On the Finite Model Property in Order-Sorted Logic

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

[1] Worcester Polytechnic Institute
[2] Brown University

Abstract. The Schoenfinkel-Bernays-Ramsey class is a fragment of first-order logic with the Finite Model Property: a sentence in this class is satisfiable if and only if it is satisfied in a finite model. Since an upper bound on the size of such a model is computable from the sentence, the satisfiability problem for this family is decidable. Sentences in this form arise naturally in a variety of application areas, and several popular reasoning tools explicitly target this class.
Others have observed that the class of sentences for which such a finite model theorem holds is richer in a many-sorted framework than in the one-sorted case. This paper makes a systematic study of this phenomenon in the general setting of order-sorted logic supporting overloading and empty sorts. We establish a syntactic condition generalizing the Schoenfinkel-Bernays-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 algorithms have been incorporated into Margrave, a tool for analysis of access-control and firewall policies, and are available in a standalone application suitable for analyzing input to the Alloy model finder.

1 Introduction

The Schoenfinkel-Bernays-Ramsey class, or sometimes, "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: Schoenfinkel and Bernays [2] and Ramsey [26] 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 can be reduced to exhaustive search. More important to applications is the fact that model-finders for EPL sentences can restrict their search to models whose elements are constants; similarly, instantiation-based theorem provers can restrict attention to instantiation by constants.

Model-finders and theorem-provers can benefit from the additional information that a many-sorted framework can provide [7, 17, 20, 21]. More strikingly, the class of sentences supporting finite model theorems is richer in a many-sorted framework than in the one-sorted case [1,11,15]. In this paper we make a systematic study of this latter phenomenon, in a very general *order-sorted* framework. Our main results are as follows.

- We identify (Definition 29) a syntactically-determined class of sentences extending EPL, comprising *Order-Sorted Effectively Propositional Logic (OS-EPL)*, for which the Finite Model Property holds (Theorem 28).

– We present a linear-time algorithm (Corollary 35) for membership in OS-EPL and a cubic-time algorithm (Theorem 36) for computing an upper bound on the size of models required for testing satisfiability; the bound itself might be exponential in the size of the sentence.
– We make some—perhaps surprising—observations about the foundational and algorithmic consequences of the use of sorts as predicates (as done in several tools), including the failure of Herbrand's Theorem. These issues are addressed by a suitable translation into the core language (Section 7).

Our work is motivated by the needs of our Margrave tool (http://www.margrave-tool.org) for the verification and analysis of policies. Margrave reduces policies for access-control, routing, address translation, and other concerns to sorted first-order logic. The results of this paper, which have been implemented in Margrave, enable us to calculate domain bounds and offer complete answers to many important questions, such as change-impact queries [10] posed by policy authors.

We have also created a Web interface (http://sortedtermcount.appspot.com) for readers to experiment with the algorithms in this paper. The tool accepts input in the notation used by Alloy [19], a popular system for the analysis of system requirements and designs. Our tool can be used as preprocessor for the Alloy Analyzer model-finder. Given an assertion σ about an Alloy specification the tool can first check to see if the input is in OS-EPL and if so, it computes bounds on the sizes of the sorts (Alloy's "sigs") with a guarantee that model-finding up to these bounds is a *complete* method for assertions-checking.

The following simple example gives the flavor of our results. Consider the class of unsorted sentences of the form

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

This prefix class has an undecidable satisfiability problem. But the following sorted version

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

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. Suppose in addition that sort A is a subsort of sort B. Under these conditions σ is in OS-EPL. Indeed we can show that 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 $(2n_A + n_B)$. So we have a finite model theorem and satisfiability for this class of formulas is decidable. On the other hand if we were to require instead that B is a subsort of A, then the resulting sentence is not in OS-EPL; some such σ have only infinite models. These assertions are all consequences of our main theorem, Theorem 28 below.

Outline An instructive way to present the technical challenges and contributions of this paper is to consider a standard approach to showing that a class of sentences has the finite-model property. Given a sentence σ in unsorted first-order logic we might reason as follows.

1. By Skolemization, there is a universal sentence σ_{sk} equi-satisfiable with σ;
2. Any potential model $\mathcal{M}$ for σ_{sk} has a *Skolem hull*, obtained by closing the interpretation of the constants by the interpretation of the functions [4]. This makes a submodel of $\mathcal{M}$ in which every element is named by a term in the language.
3. An easy theorem of unsorted logic is that the truth of universal sentences is preserved under submodel.

Thus, if the signature of σ_{sk} has only finitely many terms, we can conclude our finite model theorem. This last part of the argument succeeds in the one-sorted case only for formulas in the EPL class, but the starting point of this research is the observation that the many-sorted framework offers more opportunities for a finite Herbrand universe. However, the following facts complicate matters.

