Toward a More Complete Alloy ⋆,⋆⋆

Timothy Nelson[1], Daniel J. Dougherty[1], Kathi Fisler[1], and Shriram Krishnamurthi[2]

[1] Worcester Polytechnic Institute
[2] Brown University

Abstract. Many model-finding tools, such as Alloy, charge users with providing bounds on the sizes of models. It would be preferable to automatically compute sufficient upper-bounds whenever possible. The Bernays-Schönfinkel-Ramsey fragment of first-order logic can relieve users of this burden in some cases: its sentences are satisfiable iff they are satisfied in a finite model, whose size is computable from the input problem.
Researchers have observed, however, that the class of sentences for which such a theorem holds is richer in a many-sorted framework—which Alloy inhabits—than in the one-sorted case. This paper studies this phenomenon in the general setting of order-sorted logic supporting overloading and empty sorts. We establish a syntactic condition generalizing the Bernays-Schönfinkel-Ramsey form that ensures the Finite Model Property. We give a linear-time algorithm for deciding this condition and a polynomial-time algorithm for computing the bound on model sizes. As a consequence, model-finding is a complete decision procedure for sentences in this class. Our work has been incorporated into Margrave, a tool for policy analysis, and applies in real-world situations.

1 Introduction

The undecidability of first-order logic poses a challenge to using Alloy for verification: analysis performed under bounds may not be complete. While incompleteness is unavoidable for some classes of formulas, there are also classes for which analysis is complete under domains of finite size. Alloy asks users to specify domain-size bounds, but does not help users determine whether their bounds suffice for completeness. Ideally, tools such as Alloy would provide such feedback or, better still, compute sufficient bounds automatically when possible.

Sufficient-bounds results are long-established for classical first-order logic. Alloy's logic, however, is different in ways that impact computing bounds. Alloy signatures yield first-order logic with *sorts*: the class of many-sorted first-order logic formulas with sufficient bounds properly includes that for the unsorted case. Existing results on sufficient bounds for many-sorted logic, however, make assumptions that are not valid for many Alloy specifications: Alloy allows sorts to be empty, and also allows sorts to overlap. These features, which are critical for modeling realistic systems, require an

⋆ This research is partially supported by the NSF.

⋆⋆ An expanded version of this paper, with complete proofs, is available at http://tinyurl.com/osepl-tr-pdf

extended theory of bounds-computation. This paper presents the theory and algorithms for computing sufficient bounds for a substantial class of Alloy formulas.

We actively use our results within our Margrave tool (www.margrave-tool.org) for analyzing policies (such as access-control, firewall, and routing policies). One of our standard policy examples—from a deployed conference-paper manager—requires the results in this paper. Margrave uses the presented algorithms to compute how many papers are required for complete reasoning (typically a single-digit number). In other examples, Margrave computes sufficient bounds on *some* sorts (even when others cannot be bounded). This can help a user decide how to allocate the computational resources of model finding. Margrave is built upon Kodkod [24], the backend model-finder for the Alloy Analyzer. For Alloy users, we provide an implementation online (sortedtermcount.appspot.com) that takes a formula σ in Alloy notation, checks whether σ lies in our class of decidable formulas, and computes sufficient sizes for whichever Alloy signatures we can bound.

The bulk of this paper presents our results and algorithms in mathematical detail. For a more casual reader, Section 2 gives an intuitive view of our results, including the nuances that make empty sorts and overlapping sorts more difficult to handle.

2 Overview of Results

The Bernays-Schönfinkel-Ramsey class, sometimes called "Effectively Propositional Logic" (EPL), comprises the set of first-order sentences of the form

$$\exists x_1 \ldots \exists x_n \forall y_1 \ldots \forall y_m \,.\, \varphi$$

where φ is quantifier-free and has no function symbols. The satisfiability problem for this class is decidable: Bernays and Schönfinkel [2] and Ramsey [22] showed that such a sentence has a model if and only if it has a model of size bounded by n plus the number of constants in φ. When such a *finite model property* holds, satisfiability-testing reduces to exhaustive search for a model within bounded domains. Furthermore, the search need only consider models whose elements are constants. In effect, satisfiability for these formulas reduces to propositional satisfiability.

The EPL results assume that all variables quantify over the same domain. Alloy uses a *sorted* first-order logic, in which values come from several domains (Alloy signatures) and each variable is associated with a particular domain. Sorts provide additional information that model finders and theorem provers can exploit in the search for models [12, 14, 16, 19]. More strikingly, the class of sentences with the finite model property is richer in a sorted framework [1, 8, 11]. The following simple example illustrates the interplay between sorts and bounds for completeness. Consider the class of unsorted sentences of the form

$$\forall y_1 \exists x \forall y_2 \,.\, \varphi.$$

Satisfiability is undecidable for this prefix class [3]. In contrast, this sorted verion

$$\sigma \equiv \forall y_1^A \exists x^B \forall y_2^A \,.\, \varphi \tag{1}$$

is better behaved. Suppose that φ contains constants, say n_A constants of sort A and n_B of sort B, but no function symbols. If we were to postulate that sort A is a subsort of sort

B, then if σ has any models at all then it has a model whose size at sort A is bounded by n_A and whose size at sort B is bounded by $(2n_A + n_B)$. On the other hand, if our signature declared that B were a subsort of A, then some σ would only have infinite models. In considering subsort relationships, this example illustrates *order-sorted* logic, in which there is a partial order on the sorts rather than an assumption that all sorts are disjoint. We give a formal treatment of this example below, as Example 15.

Alloy's use of order-sorted logic, as well as its allowance of empty sorts, demand new methods for computing sufficient bounds. To illustrate why, we first consider a standard approach to establishing the finite-model property. Let σ be a sentence in unsorted first-order logic.

