $\forall x.x = t \rightarrow f(x) = c^*$ to the formula.

► **Corollary 5.6.** *If $\text{OSC}^*[\alpha]$ is decidable, then so is $\text{OSC}[\alpha]$.*

► **Note 5.7.** Since signatures in OSC^* are single-sorted, we refer to the domains $\mathcal{D}^M$ of explicit structures and the domains $\mathcal{D}^S$ of symbolic structures as sets of elements, respectively nodes, i.e., with abuse of notation, $\mathcal{D}^M = \mathcal{D}^M(s^\infty)$ and $\mathcal{D}^S = \mathcal{D}^S(s^\infty)$.

6 Decidability via a Symbolic Model Property

In this section we prove the various decidability results of this paper (Theorem 5.3), following and generalizing the proof method of [15]. We use symbolic structures over decidable base theories as means to prove decidability for several fragments of the OSC^* family: we show that every satisfiable formula in these fragments admits a satisfying symbolic structure over a decidable base theory. Since model-checking of such symbolic structures is decidable, this *symbolic model property* ensures that satisfiability is recursively enumerable, and since unsatisfiability of all FOL formulas is recursively enumerable, decidability follows. Formally:

► **Definition 6.1.** *Let $\mathcal{F}$ be a fragment of FOL. We say that $\mathcal{F}$ admits a symbolic model property when there exists a decidable theory $\mathcal{T}$ such that for every satisfiable sentence $\psi \in \mathcal{F}$ there exists a symbolic structure S over base theory $\mathcal{T}$ such that $\mathcal{E}(S) \models \psi$.*

► **Theorem 6.2.** *Every fragment of FOL that admits the symbolic model property is decidable.*

Next we provide a general recipe for establishing a symbolic model property for $\text{OSC}^*[\alpha]$ fragments (Section 6.1). In Section 6.2, we restate the symbolic-model-property proof of $\text{OSC}^*[\text{TOT}]$ using the proof recipe, and Sections 6.3–6.5 give the new decidability results for 5 additional fragments of OSC^* by instantiating the general proof recipe to prove symbolic model properties for them.

6.1 Abstract Proof Recipe for OSC^{*} Decidability

The proof of the symbolic model property for fragments of the OSC^{*} family works by showing that given a formula $\alpha \wedge \varphi \in \text{OSC}^*[\alpha]$ and a satisfying model $M \models \alpha \wedge \varphi$, a symbolic structure S over a certain decidable base theory $\mathcal{T}$ can be constructed out of M such that $\mathcal{E}(S) \models \alpha \wedge \varphi$.

Fix a model $M \models \alpha \wedge \varphi$, the construction of S and its correctness are derived from the following insight: since $\text{OSC}^*[\alpha]$ is made up of positive Boolean combinations of universal formulas with a single variable x , we can prove that M and $\mathcal{E}(S)$ are OSC^{*}-equivalent, i.e., satisfy the same OSC^{*} $[\alpha]$ formulas, by proving the following two properties:

► **Desired Property 6.3.** If $M \models \alpha$ then $\mathcal{E}(S) \models \alpha$.

► **Desired Property 6.4.** There exists a surjective function $\tau: \mathcal{D}^M \rightarrow \mathcal{D}^S$ such that for every atomic OSC^{*} formula $\gamma(x)$, element $d \in \mathcal{D}^M$, and explicit element $\langle \tau(d), e \rangle \in \mathcal{E}(\tau(d))$, we have $M, [x \mapsto d] \models \gamma$ iff $\mathcal{E}(S), [x \mapsto \langle \tau(d), e \rangle] \models \gamma$.¹

Desired Property 6.4 ensures that every node n in the symbolic domain of S “mimics” some explicit element d in the domain of M , in the sense that all the elements in the explication of n satisfy the same atomic formulas as d . In particular, since $M \models \alpha \wedge \varphi$, Desired Properties 6.3 and 6.4 imply that $\mathcal{E}(S) \models \alpha \wedge \varphi$ (by induction on the structure of φ). The construction of S is divided into two parts: a generic part, that is shared among the various fragments of OSC^{*} we consider, and a specific part that depends on α and informs the choice of the base theory. Similarly, the proof recipe breaks down the proofs of Desired Properties 6.3 and 6.4 into different lemmas, some of which are proved correct based on the generic part of the construction alone, while the rest are proved for each specific setting. This section lays out the overall proof recipe and the generic part of the construction, while the fragment- and base-theory-specific details are deferred to Sections 6.2–6.5.

6.1.1 Defining the Symbolic Domain

We define the symbolic domain $\mathcal{D}^S$ and the surjective function τ by partitioning the domain of M into finitely many equivalence classes, and set the symbolic domain to be the finite set of equivalence classes. To obtain Desired Property 6.4, we do this partitioning according to the atomic formulas. We observe that it suffices to consider atomic formulas over Σ where x appears, to which we refer as *atoms*:

► **Definition 6.5** (Atoms and non-ground terms). *An atom is an atomic formula γ where x appears as a free variable. We denote the set of all atoms by $\mathbb{A}$, and the set of non-ground terms appearing in atoms by $\mathbb{T} \triangleq \{x\} \cup \{f(x) \mid f \in \Sigma\}$.*

We say that an element $d \in \mathcal{D}^M$ satisfies an atom γ if $M, [x \mapsto d] \models \gamma$, and denote the set of all atoms d satisfies by $\Gamma(d)$:

$$\Gamma(d) \triangleq \{\gamma \in \mathbb{A} \mid M[x \mapsto d] \models \gamma\}.$$

We partition $\mathcal{D}^M$ into classes with identical sets of atoms, defining the equivalence class of an element $d \in \mathcal{D}^M$ as

$$[d] \triangleq \{d' \in \mathcal{D}^M \mid \Gamma(d') = \Gamma(d)\}.$$

For every class $[d]$ and elements $d_1, d_2 \in [d]$, $\Gamma(d_1) = \Gamma(d_2)$, and with abuse of notation we denote those shared atoms by $\Gamma([d])$.

¹ Recall that with abuse of notation we denote $\mathcal{D}^M = \mathcal{D}^M(s^\infty)$ and $\mathcal{D}^S = \mathcal{D}^S(s^\infty)$.

► **Lemma 6.6.** *The above partitioning induces an equivalence relation over $\mathcal{D}^M$, and, in particular, for every element $d \in \mathcal{D}^M$ and atom $\gamma \in \mathbb{A}$, $M, [x \mapsto d] \models \gamma \iff \gamma \in \Gamma(d) \iff \gamma \in \Gamma([d])$.*

The domain of the symbolic structure $\mathcal{D}^S$ is defined to be the set of all equivalence classes, and we use the names “node” and “class” for the elements of $\mathcal{D}^S$ interchangeably. We define the surjective mapping function $\tau: \mathcal{D}^M \rightarrow \mathcal{D}^S$ by mapping each element to its equivalence class, $\tau(d) \triangleq [d]$.

We use the name *regular node* for any node n where there exists some constant symbol c such that $x = c \in \Gamma(n)$, and otherwise use the name *summary node*. Intuitively, regular nodes will be explicated into a single element, while summary nodes will be explicated into (infinitely) many elements.

Note that $\mathcal{D}^S$ is a finite set, and we define an arbitrary order over its elements, denoted by $\ll$, which will be used in the sequel.

Given the above definitions, we say that an atom $\gamma \in \mathbb{A}$ is *observed* by a node $n \in \mathcal{D}^S$ if for every explicit element $\langle n, e \rangle \in \mathcal{E}(n)$, $\mathcal{E}(S), [x \mapsto \langle n, e \rangle] \models \gamma \iff \gamma \in \Gamma(n)$. Note that observance of all atoms implies the necessary conditions of Desired Property 6.4, which can thus be restated as the following:

► **Desired Property 6.7** (Atom observance). All atoms are observed by all nodes: for every node n , every explicit element $\langle n, e \rangle \in \mathcal{E}(n)$ and atom $\gamma \in \mathbb{A}$, $\gamma \in \Gamma(n) \iff \mathcal{E}(S), [x \mapsto \langle n, z \rangle] \models \gamma$.