1. By Skolemization, there is a universal sentence σ_{sk} equi-satisfiable with σ. The language of σ_{sk} is richer than that of σ, since constants and function symbols have been introduced on behalf of existential quantifiers of σ.
2. Any potential model $\mathcal{M}$ for σ_{sk} has a *Skolem hull* [4] consisting of the interpretation in the model of the ground terms of the language. The set of ground terms is called the *Herbrand universe*. The Skolem hull forms a submodel of $\mathcal{M}$ in which every element is named by a term in the language.
3. A fundamental classical theorem is that the truth of universal sentences is preserved under submodel. Thus, if the signature of σ_{sk} has only finitely many terms, that is, if the Herbrand universe is finite, then σ has the finite-model property.

When the language has only a single sort, the only way to guarantee that the Herbrand universe is finite is to have no function symbols (other than constants). In that setting, the sentences whose Skolemization produces no function symbols comprise the EPL class. The many-sorted setting is more lenient. Consider for example a sentence σ whose Skolemization leads to a language with simply a constant a of sort A, a function f of sort $A \to B$. Then the only ground terms that can be constructed are a and $f(a)$ (terms such as $f(f(a))$ are not well-sorted). This suggests—correctly—that a richer classes of finite-model results are available.

But there are technical obstacles to generalizing the above argument. In particular,

- When empty sorts are allowed, the Skolem form of σ is not equi-satisfiable with σ. For example the sentence $(\forall y^A.y = y) \vee (\exists x^B.x \neq x)$ is true in models where the sort B is empty. Skolemization, with a new constant b of sort B, yields the sentence $(\forall y^A.y = y) \vee (b \neq b)$ which is is unsatisfiable. Section 5 addresses this issue formally.
- When sorts are not assumed to be disjoint (the order-sorted setting), not every element in the Skolem hull of a model is named by a term. Indeed the Skolem hull of $\mathcal{M}$ can be infinite even when a finite submodel of $\mathcal{M}$ *does* exist. Example 9 in Section 5 illustrates this case.

Contributions This paper adapts the approaches in the standard argument to accommodate ordered sorts and empty sorts. In doing so, it enables automatic bounds computation for additional Alloy formulas through the following contributions:

- We identify (Definition 11) a syntactically-determined class of sentences extending EPL, comprising *Order-Sorted Effectively Propositional Logic (OS-EPL)*, for which the Finite Model Property holds (Theorem 10, Section 6).

– We present a linear-time algorithm (Corollary 16) for membership in OS-EPL. We present a cubic-time algorithm (Theorem 17) for computing an upper bound on the size of models required for testing satisfiability. It is interesting to note that the bound itself can be exponential in the size of the sentence (Section 7), even though it can be computed in polynomial time.

We view identification of the OS-EPL class as a contribution to a taxonomy of decidability classes in order-sorted logic. In the presence of possibly-empty sorts, sentences do not always have equivalent prenex-normal forms, so we cannot attempt a decidability classification in terms of quantifier prefix as in [3]. As Section 6 shows, our decidability criterion is based entirely on the signature of the Skolemization of the given formula. This signature can be viewed as a generalization of the idea of quantifier prefix, as it implicitly records the pattern of nesting between universal and existential quantification.

2.1 A Sample Application

The PLT Scheme application Continue [15] automates many conference-management tasks. Margrave has been helpful in developing and analyzing Continue; here we hint at some of the ways that the algorithms in this paper have improved some of these analyses.

The access-control policy of a conference can be represented as a first-order theory, over a language representing facts about the world and access-control decisions such as *permit* and *deny*. A user will query the policy to verify or falsify properties of the system; Margrave's mode of interaction is to generate models, or "scenarios" for situations being explored by the user.

For example, a certain policy rule might say "The conference administrator can advance the conference out of the `bidding` phase if every reviewer has bid on some paper." This rule gives rise to the sentence

$$permit(s, advancePhase, conference) \longleftarrow Admin(s) \wedge (phase = Bidding) \wedge \forall u^{User} \exists p^{Paper} \,.\, bidOn(u, p)).$$

The set of such policy rules, together with assumed facts about the application domain, comprise a background theory for analysis. Now suppose one wants to verify the property: "The conference chair can modify user passwords." It suffices to determine that there are no models of *the negation of* the formula

$$\forall r^{User} \,.\, permit(chair, modifyPassword, r)$$

together with the background theory. The question, of course, is determining an upper bound on the scope of the search. The fact that the formula being explored by the user is purely existential is of little help by itself, since the entire policy theory is part of the satisfiability query. Besides rules such as the permit rule quoted above, the language also includes such function symbols as $paperPhase : Paper \rightarrow PaperPhase$ and $decision : Paper \rightarrow Decision$.

In the absence of sensitivity to sorts this theory would not submit to a finite-model discipline. But in fact, for the Continue theory and the associated query above Margrave automatically computes sufficient bounds:

```
Counts for finitary, populated sorts:
Conference:7  PaperPhase:12 Object:14 ConferencePhase:10
Action:16 User:9 Resource:6 univ:59 Paper:6
```

Without the support of the finite-model algorithms of this paper the user would—à la Alloy—have to instruct the tool to restrict attention to a finite search space that was presumably arrived at in an *ad-hoc* manner.

3 Related Work

The decidability of the satisfiability problem for the $\exists\forall$ class in pure logic is a classical result of Bernays and Schönfinkel [2] in the absence of equality, extended by Ramsey [22] to allow equality. The problem is known to be EXPTIME-complete [17]. An example of the usefulness of multiple sorts in pure logic is Feferman's work [7] on interpolation theorems. Goguen and Meseguer did seminal work [10] on order-sorted algebra; order sorted predicate logic was first considered by Oberschelp [21].

Harrison was one of the first to observe that many-sortedness can not only yield efficiencies in deduction but can also support new decidability results. In unpublished notes [11] he presents some examples of this phenomenon, and suggests searching for typed analogs of classical decidability classes, as we have done here.

Order-sorted signatures (without relation symbols) can be viewed as tree automata [6,23], so the question of whether the set of closed terms is finite can be answered using standard automata techniques. We believe that the algorithm in this paper for counting terms is new.

Fontaine and Gribomont [8], working in "flat" many-sorted logic (*i.e.*, without subsorting) prove that if there are no functions having result sort A and σ is a universal sentence then σ has a model if and only if it has a model in which the size of A is bounded by the number of constants of sort A. This result is used to eliminate quantifiers in certain verification conditions. This theorem has application even when not all sorts are finite and can be used in a setting where some functions and predicates are interpreted.

Claessen and Sorensson [5] have integrated a *sort inference* algorithm into the Paradox model-finder that deduces sort information for unsorted problems and, under certain conditions, can bound the size of domains for certain sorts and improve the performance of the instantiation procedure. Order-sorting is not used, and there are restrictions on the use of equality.

Momtahan [18] computes a refutationally-complete upper bound on the size of a single sort (as a function of the user-provided bounds on the other sorts) for a fragment of the Alloy kernel language. The conditions defining this fragment are not directly comparable to ours, but in some respects constrain the sentences rather severely. For example existential quantification in the scope of more than one universal quantifier is usually not allowed.

Abadi *et al.* [1] identify, as we do, a decidable fragment of sorted logic that is decidable by virtue of having a finite Herbrand universe. Although they target Alloy in their examples they work in a many-sorted logic without subsorts or empty sorts; their condition for decidability is the existence of a "stratification" of the function vocabulary; they do not provide algorithms for checking the stratification condition or computing size bounds on the models.

Ge and de Moura [9] present a powerful method for deciding satisfiability modulo theories with an instantiation-based theorem prover. Given a universal (Skolemized) sentence σ they construct a system of set constraints whose least solution constitutes a set of ground terms sufficient for instantiation; satisfiability is thus decidable for the set of sentences for which this solution-set is finite (in the many-sorted setting this subsumes the Abadi *et al.* class). They do not treat empty sorts nor subsorting. They can treat certain sentences that fall outside our OS-EPL class; detection of whether a given sentence falls into their decidable class seems to require solving the associated set-constraints, as compared to our linear-time algorithm. Generally speaking they do detailed fine-grained analysis of individual sentences; we have focused on an easily recognized class of sentences.

The problem of efficiently deciding satisfiability in the EPL class is an active area of research. Our work is complementary to these efforts in that it identifies an extended class of sentences to which contemporary techniques can hopefully be applied.

A preliminary version of this work was presented at the workshop on Synthesis, Verification, and Analysis of Rich Models (SVARM), July 20-21, 2010.

4 Background: Order-Sorted Predicate Logic and Term Models

We begin by formalizing several foundational concepts that underlie the high-level argument in points 1–3 of Section 2. Naturally, we define signatures and models for order-sorted first-order logic. Finite-model properties derive from arguments that every model of a sentence has a truth-preserving submodel with only finitely-many elements. Two concepts are key to such an argument: homomorphisms between models (and hence submodels), and a *term model*, which is a particular model over the ground-terms of a sentence. Establishing that the term model is a submodel (under homomorphism) of every model of a sentence is essential to proving completeness under finite bounds; a theorem in this section captures this requirement.

The definitions and results in this section are either directly from Goguen and Meseguer's work [10] or they are the obvious extensions required to handle relations as well as functions.

Notation We use $\langle\rangle$ for the empty sequence. If $(\mathcal{S},\leq)$ is an ordering we extend $\leq$ to words in $\mathcal{S}^*$ and then to products, pointwise.

Signatures An *order-sorted signature* is a triple $\mathcal{L} = (\mathcal{S},\leq,\Sigma)$ where $(\mathcal{S},\leq)$ is a finite poset of sorts and Σ is an indexed family of symbols, the vocabulary, comprising

- $\{\Sigma_w \mid w \in \mathcal{S}^*\}$, an $\mathcal{S}^*$-sorted family of *relation symbols,* and
- $\{\Sigma_{w,A} \mid w \in \mathcal{S}^* , A \in \mathcal{S}\}$, an $(\mathcal{S}^* \times \mathcal{S})$-sorted family of *function symbols*.

We assume that the Σ_w and $\Sigma_{w,A}$ are pairwise disjoint.

We stress that an order-sorted signature is not the same as an Alloy signature. Such a signature denotes a language of discourse: available sorts, functions, etc. along with an ordering on the sorts. In this way, an Alloy signature is a sort or a predicate within an overall order-sorted signature.

Formalizing relations and functions through words—rather than through tuples of sorts—simplifies certain definitions and eases capturing overloaded function symbols (as [10] does). Our work assumes function symbols are not overloaded (through the disjointness condition on the $\Sigma_{w,A}$); this is consistent with Alloy. Most of the results of the paper generalize to handle overloading, including our finite model theorem (Theorem 14). The one exception is our term-counting algorithm (Theorem 17), which relies on the lack of overloading to compute precise bounds; with overloading, our algorithm only promises upper bounds on the sort-sizes.

When $R \in \Sigma_w$ we say that w is the *arity* of R. When $f \in \Sigma_{w,A}$ we say that w is the *arity* of f and A is the *result sort* of f. If $\mathcal{L} = (\mathcal{S}, \leq, \Sigma)$ and $\mathcal{L}' = (\mathcal{S}, \leq, \Sigma')$ are such that for each w and A, $\Sigma_w \subseteq \Sigma'_w$ and $\Sigma_{w,A} \subseteq \Sigma'_{w,A}$ we say that $\mathcal{L}'$ is an *expansion* of $\mathcal{L}$, and that $\mathcal{L}$ is a *reduct* of $\mathcal{L}'$.

Following standard usage, a function symbol $a \in \Sigma_{\langle\rangle,A}$, taking no arguments, is referred to as a "constant" of sort A, and in concrete syntax we write simply a instead of $a()$.

The *connected components* of an ordering $(\mathcal{S}, \leq)$ are the equivalence classes for the equivalence relation generated by $\leq$. A signature $\mathcal{L} = (\mathcal{S}, \leq, \Sigma)$ is *coherent* if each pair of sorts in the same connected component has an upper bound. Henceforth we assume that our signatures are coherent. See [10] for an extended discussion of the importance of coherence. Note: in [10] the notion of coherence also requires that signatures be *regular*, a technical condition that is trivially satisfied in the absence of overloading.