The rest of the steps of the recipe explain how to achieve this property while ensuring that $\mathcal{E}(S) \models \alpha$, where different parts of the construction address different atoms. Namely, we divide the atoms into several different *kinds*:

Element atoms Atoms containing no function applications, internally sub-divided into *element relation atoms* of the form $P(x)$ where $P \in \Sigma$ is a unary relation; *element equivalence atoms* of the form $x = c$ where $c \in \Sigma$ is a constant symbol; and *element order atoms* of the form $x \prec c$ and $c \prec x$.

Image atoms Atoms of the form $\gamma[x \mapsto f(x)]$, where γ is an element atom, and $f \in \Sigma$ is a unary function symbol. For a given function symbol f , we use the name *f-image atoms* for the set of image atoms associated with it.

Mixed order atoms Atoms of the form $t \prec t', t = t'$, where t, t' are non-ground terms in $\mathbb{T}$.

Then, for each kind of atom, a different part of the construction is important. Some are generic and others specific. We shall see that for element and image atoms, Desired Property 6.7 can be proved independently of the order axioms and base theory in a generic fashion, while for the mixed order atoms the proof is more involved, and hinges on the specifics of the order and base theory.

6.1.2 Observing Element Equivalence Atoms via Interpretation of Constants and Bound Formulas of Regular Nodes

Recall that nodes n where $x = c \in \Gamma(n)$ for some constant symbol c are regular nodes. The following lemma ensures that every constant symbol is associated with a unique regular node.

► **Lemma 6.8.** *For every constant symbol $c \in \Sigma$, there exists exactly one regular node n such that $x = c \in \Gamma(n)$.*

Accordingly, for every constant symbol $c \in \Sigma$, we define its interpretation $c^S \triangleq \langle n, t^{\text{reg}} \rangle$ where n is the regular node where $x = c \in \Gamma(n)$, and t^{reg} is some ground term of the base theory. The choice of t^{reg} is theory-specific, and also depends on other theory-specific properties of the construction, in particular the interpretations of function symbols. We provide the specific term for each fragment in the corresponding sections.

To ensure the well-definedness of constants we set the bound formula of every regular node n to be $\mathcal{B}(n) \triangleq x = t^{\text{reg}}$.

From these definitions follows our first lemma towards proving Desired Property 6.7:

► **Lemma 6.9.** *All element equivalence atoms are observed.*

6.1.3 Observing Element Relation Atoms via Interpretation of Unary Relations

Since the atoms inducing the equivalence classes include atomic formulas of unary relations, the interpretation of unary relation symbols is defined straightforwardly. For every node $n \in \mathcal{D}^S$ and every unary relation $P \in \Sigma$, we define $P^S(n)$ as

$$P^S(n) \triangleq \begin{cases} \top & \text{if } P(x) \in \Gamma(n), \\ \perp & \text{otherwise,} \end{cases}$$

and derive the following lemma:

► **Lemma 6.10.** *All element relation atoms are observed.*

6.1.4 Observing Element Order Atoms via Interpretation of $\prec$ for Semi-regular Pairs

For pairs of nodes n, m where at least one of the nodes is regular (i.e., semi-regular pairs), we define the interpretation of the order $\prec$ by considering element order atoms, essentially inheriting the order interpretation from the order between elements and constants:

$$\prec^S(n, m) \triangleq \begin{cases} \top & \text{if } x = c \in \Gamma(n) \text{ and } c \prec x \in \Gamma(m) \\ \perp & \text{if } x = c \in \Gamma(n) \text{ and } c \prec x \notin \Gamma(m) \\ \top & \text{if } x = c \in \Gamma(m) \text{ and } x \prec c \in \Gamma(n) \\ \perp & \text{if } x = c \in \Gamma(m) \text{ and } x \prec c \notin \Gamma(n) \end{cases}$$

The correctness of this definition is given by the following two lemmas.

► **Lemma 6.11.** *At least one of the conditions above is true for semi-regular pairs, and when multiple conditions are true, they agree on the definition ($\top$ or $\perp$).*

► **Lemma 6.12.** *All element order atoms are observed.*

6.1.5 Observing Image Atoms by Choosing Targets for Function Interpretations

Recall that in symbolic structures function interpretations are defined by pairs of nodes and theory terms. In particular, for the unary functions of OSC^* we have $f^S: \mathcal{D}^S \rightarrow \mathcal{D}^S \times \text{TERMS}^T(\{x_1\})$. When constructing S we can define the function interpretations in two phases: first defining the function target node, in a way that will ensure observance of image atoms, then using the base theory to define the function terms, such that the functions are well-defined and observance of the mixed order atoms is ensured.

Given a node n and function f , we define its interpretation as $f^S(n) \triangleq \langle m_n^f, t_n^f \rangle$, where the target m_n^f is chosen such that all f -image atoms of n match the element atoms of m :

$$\gamma[x \mapsto f(x)] \in \Gamma(n) \iff \gamma \in \Gamma(m_n^f).$$

When multiple candidates exist, we take the least according to the arbitrary order $\ll$ (to ensure consistency among different definitions). The following lemma ensures that such node always exists.

► **Lemma 6.13.** *For every node n and function symbol f , there exists some node m_n^f such that for every element atom γ , $\gamma \in \Gamma(m_n^f) \iff \gamma[x \mapsto f(x)] \in \Gamma(n)$.*

► **Lemma 6.14.** *All image atoms are observed.*

The function terms t_n^f will be defined later, with the exception that if m_n^f is a regular node, then $t_n^f \triangleq t^{\text{reg}}$, as dictated by the well-definedness requirement of symbolic structures.

6.1.6 Observing Mixed Order Atoms via Function Terms, Bound Formulas of Summary Nodes and Interpretation of $\prec$ for Summary Nodes

Finally, the only atoms not yet handled by the previous steps are the mixed order atoms, whose handling is tightly coupled with the interpretation of the order symbol and the function terms, which are in turn entangled with the definition of bound formulas for summary nodes. Beyond the mixed order atoms, the interpretation of $\prec$ must also satisfy the order axiom α .

Thus far, the order interpretation was only defined for pairs of nodes where at least one is regular, and only function targets were defined. Defining the function terms for function interpretations, bound formulas of summary nodes and the interpretation of $\prec$ for summary nodes is heavily dependent on the specific order axiom α and imposes requirements on the base theory. In Sections 6.2–6.5 we complete the details and present the chosen base theories for the order axioms we consider.

The key idea for defining the remaining components is partitioning the summary nodes to *segments* according to their relation to constant symbols (i.e., according to the element order atoms). The exact criteria for segmenting is dependent on the specific order axiom α , and is chosen to allow us to split the remainder of the construction and the proof into two tasks: (i) the *inter-segment task*; and (ii) the *intra-segment task*.

The inter-segment task. This task defines the order interpretation between nodes of different segments such that mixed order atoms involving functions whose target nodes reside in different segments are observed (regardless of the definition of function terms).

The intra-segment task and embeddability. This task defines the order interpretation between nodes within the same segment, as well as the function terms and bound formulas, carefully examining the interaction between function terms and the order constraints to ensure that mixed order atoms also hold within a segment and that the resulting symbolic structure is well-defined. The base theory is key for this task. Namely, the base theory is used to define (i) an order interpretation within a summary node, which is expanded into an order between any two summary nodes within a segment by interleaving the nodes; (ii) function terms such that the required order on the non-ground terms $\mathbb{T}$ (as induced by the mixed order atoms) is maintained for every explicit element; and (iii) bound formulas that ensure

that function terms are well-defined, and that the order definition satisfies the order axioms. The combination of bound formulas and the order interpretation determines the *internal shape* of each summary node, i.e., the ordered set of explicit elements represented by the node. Defining the function terms in a way that observes the mixed order atoms requires the internal shape of every summary node (which may serve as the target for some function definition) to have the property of *finite embeddability*: the ability to embed within the node the set of non-ground terms $\mathbb{T}$, ordered in any way that adheres to the order axioms. Moreover, to support mixed order atoms involving the term x , we need the stronger property of *universal finite embeddability*: the theory terms produced must be able to embed the order around every explicit element of the node. The lifting of the order by interleaving explicit elements of same-segment nodes ensures that the same terms still comply with the mixed order atoms even when functions target different summary nodes within the same segment. Figuring out how to achieve and encode the universal finite embeddability property within the base theory is one of the key insights required for completing the proof, and we shall see how it dictates the internal shape of summary nodes in the proofs of Sections 6.2–6.5.

► **Remark 6.15.** As we shall see, the proofs given in Sections 6.2–6.5 actually show a stronger property: the fragments of OSC^* we consider enjoy a *bounded* symbolic model property, where every satisfiable formula has a satisfying symbolic structure with bounded size, where all bound formulas, function terms and relation formulas are taken from a finite set of theory terms and formulas, dependent only on syntactic properties of the formula.

6.2 Revisiting the Decidability Proof of the Totally-Ordered Self-Cycle Fragment

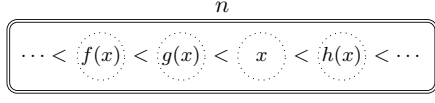
In this section we restate the proof of the symbolic model property of $\text{OSC}^*[\text{TOT}]$ from [15], using the more general proof framework we have introduced. This section is meant as a gentle introduction to the proof technique, using the simpler case of a total (linear) order. The construction for $\text{OSC}^*[\text{TOT}]$ uses the theory of LIA as the decidable base theory for the constructed symbolic structure.

Following the steps given in Section 6.1, we first define the regular node term to be $t^{\text{reg}} = 0$, meaning regular nodes have bound formulas $x = 0$, function terms are defined $t_n^f \triangleq 0$ whenever m_n^f is a regular node, and each constant c is interpreted as $\langle n, 0 \rangle$ where n is the regular node for which $x = c \in \Gamma(n)$.

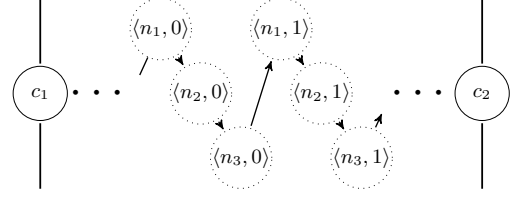
Next we define the segments of summary nodes in a way that lets us easily resolve the inter-segment task, namely, define the inter-segment order interpretation. For every summary node n we denote the set of constant symbols “smaller” than n by $C_{\prec}(n) \triangleq \{c \in \Sigma \mid c \prec x \in \Gamma(n)\}$, and we define a segment of summary nodes as a set of nodes with identical $C_{\prec}(\cdot)$. Using $C_{\prec}(\cdot)$ we define the inter-segment order as follows. Given two nodes n, m with segments $C_{\prec}(n) \neq C_{\prec}(m)$,

$$\prec^S(n, m) \triangleq \begin{cases} \top & \text{if } C_{\prec}(n) \subsetneq C_{\prec}(m), \\ \perp & \text{otherwise.} \end{cases}$$

This definition of the inter-segment order interpretation is dictated by the total order axiom TOT. Intuitively, if there exists some constant symbol c such that $c \prec x \in \Gamma(m)$ and $c \prec x \notin \Gamma(n)$ then by totality, $x \prec c \in \Gamma(n)$ (n is a summary node, thus $x = c \notin \Gamma(n)$), and by transitivity, it must be the case that all explicit elements of n are smaller than those of m . Note that given that $C_{\prec}(n) \neq C_{\prec}(m)$, the total order axiom ensures that if $C_{\prec}(n) \subsetneq C_{\prec}(m)$ does not hold, then the other direction must hold, i.e., $C_{\prec}(m) \subsetneq C_{\prec}(n)$.



■ **Figure 3** Example of an order induced by mixed order atoms within a single summary node.



■ **Figure 4** A segment with three interleaving summary nodes.

We have yet to define the interpretation of $\prec$ for same-segment summary nodes, thereby completing the definition of $\prec$. Yet, remarkably, due to the careful definition of the segments we can prove the correctness of the construction regarding the order axiom in a modular fashion, before defining the intra-segment order.

► **Lemma 6.16.** *For any definition of the intra-segment order that produces a strict total order within each segment, the interpretation $\prec^{\mathcal{E}(S)}$ is a total order, i.e., $\mathcal{E}(S) \models \text{TOT}$.*

Moreover, the inter-segment order definition ensures observance of mixed order atoms involving nodes of different segments:

► **Lemma 6.17.** *Let n be a summary node and let $f, g \in \Sigma$ be function symbols such that m_n^f and m_n^g reside within different segments, then the mixed order atoms $f(x) \prec g(x)$ and $g(x) \prec f(x)$ are observed. Further, for f such that m_n^f and n reside in different segments, $x \prec f(x)$ and $f(x) \prec x$ are observed.*

Next we approach the intra-segment task. Namely, we define the intra-segment order, function terms and bound formulas for summary nodes, in a way that observes all mixed order atoms. As explained in Section 6.1, this is achieved by the universal embeddability property, which ensures that for any possible set of constraints over the non-ground terms, i.e., any possible set of mixed order atoms in some summary node, these constraints can be satisfied for every explicit element associated with the summary node.

For example, consider a summary node n where the image atoms of n constrain the functions f, g and h to map back to n (i.e., n is the function target for f, g and h), and the mixed order atoms of n induce the following order: $f(x) \prec g(x) \prec x \prec h(x)$ (see Figure 3). We can see that for any explicit element of n , there must be two smaller explicit elements of n , which $f(x)$ and $g(x)$ will map to, and one larger explicit element of n , which $h(x)$ will map to. Further note that all of these explicit elements must be linearly ordered.

One natural shape for such a summary node is a bidirectionally-infinite line – the integers ordered by $<$. Notably, for the case of $\text{OSC}^*[\text{TOT}]$, this shape is uniform for all summary nodes. (We shall see that for prefix orders, two internal shapes are required). Using the base theory of LIA, we define the bound formulas for all summary nodes as simply $\mathcal{B}(n) = \top$.

Back in our example, we can now define the function terms for f, g and h by considering the arrangement of the terms $f(x), g(x), h(x)$ relative to x , according to the mixed order atoms. We choose function terms that realize this arrangement, for example, $t_n^f = x_1 - 2$, $t_n^g = x_1 - 1$ and $t_n^h = x_1 + 1$.

The discussion so far considered a single summary node. As explained earlier (Section 6.1.6), when a segment includes multiple summary nodes, we lift the order to nodes within the same segment by interleaving the explicit elements of the different summary nodes within the segment (which ensure function term compliance with mixed order atoms across

different target summary nodes within the segment). An example is given in Figure 4. The interleaving order is unimportant but should be consistent, and we use the arbitrary order $\ll$ to deterministically define it.

Finally, we provide the formal definition of the intra-segment order and the function terms. We define the intra-segment order between two nodes n, m as follows:

$$n \prec^S m \triangleq \begin{cases} x_1 < x_2 \vee x_1 = x_2 & \text{if } n \ll m, \\ x_1 < x_2 & \text{otherwise.} \end{cases}$$

To define the function terms for the cases where the target node of the function interpretation is a summary node (as illustrated above), we consider for every node its set of mixed order atoms. We order all non-ground terms $\mathbb{T}$ linearly, according to the order induced by the mixed order atoms. Note that two terms t, t' may occupy the same “spot” in the order, if $t = t' \in \Gamma(n)$ (e.g., $f(x) = g(x) \in \Gamma(n)$). Using the non-ground term x as an anchor, for each non-ground term $f(x)$ we denote by k_n^f the distance between it and the anchor term x , and define $t_n^f \triangleq x_1 + k_n^f$.