The set of *formulas* is defined inductively by closing the set of atomic formulas under the propositional operators $\wedge$, $\vee$, and $\neg$ and the quantifiers $\exists$ and $\forall$. We will indicate quantification over a sorted variable $x \in X_A$ by $\exists x^A$ or $\forall x^A$ (where X_A is the set of variables of sort A). The notions of free and bound variable are standard; let $FV(\varphi)$ denote the set of free variables of formula φ. A *sentence* is a formula with no free variable occurrences.

Models Fix a signature $\mathcal{L} = (\mathcal{S}, \leq, \Sigma)$. An $\mathcal{L}$-*model* $\mathcal{M}$ comprises

- an $\mathcal{S}$-sorted family $\{\mathcal{M}_A \mid A \in S\}$ of sets, the *universe* of $\mathcal{M}$, such that $A \leq A'$ implies $\mathcal{M}_A \subseteq \mathcal{M}_{A'}$,
- for each $R \in \Sigma_w$ a relation $R^{\mathcal{M}_w} \subseteq \mathcal{M}_w$
- for each $f \in \Sigma_{w,A}$ a function $f^{\mathcal{M}_{w,A}} : \mathcal{M}_w \to \mathcal{M}_A$

As described in the introduction, the first step in investigating the finite model property for a sentence is Skolemization, the process of eliminating existential quantifiers in favor of function symbols. As a consequence we need to be attentive to the ways that the language over which our models are defined can shift. If $\mathcal{M}$ is a model for $\mathcal{L} = (\mathcal{S}, \leq, \Sigma)$ and $\mathcal{L}'$ is an expansion of $\mathcal{L}$ then an *expansion* of $\mathcal{M}$ to $\mathcal{L}'$ is a model of $\mathcal{L}'$ with the same universe as $\mathcal{M}$ which agrees with $\mathcal{M}$ on the symbols in Σ.

An *environment* η over a model $\mathcal{M}$ is an $\mathcal{S}$-indexed family of finite functions $\{\eta_A : X_A \to \mathcal{M}_A \mid A \in \mathcal{S}\}$ such that $\eta_A = (\eta_{A'})\restriction_{X_A}$ (the restriction to X_A) whenever $A \leq A'$. An environment η can be extended to terms in the usual way. When $\mathcal{M}$ is a model, φ a formula, and η an environment such that $FV(\varphi) \subseteq dom(\eta)$ the relation $\mathcal{M} \models_\eta \varphi$ is defined by the usual induction.

Homomorphism A *homomorphism* $h : \mathcal{M} \to \mathcal{N}$ between models $\mathcal{M}$ and $\mathcal{N}$ is an $\mathcal{S}$-sorted family of functions $\{h_A : \mathcal{M}_A \to \mathcal{N}_A \mid A \in \mathcal{S}\}$ satisfying the following conditions (suppressing sort information for readability).

$$A \leq A' \text{ implies } h_A = (h_{A'}) \restriction_{\mathcal{M}_A}$$
$$h(f^{\mathcal{M}}(a_1,\ldots,a_n)) = f^{\mathcal{N}}(h(a_1),\ldots,h(a_n)), \text{ and}$$
$$R^{\mathcal{M}}(h(a_1,\ldots,a_n)) \text{ implies } R^{\mathcal{N}}(h(a_1),\ldots,h(a_n))$$

The Term Model When the set of relation symbols in $\mathcal{L}$ is empty then the set of ground terms forms the universe of a model for $\mathcal{L}$, the *term algebra* [10]. We may view this as a model for an arbitrary order-sorted signature, as follows.

Fix $\mathcal{L} = (\mathcal{S}, \leq, \Sigma)$. The family $\{\mathcal{T}^{\mathcal{L}}_A \mid A \in \mathcal{S}\}$ of *ground terms* over $\mathcal{L}$ is the $\subseteq$-least family such that (i) $\mathcal{T}^{\mathcal{L}}_A \subseteq \mathcal{T}^{\mathcal{L}}_{A'}$ whenever $A \leq A'$ and (ii) if $f \in \Sigma_{w,A}$ with $w = A_1 \ldots A_n$ and for each i, $t_i \in \mathcal{T}^{\mathcal{L}}_{A_i}$ then $f(t_1,\ldots,t_n) \in \mathcal{T}^{\mathcal{L}}_A$. The ground terms determine a model $\mathcal{T}^{\mathcal{L}}$ of $\mathcal{L}$, the *term model*, by taking the interpretation of each $f \in \Sigma_{\langle A_1 \ldots A_n \rangle, A}$ to be the function taking each tuple $(t_1,\ldots,t_n) \in (\mathcal{T}^{\mathcal{L}}_{A_1} \times \cdots \times \mathcal{T}^{\mathcal{L}}_{A_n})$ to the term $f(t_1,\ldots,t_n)$, and taking the interpretation of each relation symbol to be the empty relation.

Theorem 1. *Suppose $\mathcal{L} = (\mathcal{S}, \leq, \Sigma)$ is a signature such that Σ has no relation symbols. Then for any model $\mathcal{M}$ of $\mathcal{L}$ there is a unique homomorphism from $\mathcal{T}^{\mathcal{L}}$ to $\mathcal{M}$ (i.e., $\mathcal{T}^{\mathcal{L}}$ is initial).*

Proof. Initiality of $\mathcal{T}^{\mathcal{L}}$ in the category of *algebras* was shown by Goguen and Meseguer [10]. Now, given an $\mathcal{L}$-model $\mathcal{M}$, we let $\mathcal{M}'$ be the reduct of $\mathcal{M}$ to the language $\mathcal{L}'$ obtained by removing the relation symbols. So $\mathcal{M}'$ is a $\mathcal{L}'$-algebra so that Goguen and Meseguer's theorem applies. But the unique algebra homomorphism from $\mathcal{T}^{\mathcal{L}}$ to $\mathcal{M}'$ is itself a $\mathcal{L}$-homomorphism from $\mathcal{T}^{\mathcal{L}}$ to $\mathcal{M}$, simply because each $\mathcal{T}^{\mathcal{L}}$-relation is empty, and the result follows. ///