We formalize the correctness of the construction as follows:

► **Lemma 6.18.** *The intra-segment definition of the interpretation of $\prec$ produces a strict total order within each segment.*

► **Lemma 6.19.** *The constructed symbolic structure is well-defined.*

► **Lemma 6.20.** *All mixed order atoms are observed.*

This finishes the proof of Desired Property 6.7. Lemmas 6.16 and 6.18 imply that $\mathcal{E}(S) \models \text{TOT}$, and therefore:

► **Corollary 6.21.** $\mathcal{E}(S) \models \text{TOT} \wedge \varphi$.

6.3 Decidability of the Prefix-Ordered Self-Cycle Fragment

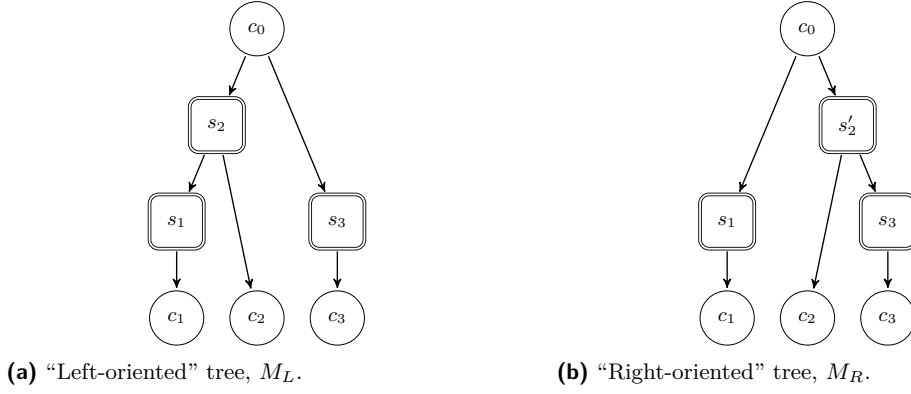
In this section we prove the main result of this paper: the decidability of the $\text{OSC}[\text{PREF}]$ fragment. To that end, we prove the symbolic model property for $\text{OSC}^*[\text{PREF}]$ using the STR base theory. As before, we follow the steps of Section 6.1. However, $\text{OSC}^*[\text{PREF}]$ requires more involved handling by the base theory at every step.

First, we fix the base theory to be STR_Δ over alphabet $\Delta = \{0, \dots, \ell\}$, where $\ell = |\mathbb{T}|$, i.e., ℓ is the number of non-ground terms $x, f(x), g(x), \dots$ that appear in atoms (and assumed to be at least 2). Intuitively, ℓ letters in the alphabet are sufficient to accommodate ℓ non-ground terms that may be incomparable (for some element), and the letter 0 is used to represent elements that are smaller than all of them.

The regular-node term is the empty string $t^{\text{reg}} = \epsilon$, thereby defining bound formulas for regular nodes, interpretations of constants, and function terms for regular-node targets.

Next we define the segments of summary nodes. Whereas in the total order, each segment represented some interval after a subset of the linearly ordered constants, for a prefix order, a finite set of constants may be ordered as a forest of rooted trees. Moreover, the space between a constant and its successor constants in a tree may itself be structured as a tree, inducing multiple segments.

► **Example 6.22.** Figures 5a and 5b depict two possible sets of segments between 4 constant symbols that are ordered in a tree. We denote by M_L, M_R two structures for a signature $\{c_0, c_1, c_2, c_3\}$, where for both we have $M_* \models c_0 \prec c_1 \wedge c_0 \prec c_2 \wedge c_0 \prec c_3$ and for any other



■ **Figure 5** Two possible sets of segments laid over 4 constants ordered in a tree. The arrows depict the transitive reduction.

pairs c_i, c_j , $M_* \not\models c_i \prec c_j$. However, M_L and M_R induce different segments, as in M_L there exists an element that is smaller than c_1 and c_2 but incomparable to c_3 , and in M_R there exists an element that is smaller than c_2 and c_3 but incomparable to c_1 .

Formally, we define the segments in the construction as pairs of sets of constant symbols, recording for each node n both the set of constant symbols smaller than n , $C_{\prec}(n) \triangleq \{c \in \Sigma \mid c \prec x \in \Gamma(n)\}$ (as in $\text{OSC}^*[\text{TOT}]$), as well as the set of constant symbols larger than n , $C_{\succ}(n) \triangleq \{c \in \Sigma \mid x \prec c \in \Gamma(n)\}$. We denote the segment of a node n by $\text{seg}(n) \triangleq (C_{\prec}(n), C_{\succ}(n))$.

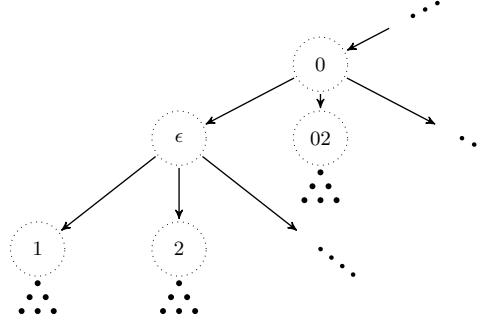
Having defined the segments of summary nodes, we now address the inter-segment task. We start with an example.

► **Example 6.23.** Going back to Example 6.22, let us focus on M_L of Figure 5a. Formally, we have 3 summary nodes s_1, s_2, s_3 where $\text{seg}(s_1) = (\{c_0\}, \{c_1\})$, $\text{seg}(s_2) = (\{c_0\}, \{c_1, c_2\})$ and $\text{seg}(s_3) = (\{c_0\}, \{c_3\})$. The element order atoms imply that s_1 is incomparable to s_3 , as otherwise, by transitivity we would expect $x \prec c_3 \in \Gamma(s_1)$ or $x \prec c_1 \in \Gamma(s_3)$. Similarly, s_2 and s_3 are incomparable. In contrast, since both s_1 and s_2 include the $x \prec c_1$ atom, by downwards totality they must be ordered. Moreover, s_2 must come before s_1 , since otherwise, by transitivity, we would expect $x \prec c_2 \in \Gamma(s_1)$.

The intuitive inter-segment order derived in Example 6.22 is generalized and formally defined in the following way. Given two nodes n, m such that $\text{seg}(n) \neq \text{seg}(m)$:

$$\prec^S(n, m) \triangleq \begin{cases} \top & \text{if } C_{\succ}(n) \cap C_{\prec}(m) \neq \emptyset, \\ \top & \text{if } C_{\prec}(n) = C_{\prec}(m) \text{ and } C_{\succ}(m) \subsetneq C_{\succ}(n), \\ \perp & \text{otherwise.} \end{cases}$$

► **Remark 6.24.** Interestingly, these same definitions also work for $\text{OSC}^*[\text{TOT}]$. Though redundant, we could have defined the segments for $\text{OSC}^*[\text{TOT}]$ to include both $C_{\prec}(\cdot)$ and $C_{\succ}(\cdot)$, and use the above definition for the inter-segment order. Note that the first case is equivalent to the first case in the definition of the inter-segment order for $\text{OSC}^*[\text{TOT}]$, and the second case is void in $\text{OSC}^*[\text{TOT}]$. In a total order, if $C_{\prec}(n) = C_{\prec}(m)$ then $C_{\succ}(n) = C_{\succ}(m)$ (and in particular it cannot be that $C_{\succ}(m) \subsetneq C_{\succ}(n)$). This property can be understood intuitively, since a total order is a special case of a prefix order.



■ **Figure 6** The universal tree, a bidirectionally-infinite tree. The arrows show the transitive reduction of the order of the vertices. The $\cdot\cdot\cdot$ symbol represents an infinite *rooted* tree, isomorphic to the tree represented by the language $\langle (1|\cdots|\ell)^* \rangle$ with the standard prefix order, appended to the string of the node it is attached to.

Similarly to the construction for $\text{OSC}^*[\text{TOT}]$, the definition of the inter-segment order obtains the following lemmas:

► **Lemma 6.25.** *For any definition of the intra-segment order that produces a strict prefix order within each segment, the interpretation $\prec^{\mathcal{E}(S)}$ is a strict prefix order, i.e., $\mathcal{E}(S) \models \text{PREFIX}$.*

► **Lemma 6.26.** *Let n be a summary node and let $f, g \in \Sigma$ be function symbols such that m_n^f and m_n^g reside within different segments, then the mixed order atoms $f(x) \prec g(x)$ and $g(x) \prec f(x)$ are observed. Further, for f such that m_n^f and n reside in different segments, $x \prec f(x)$ and $f(x) \prec x$ are observed.*

Moving on to the intra-segment task, we first explain the internal shape of summary nodes, which is defined by the combination of bound formulas and order interpretation. We distinguish two kinds of summary nodes: linear nodes, any node n where $C_{\prec}(n) \neq \emptyset$, and spanning nodes, where $C_{\prec}(n) = \emptyset$. Note that when $C_{\prec}(n) \neq \emptyset$, there exists some constant c such that n has atom $x \prec c \in \Gamma(n)$, thus all explicit elements of n must be smaller than c , and therefore, by downwards totality, totally ordered.

The internal shape of a linear node n is isomorphic to the integer number line, which we express using a unary encoding: a non-negative integer z is represented by the string 1^z (i.e., concatenating the letter “1” z times), and a negative integer z is represented by the string 0^{-z} . The resulting bound formula $\mathcal{B}(n)$ is given by the regular expression $\langle 0^+ | 1^* \rangle / (x)$ (recall that any regular expression is a relation symbol in the signature of the STR theory).

For a spanning node m , a richer internal shape must be supported to ensure universal finite embeddability of any prefix order over ℓ elements, where ℓ is the number of non-ground terms in $\mathbb{T}$, thereby facilitating observance of any combination of mixed order atoms. The general shape is shown in Figure 6, a bidirectionally-infinite tree with an out-degree of ℓ . It can be seen as an extension of the infinite line of linear segments, where each point becomes a root of an infinite tree. Each (rooted) infinite tree with out-degree ℓ can be represented by the regular expressions $\langle (1|\cdots|\ell)^* \rangle$, and the entire universal tree is given by the bound formula $\mathcal{B}(m) = \langle (1|\cdots|\ell)^* | (0^+ \cdot ((2|\cdots|\ell) \cdot (1|\cdots|\ell)^*)) \rangle / (x)$.

Having defined the bound formulas, it remains to define the order interpretation in a way that will induce the desired internal shape for each summary node.

6.3.1 Defining the Intra-segment Order by Encoding the Order of the Universal Tree

Order between two vertices in the universal tree, represented by the words x_1, x_2 is given by the formula $\beta(x_1, x_2)$, defined in the sequel. Conveniently, linear summary nodes are a special case of the universal tree (where the out-degree is 1 instead of ℓ), and so the same $\beta(x_1, x_2)$ is also suitable for them.

As before, the intra-segment order is given by interleaving the summary nodes of each segment arbitrarily (using $\ll$), so for nodes n, m where $\text{seg}(n) = \text{seg}(m)$ we define

$$\prec^S(n, m) \triangleq \begin{cases} \beta(x_1, x_2) \vee x_1 = x_2 & \text{if } n \ll m, \\ \beta(x_1, x_2) & \text{otherwise.} \end{cases}$$

In order to derive the order formula $\beta(x_1, x_2)$ between two vertices in the universal tree, we identify the three different parts of the universal tree, shown in Figure 6, and then define the order within and across each part. We start by identifying the three different parts: (i) the positive tree, the rooted, infinitely increasing tree, represented by all strings over sub-alphabet $\{1, \dots, \ell\}$, (ii) the negative spine, the infinitely decreasing line, represented by the unary non-empty strings $0, 00, \dots$, and (iii) the negative branches, all trees that span out from vertices on the negative spine. These three parts are captured by the following regular expressions:

$$\begin{aligned} \text{pos}(x) &\triangleq / (1|\dots|\ell)^* / (x) \\ \text{neg-sp}(x) &\triangleq / 0^+ / (x) \\ \text{neg-br}(x) &\triangleq / 0^+ \cdot (2|\dots|\ell) \cdot (1|\dots|\ell)^* / (x) \end{aligned}$$

We now consider all possible pairs of words x_1, x_2 such that x_1 should come before x_2 according to the order within the universal tree. Immediately we can see that all vertices in the negative spine are smaller than vertices in the positive tree. Two other simple cases are when both words are in the positive tree or both are in the negative spine: in the first case they are ordered according to the prefix order, in the second according to the reverse prefix order.

Lastly, we consider the cases where the second word x_2 is on the negative branches. In these cases we need to consider the vertex on the negative spine from which the negative branch of x_2 is spanning out. We identify the root of the negative branch for a word x with the term $\text{neg-rt}(x) \triangleq \text{pref}_0(x)$, where $\text{pref}_0(x)$ is the longest prefix of x containing only “0”s. This function is definable in STR, by, e.g., the following formula $\forall x, y. y = \text{pref}_0(x) \leftrightarrow (y \sqsubset x \wedge / 0^* / (y) \wedge \neg(y \cdot 0 \sqsubset x))$. If the first word x_1 is on the negative spine and comes before the root of x_2 , then x_1 is smaller than x_2 . If x_1 is on the same negative branch as x_2 , then x_1 and x_2 are compared according to the regular prefix order. In all other cases, x_1 and x_2 are incomparable.

Putting all of the above together, $\beta(x_1, x_2)$ is defined as the following if-then-else expression:

$$\beta(x_1, x_2) \triangleq \begin{cases} \top & \text{if } \text{neg-sp}(x_1) \wedge \text{pos}(x_2) \\ x_1 \sqsubset x_2 & \text{if } \text{pos}(x_1) \wedge \text{pos}(x_2) \\ x_2 \sqsubset x_1 & \text{if } \text{neg-sp}(x_1) \wedge \text{neg-sp}(x_2) \\ \text{neg-rt}(x_2) \sqsubset x_1 & \text{if } \text{neg-sp}(x_1) \wedge \text{neg-br}(x_2) \\ x_1 \sqsubset x_2 & \text{if } \text{neg-br}(x_1) \wedge \text{neg-br}(x_2) \wedge \text{neg-rt}(x_1) = \text{neg-rt}(x_2) \\ \perp & \text{otherwise} \end{cases}$$

► **Lemma 6.27.** *The intra-segment definition of the interpretation of $\prec$ produces a strict prefix order within each segment.*

6.3.2 Defining Function Terms by Encoding Finite Paths in the Universal Tree

Finally we explain how to define function terms in a way that respects the mixed order atoms. Unlike the uniform segments used in the construction for $\text{OSC}^*[\text{TOT}]$, the segments for $\text{OSC}^*[\text{PREF}]$ are of two kinds: linear and spanning, with different internal shapes of nodes. (Note that all nodes in a segment are of the same kind; we thus refer to segments as linear or spanning accordingly.) While in $\text{OSC}^*[\text{TOT}]$, the uniformity of segments allows defining all function terms without regard to the target nodes, a subtlety arises in $\text{OSC}^*[\text{PREF}]$, since linear target nodes can only embed linear (total) orders, which means that we may fail to embed the set of *all* non-ground terms in $\mathbb{T}$, ordered according to the mixed order atoms, in linear nodes.