5 Skolemization

A formula is in *negation-normal* form if the negation sign is applied only to atomic formulas. As for standard one-sorted logic, DeMorgan's laws for pushing negations below $\wedge$ and $\vee$, and the equivalences between $\neg\exists x^A \alpha$ and $\forall x^A \neg\alpha$ all hold, even in the presence of empty sorts. So every formula is logically equivalent to a formula in negation normal form. But the fact that models can have empty sorts changes the rules for how quantifiers may be moved within a formula. In particular the passage between $((\exists x^A \alpha) \vee \beta)$ and $\exists x^A(\alpha \vee \beta)$ (when x is not free in β) does not hold if A can be empty (and of course the dual equivalence involving $\forall$ fails as well) and so we cannot in

general percolate quantifiers to the front of a formula. So we cannot restrict our attention to formulas in prenex normal form, but we will always pass to negation-normal form.

Definition 2 (Skolemization). Let φ be a negation-normal form formula over signature $\mathcal{L} = (\mathcal{S}, \leq, \Sigma)$; the result of a *Skolemization-step* of φ is any formula φ' that can be obtained as follows. If $\exists x^A.\psi(x^A, x_1^{A_1}, \ldots, x_n^{A_n})$ is a subformula occurrence of φ that is not in the scope of an existential quantifier, let f be a function symbol not in Σ, and let φ' be the result of replacing the occurrence of $\exists x^A.\psi(x, x_1, \ldots, x_n)$ by $\psi(f(x_1, \ldots, x_n), x_1, \ldots, x_n)$. Note that φ' is a formula in an expanded signature obtained by adding f to $\Sigma_{\langle A_1,\ldots,A_n\rangle,A}$.

A *Skolemization* of a formula φ is a sentence with no existential quantifiers, obtained from φ by a sequence of such steps.

The following lemma is straightforward.

Lemma 3. *For any σ we have $\sigma_{sk} \models \sigma$.*

In contrast to the classical case we do not have the fact that "σ satisfiable implies σ_{sk} satisfiable." That holds in one-sorted logic because we can always expand a model of σ to properly interpret the Skolem functions and make σ_{sk} true, but this expansion is not always possible in the presence of empty sorts.

Example 4. Let σ be $(\exists x^A \,.\, (x = x) \vee \exists y^B \,.\, (y = y)\,) \wedge (\forall z^A \,.\, (z \neq z))$. Then σ is satisfiable but its Skolemization $((a = a) \vee (b = b)) \wedge (\forall z^A \,.\, (z \neq z))$ is not.

The phenomenon in Example 4 is essentially the only thing that can go wrong: models can be expanded to interpret Skolem functions if we do not existentially quantify over empty sorts. This points the way to recovering a weak version of the classical equisatisfiability result which will be good enough for our present purposes.

Lemma 5. *If σ is satisfiable then there exists a formula $\sigma_\perp$ such that (i) $\sigma_\perp \models \sigma$ and (ii) $\sigma_{\perp sk}$ is satisfiable.*

Proof. Suppose $\mathcal{M} \models \sigma$. The sentence $\sigma_\perp$ is obtained by replacing $\exists x^A \,.\, \alpha$ by $\perp$ precisely when $\mathcal{M}_A = \emptyset$. It is straightforward to see that $\sigma_\perp \models \sigma$. Since in $\sigma_\perp$ there is no existential quantification over sorts empty in $\mathcal{M}$ one can show that there is an expansion $\mathcal{M}^*$ of $\mathcal{M}$ to the signature of $\sigma_{\perp sk}$ such that $\mathcal{M}^* \models \sigma_{\perp sk}$. ///

6 A Finite Model Theorem for Order-Sorted Logic

Model $\mathcal{M}$ is a *submodel* of model $\mathcal{N}$ if (i) for each sort A, $\mathcal{M}_A \subseteq \mathcal{N}_A$ and (ii) each $f^{\mathcal{M}}$ and $R^{\mathcal{M}}$ are obtained as the restrictions of $f^{\mathcal{N}}$ and $R^{\mathcal{N}}$ to $\mathcal{M}$. Note that we use "submodel" in this strong sense rather than just requiring each $R^{\mathcal{M}}$ to be a subset of $R^{\mathcal{N}}$ (as is done by some authors).

If $X = \{X_A \mid A \in \mathcal{S}\}$ is a family of sets with $X_A \subseteq \mathcal{M}_A$ for each $A \in \mathcal{S}$ then we say that X is closed under a function $g : \mathcal{M}_{A_1} \times \cdots \times \mathcal{M}_{A_n} \to \mathcal{M}_A$ if whenever $(a_1, \ldots, a_n) \in X_{A_1} \times \cdots \times X_{A_n}$ we have $g(a_1, \ldots, a_n) \in X_A$. Note that this is a stronger claim than saying that the single set $\bigcup X$ is closed under g.

Lemma 6. *Let $h : \mathcal{P} \to \mathcal{M}$ be a homomorphism between models of $\mathcal{L} = (\mathcal{S}, \leq, \Sigma)$. There is a unique submodel of $\mathcal{M}$ with universe $\{h_A(\mathcal{P}_A) \mid A \in \mathcal{S}\}$.*

Proof. It is easy to check that the family $\{h_A(\mathcal{P}_A) \mid A \in \mathcal{S}\}$ is closed under the interpretations in $\mathcal{M}$ of the function symbols in Σ. So if we define the interpretations of the relation symbols in Σ to be the restriction of the interpretations in $\mathcal{M}$ the result is a submodel. Since there is no choice in the interpretations of the symbols in Σ once the universe $\{h_A(\mathcal{P}_A) \mid A \in \mathcal{S}\}$ is determined, uniqueness follows. ///

Next we establish the fundamental fact about preservation of universal sentences under submodel.

Theorem 7. *Let σ be a sentence that is existential-free and in negation-normal form and let $\mathcal{M}'$ be a submodel of $\mathcal{M}$. If $\mathcal{M} \models \sigma$ then $\mathcal{M}' \models \sigma$.*

The proof is a straightforward induction.

Definition 8 (The *kernel* of a model). Let $\mathcal{M}$ be a model for the signature $\mathcal{L} = (\mathcal{S}, \leq, \Sigma)$. Let h be the unique homomorphism from $\mathcal{T}^{\mathcal{L}}$ to $\mathcal{M}$ (*c.f.* Theorem 1). The image of h is a submodel of $\mathcal{M}$ by Lemma 6; this is the *kernel* of $\mathcal{M}$.

The crucially important fact for us is that for the kernel $\mathcal{K}$ of $\mathcal{M}$ we have, for each sort A, the cardinality of $\mathcal{K}_A$ is bounded by the cardinality of $\mathcal{T}^{\mathcal{L}}_A$, simply because $\mathcal{K}_A$ is the image of $\mathcal{T}^{\mathcal{L}}_A$ under h.

The kernel and the Skolem hull Recall the classical treatment of Skolemization (see *e.g.*, [4]): given a model $\mathcal{M}$, let $\mathcal{M}^*$ be a model interpreting the Skolem functions that satisfies the Skolem theory (the sentences saying that the Skolem functions witness the truth of the associated existential formula). Then given a subset X of the universe of $\mathcal{M}$, the Skolem hull $\mathcal{H}_{\mathcal{M}}(X)$ is the smallest subset of the universe containing X and closed under the functions and constants of the enriched language; this determines an elementary submodel $\mathcal{H}_{\mathcal{M}}(X)$ of $\mathcal{M}^*$. In particular $\mathcal{H}_{\mathcal{M}}(\emptyset)$ can be viewed as a "minimal" submodel of $\mathcal{M}$.

But in the order-sorted setting, *the kernel of a model is not in general the same as the Skolem hull.* The latter notion, although perfectly sensible in order-sorted logic, does not play the same role of "minimal" submodel as it does in the one-sorted setting. Indeed it is possible for the kernel of a model to be finite while the Skolem hull is infinite.

Example 9. Consider $\mathcal{L} = (\{A, B\}, \emptyset, \Sigma)$ with $a \in \Sigma_{\langle\rangle, A}$ and $f \in \Sigma_{B,B}$ the only vocabulary symbols. Let $\mathcal{M}$ have $\mathcal{M}_A = \{b_0 = a^{\mathcal{M}}\}$, $\mathcal{M}_B = \{b_0, b_1, b_2, \ldots\}$, and $f^{\mathcal{M}}$ map b_i to b_{i+1}. Then the Skolem hull $\mathcal{H}(\emptyset)$ of $\mathcal{M}$ is $\mathcal{M}$ itself. Yet the kernel $\mathcal{K}$ of $\mathcal{M}$ is the model of size 1 with $\mathcal{K}_A = \{b_0\}$, $\mathcal{K}_B = \emptyset$, $f^{\mathcal{K}} = \emptyset$.

Here we present our main theorem.

Theorem 10. *Let σ be an $\mathcal{L}$-sentence whose Skolemization σ_{sk} has signature $\mathcal{L}^*$. Then σ is satisfiable if and only if σ has a model $\mathcal{H}$ such that for each sort A, the cardinality of $\mathcal{H}_A$ is no greater than the cardinality of $\mathcal{T}^{\mathcal{L}^*}_A$.*

Proof. For the non-trivial direction, suppose σ is satisfiable. By Lemma 5 there is an approximation $\sigma_\perp$ of σ such that $\sigma_{\perp sk}$ is satisfiable. Let $\mathcal{L}^{**}$ be the signature for $\sigma_{\perp sk}$; note that $\mathcal{L}^{**}$ is a reduct of $\mathcal{L}^*$ and the sentence $(\sigma_\perp)_{sk}$ is existential-free.

Let $\mathcal{M}$ be a model of $(\sigma_\perp)_{sk}$, and let $\mathcal{H}$ be the kernel of $\mathcal{M}$. Since $(\sigma_\perp)_{sk}$ is existential-free, $\mathcal{H} \models (\sigma_\perp)_{sk}$. Since $\mathcal{H}$ is a kernel we have that for each sort A, the cardinality of $\mathcal{H}_A$ is no greater than the cardinality of $\mathcal{T}_A^{\mathcal{L}^{**}}$, and thus no greater than the cardinality of $\mathcal{T}_A^{\mathcal{L}^*}$. Since $(\sigma_\perp)_{sk} \models \sigma_\perp$ and $\sigma_\perp \models \sigma$, the model $\mathcal{H}$ is the desired model of σ. ///

Finally we can define precisely the key notion of the paper.

Definition 11. *Order-Sorted Effectively Propositional Logic (OS-EPL)* is the class of sentences σ such that the signature of the Skolemization of σ has a finite term model.

The next section shows how to decide whether a sentence is in OS-EPL and if so, to compute the sizes of the sorts in the term model. Taken together with Theorem 10, this establishes a decision procedure for satisfiability of OS-EPL sentences.

7 Algorithms

Let $\mathcal{L} = (\mathcal{S}, \leq, \Sigma)$ be a signature. We say that sort A is *finitary* in $\mathcal{L}$ if $\mathcal{T}_A^{\mathcal{L}}$ is finite.

Our membership algorithm reduces the problem of counting terms to one of asking whether a given context-free grammar yields only a finite number of strings; well-known algorithms solve the latter problem. [13, 23]. Intuitively, the grammar captures the ground terms that can be generated from the signature.