Therefore, when defining the function terms of a node n , instead of considering all function interpretations and mixed order atoms together, we first group functions according to the segments of their target nodes. Each such “function group” is associated with a unique segment of the corresponding target nodes, where the corresponding non-ground terms need to be embedded. We identify each group with the set of corresponding non-ground terms, and when n resides within the group’s segment, we also include x .

For example, consider a node n and two function symbols f, g where the target node of f , m_n^f , is within the same segment as n , and the target node of g , m_n^g , is in a different segment. Then f and g have distinct function groups, where the function group of f includes the terms $x, f(x)$ and is associated with a segment that includes both n and m_n^f , and the function group of g includes only $g(x)$ and is associated with a distinct segment that includes m_n^g .

For every function group $G \subseteq \mathbb{T}$ of non-ground terms, the function terms for all function symbols appearing in G are defined jointly, which is crucial to ensure the correct embedding of G , ordered according to the mixed order atoms. Importantly, the partitioning of non-ground terms $\mathbb{T}$ into groups of same-segment target nodes guarantees that the induced order can always be embedded in the corresponding segment.

► **Lemma 6.28.** *Given a node $n \in \mathcal{D}^S$ and function group of non-ground terms $G \subseteq \mathbb{T}$, if the mixed order atoms of n induce a non-linear order on G , then the segment of G is spanning.*

Generalizing the total order case, we define the function terms for the functions in each group relative to an anchor; if x is included in the group then it is the anchor, otherwise an arbitrary term $f(x)$ within the group is used. Instead of using the distance between a non-ground term $f(x)$ and the anchor term to produce function terms (as was possible in $\text{OSC}^*[\text{TOT}]$), we now think of the prefix order on a group’s non-ground terms as a forest, and consider paths from the anchor as a way to define the function terms. The paths are composed from “parent”, “child” and “sibling” steps, where different trees in the forest are connected by considering their roots as siblings. In the sequel we describe how to derive function terms according to the paths in this forest of non-ground terms.

The paths in the forest of non-ground terms are finite, and since there are at most ℓ vertices in the forest, the paths can be encoded by composing terms for the parent of a vertex, $\text{parent}(x)$, the j^{th} child of a vertex, $\text{child}_j(x)$ (for $1 \leq j \leq \ell$), and the j^{th} sibling, $\text{sibling}_j(x)$.

For $\text{parent}(x)$ we need to consider two cases: either x is on the non-positive spine, in which case its parent is obtained by appending 0, otherwise, its parent can be obtained from x by trimming the last letter. This can be expressed by the if-then-else term

$$\text{parent}(x) \triangleq \text{ITE}(x = \epsilon \vee \text{neg-sp}(x), x \cdot 0, \text{trim}_1(x)),$$

where the $\text{trim}_1(x)$ function can be defined by the formula $\forall x, y. y = \text{trim}_1(x) \leftrightarrow (y = x \cdot 0 \vee \dots \vee y = x \cdot \ell)$.

For $\text{child}_j(x)$ where $j \geq 2$ we simply append the letter j to x , and for $\text{child}_1(x)$ we need again to consider whether or not we are on the negative spine. If we are, the first child is obtained by trimming the last letter, otherwise, by appending 1:

$$\text{child}_j(x) \triangleq \begin{cases} \text{ITE}(\text{neg-sp}(x), \text{trim}_1(x), w \cdot 1) & \text{if } j = 1, \\ x \cdot j & \text{otherwise.} \end{cases}$$

We can see that there is a nice symmetry between parent and child_1 , which reflects the fact that together they form the entire (positive and negative) spine of the tree.

Finally, $\text{sibling}_j(x)$ requires additional considerations to ensure universal finite embeddability and the observance of mixed order atoms. Consider a case where we have atoms $f(x) \prec x, f(x) \prec g(x)$ but crucially no $x \prec g(x)$ or $g(x) \prec x$: we need to ensure that x and $g(x)$ will be incomparable, regardless of where x falls in the tree. For an explicit element where x represents a first child, we can take $g(x)$ to be a second child of the same parent, which will always result in an incomparable vertex. For explicit elements where x represents a second child, we swap the roles and have $g(x)$ be a first child. In general, for siblings, we can compose “parent” with an if-then-else term that replaces child_j with child_1 according to whether or not the vertex in question is a j^{th} child. For $j = 1$ this can be done by checking if the vertex is on the (positive or negative) spine. For other values of j , we check whether the last letter is j :

$$\text{is-child}_j(x) \triangleq \begin{cases} /(0|1)^*/(x) & \text{if } j = 1, \\ x = \text{trim}_1(x) \cdot j & \text{otherwise.} \end{cases}$$

Finally, we define the terms sibling_j as follows:

$$\text{sibling}_j(x) \triangleq \text{ITE}(\text{is-child}_j(x), \text{child}_1(\text{parent}(x)), \text{child}_j(\text{parent}(x))).$$

Given the above terms, for every node n and function group G , any path from the anchor to some non-ground term $f(x)$ in G can be composed to produce some $p_n^f(x)$, where the free variable x denotes the anchor. For the group where the anchor is x , we use the term $t_n^f \triangleq p_n^f(x_1)$ (x_1 representing an explicit element of the first argument of f , i.e., the explicit element of n), otherwise $t_n^f \triangleq p_n^f(\epsilon)$.

► **Lemma 6.29.** *The constructed symbolic structure is well-defined.*

► **Lemma 6.30.** *All mixed order atoms are observed.*

This completes the proof and the construction of S , thus:

► **Corollary 6.31.** $\mathcal{E}(S) \models \text{PREF} \wedge \varphi$.

► **Remark 6.32** (Complexity analysis of $\text{OSC}^*[\text{PREF}]$). As previously explained, the construction of a symbolic model for $\text{OSC}^*[\text{PREF}]$ provides a bound on the size of the model and identifies finite sets of bound formulas, function terms and relation formulas, which may be computed from the formula. Note however that the decision procedure induced by the symbolic model property reduces satisfiability of formulas in $\text{OSC}^*[\text{PREF}]$ to validity of formulas in STR, which is non-elementary. Thus the exact size bounds of symbolic structures are non-important for the complexity analysis.

6.4 Decidability of $\text{OSC}^*[\text{TOT} \wedge \text{PROSUCC}]$ and $\text{OSC}^*[\text{TOT} \wedge \text{REGPRED}]$

One notable property of the symbolic structures constructed in Sections 6.2 and 6.3 is that some elements have neither successors nor predecessors. In particular, both in the case of a total order and of a prefix order, regular nodes – i.e., interpretations of constants – may be “disconnected”: since summary nodes stretch from negative infinity to positive infinity, regular nodes adjacent to summary nodes have no immediate neighbors.

In this section and the next we consider two variants of total and prefix orders where the order relation $<$ is axiomatized such that there must exist a successor (in one variant) or a predecessor (in the other). To make these orders viable, we also restrict all functions to progressive or respectively regressive, so as to ensure mixed order atoms would not force an infinitely decreasing (respectively increasing) summary nodes. The induced internal shapes of summary nodes are then unidirectionally infinite, which ensures the existence of successors or predecessors.

The adapted construction for $\text{OSC}^*[\text{TOT} \wedge \text{PROSUCC}]$ (respectively $\text{OSC}^*[\text{TOT} \wedge \text{REGPRED}]$) reuses most of the construction of Section 6.2. However, one key adjustment is needed. Namely, instead of having $\top$ as the bound formulas for summary nodes, we define them to $\mathcal{B}(n) = x \geq 0$ in the $\text{TOT} \wedge \text{PROSUCC}$ variant, and $\mathcal{B}(n) = x \leq 0$ in the $\text{TOT} \wedge \text{REGPRED}$ variant. This matches the desired internal shapes, and ensures that all elements have immediate neighbors, thus satisfying the successor/predecessor axioms. Importantly, well-definedness is preserved: in $\text{OSC}^*[\text{TOT} \wedge \text{PROSUCC}]$, progressivity ensures that for any node n , the mixed order atoms will induce an order where the anchor x is minimal among all non-ground terms, and as a result, all function terms t_n^f will be of the form $x_1 + k_n^f$ for some non-negative integer k_n^f . Similarly in $\text{OSC}^*[\text{TOT} \wedge \text{REGPRED}]$, regressivity ensures all function terms to be of the form $x_1 + k_n^f$ for some non-positive integer k_n^f .

► **Lemma 6.33.** *The constructions for $\text{OSC}^*[\text{TOT} \wedge \text{PROSUCC}]$ and $\text{OSC}^*[\text{TOT} \wedge \text{REGPRED}]$ are well-defined and correct.*

6.5 Decidability of $\text{OSC}^*[\text{PREF} \wedge \text{PROSUCC}]$ and $\text{OSC}^*[\text{PREF} \wedge \text{REGPRED}]$

Similarly to the total order variants, the internal shapes of summary nodes become unidirectionally infinite for the variants of $\text{OSC}^*[\text{PREF}]$, and we adapt the bound formulas and function terms to reflect this.

For the successor variant $\text{OSC}^*[\text{PREF} \wedge \text{PROSUCC}]$, the bound formulas for linear summary nodes are $/1^*/(x)$, representing a positive line, and for spanning summary nodes they are $/(11 \cdots 1\ell)^*/(x)$, representing a positive rooted tree. When defining the function terms, in order to preserve well-definedness, we avoid $\text{parent}(x)$ and $\text{sibling}_j(x)$ terms. To that end, we always include the term x as the anchor for all function groups and construct paths from it. Due to progressivity, x will always be the minimum among the non-ground terms, and thus paths will only use $\text{child}_j(x)$ terms.

For the predecessor variant $\text{OSC}^*[\text{PREF} \wedge \text{REGPRED}]$, the predecessor axiom requires all summary nodes to have the internal shape of a negative line, using the bound formula $/0^*/(x)$ for all summary nodes. Similarly to $\text{PREF} \wedge \text{PROSUCC}$, when defining function terms we always include the term x as the anchor for all function groups, but as a mirror-image of $\text{PREF} \wedge \text{PROSUCC}$, here x will always be the maximum among the non-ground terms, and all paths will only use $\text{parent}(x)$ terms. In fact, in this case we can use the theory of LIA to express bound formulas and function terms. But notably, unlike in the symbolic

structures constructed for $\text{OSC}^*[\text{TOT} \wedge \text{REGPRED}]$, the overall order interpretation in the symbolic structures for $\text{OSC}^*[\text{PREF} \wedge \text{REGPRED}]$ may still be non-linear, due to the order between constants.

► **Lemma 6.34.** *The constructions for $\text{OSC}^*[\text{PREF} \wedge \text{PROSUCC}]$ and $\text{OSC}^*[\text{PREF} \wedge \text{REGPRED}]$ are well-defined and correct.*

7 Implementation

We implemented a tool for exploring theory-generic symbolic structures by building on the open-source **FEST** [15] Python library, which originally supported symbolic structures over LIA. We extended **FEST** to a generic framework, enabling model-checking of symbolic structures over different decidable theories. As a concrete instantiation beyond LIA, we added the STR theory of strings, whose decision procedure is implemented via a translation to Weak Monadic Second-Order Logic of One Successor (WS1S). The latest version of **FEST** is available at [14].

To support this translation, we developed Python bindings to the **MONA** [23] solver for WS1S and integrated them with the generic **FEST** infrastructure. Implementing the decision procedure for STR by translation to WS1S allows us to efficiently support common string operations via a specialized, optimized translation. For example, we support trim_1 and pref_0 , as well as specific regular expressions like $/0^*/$ (all of which are used in the symbolic structures constructed in Section 6.3).

Our implementation serves as a proof of concept, demonstrating the feasibility of our approach and providing a preliminary platform for experimenting with non-LIA symbolic structures.

8 Conclusion and Future Work

We have presented a generalization of symbolic structures to arbitrary base theories with a standard model, and have shown how this formalism can be used to establish decidability results for fragments of FOL that do not enjoy a finite model property. Our main technical contribution is a symbolic model property for several fragments in the Ordered Self-Cycle family, obtained via a generic construction of symbolic models from arbitrary models. This yields new decidability results, including for the Prefix-Ordered Self-Cycle fragment, and extends prior work that was limited to total orders and symbolic structures over LIA.

The framework developed in this paper opens several directions for further inquiry. On the logical side, it would be interesting to identify additional decidable fragments using symbolic model properties, in particular fragments based on other kinds of order relations, such as partial orders or lattice-like structures.

On the practical side, while our focus has been on foundational results, symbolic structures are intended as a means for counter-model generation in verification. An important direction for future work is to enhance the toy implementation of symbolic structures into a fully-fledged tool, and evaluate its effectiveness on realistic verification benchmarks. Understanding how symbolic counter-models can be presented to users and integrated into existing verification workflows remains an open and relevant challenge.

References

- 1 Aharon Abadi, Alexander Rabinovich, and Mooly Sagiv. Decidable fragments of many-sorted logic. *J. Symb. Comput.*, 45(2):153–172, 2010. doi:10.1016/j.jsc.2009.03.003.
- 2 Wilhelm Ackermann. Über die erfüllbarkeit gewisser zählausdrücke. *Mathematische Annalen*, 100(1):638–649, 1928. doi:10.1007/BF01448869.