Definition 12. Given a signature $\mathcal{L} = (\mathcal{S}, \Sigma, \leq)$ with multiple sorts, we define a grammar $G_\mathcal{L}$ as follows. The set of nonterminals is $\mathcal{S} \cup \{A_0\}$, where A_0 is a fresh symbol not in $\mathcal{S}$, the set of terminals is $\bigcup\{\Sigma_{w,S} \mid (w,s) \in \mathcal{S}^* \times \mathcal{S}\}$, and the set of productions comprises:

$$\begin{array}{rl} A_0 \to A & \text{for each } A \in \mathcal{S} \\ B \to f A_1 \ldots A_n & \text{whenever } f \in \Sigma_{\langle A_1 \ldots A_n \rangle, B} \\ B \to A & \text{whenever } A \leq B \end{array}$$

A non-terminal X in a context-free grammar G is said to be *useful* if there exists a derivation $A_0 \Rightarrow^* \alpha X \beta \Rightarrow^* u$ where u is a string of terminals, otherwise X is *useless.* If A is a useful non-terminal and u is a string of terminals we say that A *generates* u if there is a derivation $A \Longrightarrow^* u$.

Lemma 13. *Let A be a sort of $\mathcal{L}$ and let u be a string of terminals over $\bigcup\{\Sigma_{w,S} \mid (w,s) \in \mathcal{S}^* \times \mathcal{S}\}$. Then u is a term in $\mathcal{T}_A^{\mathcal{L}}$ if and only if there is a derivation $A \Rightarrow^* u$ in $G_\mathcal{L}$. A sort A is inhabited by a ground term if and only if A is useful in the grammar $G_\mathcal{L}$. When A is useful as a sort in $L(G_\mathcal{L})$, the set $\mathcal{T}_A^{\mathcal{L}}$ is finite if and only if A generates only finitely many terms in $L(G_\mathcal{L})$. In particular the set $\mathcal{T}^{\mathcal{L}}$ is finite if and only if $L(G_\mathcal{L})$ is finite.*

Proof. The first claim is easy to check: it holds essentially by the construction of $G_{\mathcal{L}}$. The second claim follows from the first and the facts that the u in question are strings of terminals of $G_{\mathcal{L}}$ and we have $A_0 \Rightarrow A$ for each $A \in \mathcal{S}$. ///

Theorem 14. *There is an algorithm that, given an order-sorted signature $\mathcal{L}$, determines (uniformly) for each sort A, whether $\mathcal{T}_A^{\mathcal{L}}$ is finite. The algorithm runs in time linear in the total size of $\mathcal{L}$.*

Proof. By Lemma 13, $\mathcal{T}_A^{\mathcal{L}}$ is finite if and only if A generates only finitely many terms in in $L(G_{\mathcal{L}})$. There is a well-known algorithm for testing whether a non-terminal in a context-free grammar generates infinitely many terminal strings [13]. After eliminating useless symbols from the grammar $G_{\mathcal{L}}$, form the graph whose nodes are the inhabited sorts, with an edge from B to A if and only if there is a production in $G_{\mathcal{L}}$ of the form $B \to \alpha\, A\, \beta$, that is, if and only if the set $\Sigma_{\langle A_1 \ldots A \ldots A_n \rangle, B}$ is non-empty or if $A \leq B$. Then a non-terminal A generates infinitely many terminal strings if and only if there is a path from A to a cycle. Since the size of $G_{\mathcal{L}}$ is linear in the size of $\mathcal{L}$, the overall complexity of our algorithm is linear in $\mathcal{L}$. ///

Example 15. Return to Equation 1 from Section 2. Over the signature $\mathcal{L}$ with two sorts A and B, with $A \leq B$, consider the sentence

$$\forall y_1^A \exists x^B \forall y_2^A \,.\, \varphi \tag{2}$$

where φ has no function symbols. After Skolemizing we have the signature with $b \in \Sigma_{\langle\rangle, B}$ and $f \in \Sigma_{A,B}$ in addition to those constants in the original signature. The corresponding grammar has productions $A_0 \to A$, $A_0 \to B$, $B \to b$, $B \to f\, A$ and $B \to A$, in addition to productions corresponding to the constants appearing in the original φ. The resulting graph has edges from the node A_0 to A and to B, and an edge from B to A (the latter for two reasons, due to the grammar production $B \to f\, A$ and due to the production $B \to A$). This graph is acyclic so we conclude that this class of sentences has the finite model property.

On the other hand, if we were to postulate that $B \leq A$ (instead of $A \leq B$) then we cannot deduce the finite model property. Our grammar would have the production $A \to B$ in addition to $B \to A$ and the resulting graph would have a cycle.

Corollary 16. *Membership in OS-EPL is decidable in linear time.*

Proof. Let σ be given, over signature $\mathcal{L}$. We can compute the skolemization σ_{sk} of σ in linear time, and extract the signature $\mathcal{L}^*$ of σ_{sk}. The size of this signature is clearly linear in σ, so by Theorem 14, we can decide whether all sorts of $\mathcal{L}^*$ are finitary in time linear in σ. ///

Note that in the worst case, Σ may induce a number of terms exponential in its size. Thus we would like to avoid actually generating the terms, and merely count them if we can do so in polynomial time.

Theorem 17. *There is an algorithm that, given a signature $\mathcal{L}$, computes, in time cubic in the size of $\mathcal{L}$, the size of $\mathcal{T}_A^{\mathcal{L}}$ for each finitary sort A (returning "∞" for the non-finitary sorts).*

Space does not permit a full presentation of the algorithm: see [20] for the details. Intuitively, if a sort is finitary, its terms can be of height no greater than the number of functions in Σ. So we construct a table containing the number of terms of each height of each sort, starting with constants and then applying functions. The only complication is that when counting the ways to create a new term of height h using function f, we need to make certain that each has at least one subterm of height *exactly* $h-1$. The algorithm is implemented using dynamic programming, and the cubic bound is straightforward to establish.

Summarizing, we have the following sound and complete procedure for testing satisfiability of OS-EPL sentences. Given sentence σ, compute its Skolemization σ_{sk}; let $\mathcal{L}^*$ be the signature of σ_{sk}. If the term model $\mathcal{T}^{\mathcal{L}^*}$ is finite then we know that if σ is satisfiable then σ has a model whose universe has cardinalities as given in Theorem 10. Since these bounds are computable we can effectively decide satisfiability for such sentences.