- 3 Leo Bachmair and Harald Ganzinger. Resolution theorem proving. In *Handbook of Automated Reasoning*, pages 19–99. Elsevier and MIT Press, Amsterdam, 2001. doi:10.1016/B978-044450813-3/50004-7.
- 4 Vince Bárány, Erich Grädel, and Sasha Rubin. Automata-based presentations of infinite structures. In *Finite and Algorithmic Model Theory*, volume 379 of *London Mathematical Society Lecture Note Series*, pages 1–76. Cambridge University Press, 2011. doi:10.1017/CB09780511974960.002.
- 5 Vince Bárány, Balder ten Cate, and Luc Segoufin. Guarded negation. In Luca Aceto, Monika Henzinger, and Jiri Sgall, editors, *Automata, Languages and Programming - 38th International Colloquium, ICALP 2011, Zurich, Switzerland, July 4-8, 2011, Proceedings, Part II*, volume 6756 of *Lecture Notes in Computer Science*, pages 356–367, Berlin, Germany, 2011. Springer. doi:10.1007/978-3-642-22012-8_28.
- 6 Michael Benedikt, Leonid Libkin, Thomas Schwentick, and Luc Segoufin. Definable relations and first-order query languages over strings. *J. ACM*, 50(5):694–751, 2003. doi:10.1145/876638.876642.
- 7 Achim Blumensath and Erich Grädel. Finite presentations of infinite structures: Automata and interpretations. *Theory Comput. Syst.*, 37(6):641–674, 2004. doi:10.1007/S00224-004-1133-Y.
- 8 Achim Blumensath and Erich Grädel. Finite presentations of infinite structures: Automata and interpretations. *Theory Comput. Syst.*, 37(6):641–674, 2004. doi:10.1007/s00224-004-1133-y.
- 9 Egon Börger, Erich Grädel, and Yuri Gurevich. *The Classical Decision Problem*. Perspectives in Mathematical Logic. Springer, Berlin, 1997.
- 10 J Richard Büchi. On a decision method in restricted second order arithmetic. In *The Collected Works of J. Richard Büchi*, pages 425–435. Springer New York, New York, NY, 1990.
- 11 J. Richard Büchi and Lawrence H. Landweber. Definability in the monadic second-order theory of successor. *J. Symb. Log.*, 34(2):166–170, 1969. doi:10.2307/2271090.
- 12 David C Cooper. Theorem proving in arithmetic without multiplication. *Machine intelligence*, 7(91-99):300, 1972. doi:10.1007/10930755_5.
- 13 Daniel Danielski and Emanuel Kieronski. Finite satisfiability of unary negation fragment with transitivity. In *MFCS*, volume 138 of *LIPICs*, pages 17:1–17:15, Dagstuhl, Germany, 2019. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPICs.MFCS.2019.17.
- 14 Neta Elad. Infinite models tools (FEST). <https://www.cs.tau.ac.il/~netaelad/#infinite-models-fest>. Accessed: 2024-05-06.
- 15 Neta Elad, Oded Padon, and Sharon Shoham. An infinite needle in a finite haystack: Finding infinite counter-models in deductive verification. *Proc. ACM Program. Lang.*, 8(POPL):970–1000, 2024. doi:10.1145/3632875.
- 16 Neta Elad and Sharon Shoham. Decidability results for fragments of first-order logic via a symbolic model property, 2026. arXiv:2605.05840.
- 17 Kurt Gödel. Ein spezialfall des entscheidungsproblems der theoretischen logik. *Ergebnisse eines mathematischen Kolloquiums*, 2:27–28, 1932.
- 18 Erich Grädel, Wied Pakusa, Svenja Schalthöfer, and Lukasz Kaiser. Characterising choiceless polynomial time with first-order interpretations. In *30th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2015, Kyoto, Japan, July 6-10, 2015*, pages 677–688. IEEE Computer Society, 2015. doi:10.1109/LICS.2015.68.
- 19 Yuri Gurevich. The decision problem for standard classes. *The Journal of Symbolic Logic*, 41(2):460–464, 1976. doi:10.1017/S0022481200051513.
- 20 Shachar Itzhaky, Anindya Banerjee, Neil Immerman, Aleksandar Nanovski, and Mooly Sagiv. Effectively-propositional reasoning about reachability in linked data structures. In *CAV*, volume 8044 of *Lecture Notes in Computer Science*, pages 756–772, Berlin, Germany, 2013. Springer. doi:10.1007/978-3-642-39799-8_53.

- 21 László Kalmár. Über die erfüllbarkeit derjenigen zähl ausdrücke, welche in der normalform zwei benachbarte allzeichen enthalten. *Mathematische Annalen*, 108(1):466–484, 1933.
- 22 Bakhadyr Khoussainov and Anil Nerode. Automatic presentations of structures. In Daniel Leivant, editor, *Logical and Computational Complexity. Selected Papers. Logic and Computational Complexity, International Workshop LCC '94, Indianapolis, Indiana, USA, 13-16 October 1994*, volume 960 of *Lecture Notes in Computer Science*, pages 367–392. Springer, 1994. doi:10.1007/3-540-60178-3_93.
- 23 Nils Klarlund and Anders Møller. *MONA Version 1.4 User Manual*. BRICS, Department of Computer Science, University of Aarhus, January 2001. Notes Series NS-01-1. Available from <http://www.brics.dk/mona/>. URL: <http://www.brics.dk/mona/>.
- 24 Harry R. Lewis. Complexity results for classes of quantificational formulas. *J. Comput. Syst. Sci.*, 21(3):317–353, 1980. doi:10.1016/0022-0000(80)90027-6.
- 25 Martin Löb. Decidability of the monadic predicate calculus with unary function symbols. *Journal of Symbolic Logic*, 32(4):563, 1967.
- 26 Giuliano Losa and Mike Dodds. On the formal verification of the stellar consensus protocol. In Bruno Bernardo and Diego Marmosoler, editors, *2nd Workshop on Formal Methods for Blockchains, FMBC@CAV 2020, Los Angeles, California, USA (Virtual Conference), July 20-21, 2020*, volume 84 of *OASICS*, pages 9:1–9:9, Dagstuhl, Germany, 2020. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/OASICS.FMBC.2020.9.
- 27 Raz Lotan and Sharon Shoham. Implicit rankings for verifying liveness properties in first-order logic. In Arie Gurfinkel and Marijn Heule, editors, *Tools and Algorithms for the Construction and Analysis of Systems - 31st International Conference, TACAS 2025, Held as Part of the International Joint Conferences on Theory and Practice of Software, ETAPS 2025, Hamilton, ON, Canada, May 3-8, 2025, Proceedings, Part I*, volume 15696 of *Lecture Notes in Computer Science*, pages 375–395, Berlin, Germany, 2025. Springer. doi:10.1007/978-3-031-90643-5_20.
- 28 Kenneth L. McMillan and Oded Padon. Deductive verification in decidable fragments with ivy. In Andreas Podelski, editor, *Static Analysis - 25th International Symposium, SAS 2018, Freiburg, Germany, August 29-31, 2018, Proceedings*, volume 11002 of *Lecture Notes in Computer Science*, pages 43–55, Berlin, Germany, 2018. Springer. doi:10.1007/978-3-319-99725-4_4.
- 29 Michael Mortimer. On languages with two variables. *Mathematical Logic Quarterly*, 21(1):135–140, 1975. doi:10.1002/malq.19750210118.
- 30 Adithya Murali, Lucas Peña, Ranjit Jhala, and P. Madhusudan. Complete first-order reasoning for properties of functional programs. *Proc. ACM Program. Lang.*, 7(OOPSLA2):1063–1092, 2023. doi:10.1145/3622835.
- 31 Oded Padon, Giuliano Losa, Mooly Sagiv, and Sharon Shoham. Paxos made EPR: decidable reasoning about distributed protocols. *Proc. ACM Program. Lang.*, 1(OOPSLA):108:1–108:31, 2017. doi:10.1145/3140568.
- 32 Nicolas Peltier. Building infinite models for equational clause sets: Constructing non-ambiguous formulae. *Log. J. IGPL*, 11(1):97–129, 2003. doi:10.1093/jigpal/11.1.97.
- 33 Michael O Rabin. Decidability of second-order theories and automata on infinite trees. *Transactions of the american Mathematical Society*, 141:1–35, 1969. doi:10.2307/1995086.
- 34 Frank P Ramsey. On a problem of formal logic. *Proceedures of London Mathematical Society*, 30:264–285, 1930. doi:10.1007/978-0-8176-4842-8_1.
- 35 Saharon Shelah. Decidability of a portion of the predicate calculus. *Israel Journal of Mathematics*, 28(1):32–44, 1977. doi:10.1007/BF02759780.
- 36 Orr Tamir, Marcelo Taube, Kenneth L. McMillan, Sharon Shoham, Jon Howell, Guy Gueta, and Mooly Sagiv. Counterexample driven quantifier instantiations with applications to distributed protocols. *Proc. ACM Program. Lang.*, 7(OOPSLA2):1878–1904, 2023. doi:10.1145/3622864.

- 37 Marcelo Taube, Giuliano Losa, Kenneth L. McMillan, Oded Padon, Mooly Sagiv, Sharon Shoham, James R. Wilcox, and Doug Woos. Modularity for decidability of deductive verification with applications to distributed systems. In Jeffrey S. Foster and Dan Grossman, editors, *Proceedings of the 39th ACM SIGPLAN Conference on Programming Language Design and Implementation, PLDI 2018, Philadelphia, PA, USA, June 18-22, 2018*, pages 662–677, New York, NY, 2018. ACM. doi:10.1145/3192366.3192414.
- 38 Marco Voigt. *Decidable fragments of first-order logic and of first-order linear arithmetic with uninterpreted predicates*. PhD thesis, Saarland University, Saarbrücken, Germany, 2019. URL: <https://publikationen.sulb.uni-saarland.de/handle/20.500.11880/27767>.