Remark 18. The results of the algorithm in Theorem 17 can be useful even if not all sorts are finitary. Fontaine and Gribomont [8] have implemented an instantiation-based algorithm that takes advantage of the information that certain sorts are guaranteed to have finitely many ground terms. Their algorithm does not do a sophisticated test for this condition, in fact it succeeds only if there are no non-constant terms in the sort in question. Our algorithm here is simple yet will allow their methods to be applicable to a wider class of sentences.

8 Future Work

This work suggests two major lines of further inquiry. The first is the exploration of algorithms for working with OS-EPL sentences that are efficient in practice. A natural approach is to leverage insights from existing tools for model-finding and theorem-proving that are currently optimized for the traditional EPL class. The other, more theoretical, direction is to pursue a program of classifying fragments of order-sorted logic according to decidability. Abadi *et al.* [1] suggest a taxonomy based on quantifier prefix patterns but, as pointed out in the introduction, prenex-normal form is not available when sorts are allowed to be empty. We propose that a combinatorial analysis of the signature of Skolemizations of sentences is the proper generalization of the analysis of classical quantifier prefix classes.

References

1. Abadi, A., Rabinovich, A., Sagiv, M.: Decidable fragments of many-sorted logic. Journal of Symbolic Computation 45(2), 153 – 172 (2010)
2. Bernays, P., Schönfinkel, M.: Zum entscheidungsproblem der mathematischen Logik. Mathematische Annalen 99, 342–372 (1928)
3. Börger, E., Grädel, E., Gurevich, Y.: The Classical Decision Problem. Perspectives in Mathematical Logic, Springer (1997)
4. Chang, C.C., Keisler, J.: Model Theory. No. 73 in Studies in Logic and the Foundations of Mathematics, North-Holland (1973), third edition, 1990

5. Claessen, K., Sorensson, N.: New techniques that improve MACE-style finite model finding. In: Proceedings of the CADE-19 Workshop on Model Computation (2003)
6. Common, H., Dauchet, M., Gilleron, R., Jacquemard, F., Lugiez, D., Tison, S., Tommasi, M.: Tree automata techniques and applications, draft book; available electronically at http://www.grappa.univ-lille3.fr/tata
7. Feferman, S.: Many-Sorted Interpolation Theorems and Applications. In: Proceedings of the Tarski Symposium, AMS Proc. Symp. in Pure Math. vol. 25, pp. 205–223 (1974)
8. Fontaine, P., Gribomont, E.P.: Decidability of invariant validation for parameterized systems. In: Garavel, H., Hatcliff, J. (eds.) Tools and Algorithms for Construction and Analysis of Systems (TACAS). Lecture Notes in Computer Science, vol. 2619, pp. 97–112. Springer-Verlag (2003)
9. Ge, Y., Moura, L.: Complete instantiation for quantified formulas in satisfiabiliby modulo theories. In: CAV '09: Proceedings of the 21st International Conference on Computer Aided Verification. pp. 306–320. Springer-Verlag, Berlin, Heidelberg (2009)
10. Goguen, J.A., Meseguer, J.: Order-Sorted Algebra I: Equational Deduction for Multiple Inheritance, Overloading, Exceptions and Partial Operations. Theor. Comput. Sci. 105(2), 217–273 (1992)
11. Harrison, J.: Exploiting sorts in expansion-based proof procedures (Unpublished manuscript), http://www.cl.cam.ac.uk/~jrh13/papers/manysorted.pdf
12. Hooker, J., Rago, G., Chandru, V., Shrivastava, A.: Partial instantiation methods for inference in first-order logic. Journal of Automated Reasoning 28(4), 371–396 (2002)
13. Hopcroft, J.E., Motwani, R., Ullman, J.D.: Introduction to Automata Theory, Languages, and Computation. Addison-Wesley, Reading, Massachusetts, third edn. (2006)
14. Jereslow, R.G.: Computation-oriented reductions of predicate to propositional logic. Decision Support Systems 4, 183–197 (1988)
15. Krishnamurthi, S., Hopkins, P., McCarthy, J., Graunke, P., Pettyjohn, G., Felleisen, M.: Implementation and use of the PLT Scheme web server. Higher-Order and Symbolic Computation 20(4), 431–460 (2007)
16. Lahiri, S., Seshia, S.: The UCLID decision procedure. In: 16th International Conference, Computer-Aided Verification. pp. 475–478. Springer (2004)
17. Lewis, H.: Complexity results for classes of quantificational formulas. J. Comp. and Sys. Sci. 21(3), 317–353 (1980)
18. Momtahan, L.: Towards a small model theorem for data independent systems in Alloy. Electronic Notes in Theoretical Computer Science 128(6), 37 – 52 (2005), proceedings of the Fouth International Workshop on Automated Verification of Critical Systems (AVoCS 2004)
19. de Moura, L.M., Bjørner, N.: Deciding Effectively Propositional Logic Using DPLL and Substitution Sets. In: Armando, A., Baumgartner, P., Dowek, G. (eds.) IJCAR. Lecture Notes in Computer Science, vol. 5195, pp. 410–425. Springer (2008)
20. Nelson, T., Dougherty, D.J., Fisler, K., Krishnamurthi, S.: On the finite model property in order-sorted logic. Tech. rep., Worcester Polytechnic Institute (2010), http://tinyurl.com/osepl-tr-pdf
21. Oberschelp, A.: Order sorted predicate logic. In: Workshop on Sorts and Types in Artificial Intelligence. pp. 1–17. Springer (1989)
22. Ramsey, F.P.: On a problem in formal logic. Proceedings of the London Mathematical Society 30, 264–286 (1930)
23. Schmidt-Schauß, M.: Computational Aspects of an Order-Sorted Logic with Term Declarations, Lecture Notes in Computer Science, vol. 395. Springer (1989)
24. Torlak, E., Jackson, D.: Kodkod: A relational model finder. In: Conference on Tools and Algorithms for the Construction and Analysis of Systems. vol. 4424, p. 632 (2007)