1. When empty sorts are allowed, the Skolem form of σ is not equi-satisfiable with σ (Section 4).
2. When sorts are not assumed to be disjoint—in particular, in 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 27). (This phenomenon also cause difficulties in giving a model-theoretic proof of Herbrand's Theorem for order-sorted logic: see Section 5.4).
3. When sort names are allowed to be used as predicates, as they are in many tools, preservation of universal sentences under submodel fails (Section 7).

Our development addresses each of these difficulties.

We view the identification of the OS-EPL class as a contribution to a taxonomy of decidability classes in order-sorted logic in the same spirit as for classical unsorted 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 5 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 Related Work

The decidability of the satisfiability problem for the $\exists\forall$ class in pure logic is a classical result of Schönfinkel and Bernays [2] in the absence of equality, extended by Ramsey [26] to allow equality. The problem is known to be EXPTIME-complete [22].

The monograph by Börger, Grädel, and Gurevich [3] is an comprehensive treatment of decidability for classical unsorted logic. Herbrand's Theorem [16] is the basis for automated deduction and propositional methods. Enderton [8] is a textbook treatment of many-sorted logic classically considered. In this setting sorts are assumed non-empty and pairwise disjoint. Strictly speaking, what is assumed is that there is no cross-sort equality; this implies that any model is elementarily equivalent to one with disjoint sorts. An example of the usefulness of multiple sorts in pure logic is Feferman's work [9] on interpolation theorems. Goguen and Meseguer did seminal work [14] on order-sorted algebra; Goguen and Diaconescu [13] present a good survey of the field through 1994. Order sorted predicate logic was first considered by Arnold Oberschelp [24]; Walther [28] explores many-sorted unification in the context of orderings on sorts, Weibel [29] extends this work to the order-sorted case.

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 [15] he presents some examples of this phenomenon, and suggests searching for typed analogs of classical decidability classes, as we have done here.

Fontaine and Gribomont [11], 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. As observed in Remark 37, the algorithm in Theorem 36 can be used to apply their techniques to a wider class of formulas than they address.

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 [23] defines a fragment of the Alloy kernel language and proves a result computing a refutationally-complete upper bound on the size of a single sort (as a function of the user-provided bounds on the other sorts). 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 are 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 [12] 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. Jereslow [20] described a "partial instantiation" approach to first-order theorem proving in the EPL fragment, constructing a sequence of propositional instantiations instead of working with the full set of possibilities from the outset. Work by Hooker *et al.* [17] builds directly on Jereslow's approach (see also many references there). Recent alternatives approaches include [7] and [21]. De Moura and Bjørner [6] have developed the SMT constraint solver Z3. SMT enriches propositional satisfiability by adding equality reasoning, arithmetic, bit-vectors, arrays, and some quantification. Z3 is used in software verification and analysis applications. De Moura and Bjørner [7]; and Piskac and de Moura and Bjørner [25], introduce a DPLL-based decision procedure for the EPL class; this has been implemented as part of Z3. Our work is complementary to these efforts in that it identifies an extended class of sentences to which contemporary techniques can hopefully be applied.

3 Preliminaries: Order-Sorted Predicate Logic

The definitions and results in this section are either directly from Goguen and Meseguer's work [14] or they are the obvious extensions to the setting in which relations as well as function are considered.

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. The *connected components* of $(\mathcal{S},\leq)$ are the equivalence classes for the equivalence relation generated by $\leq$. Suppose $w = A_1 \ldots A_n$ is a word in $\mathcal{S}^*$ and suppose that for each i we have defined a set X_{A_i}; then the notation X_w refers to $X_{A_1} \times \cdots \times X_{A_n}$.

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*, satisfying the monotonicity condition

$$f \in \Sigma_{w_1,A_1} \cap \Sigma_{w_2,A_2} \quad \text{and} \quad w_1 \leq w_2 \quad \text{imply} \quad A_1 \leq A_2$$

When $R \in \Sigma_w$ we say that w is the *arity* of R. When $f \in \Sigma_{w,s}$ we say that w is the *arity* of f and A is the *result sort* of f.

A signature is *regular* if whenever $f \in \Sigma_{w_1,A_1}$ and $w_0 \leq w_1$ then there is least $(w,S) \in (\mathcal{S}^* \times \mathcal{S})$ such that $w_0 \leq w$ and $f \in \Sigma_{w,s}$. A signature $\mathcal{L} = (\mathcal{S},\leq,\Sigma)$ is *coherent* if it is regular and $(\mathcal{S},\leq)$ is locally filtered (*i.e.*, each pair of sorts in the same connected component has an upper bound).

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 $\mathcal{L}'$.

This is a very general notion of signature, allowing symbols to be overloaded, *i.e.*, declared in more than one sort. Following standard usage, a symbol $a \in \Sigma_{\langle\rangle,A}$ is referred to as a "constant" of sort A, and in concrete syntax we write simply a instead of $a()$. Note that the monotonicity assumption means that constants cannot be overloaded.

On the other hand, note that sort names are *not* eligible to be used as predicates, in contrast to certain treatments of many-sorted logic. There are very good reasons for this: see Section 7.

Example 1. The following signature is filtered, but *not* regular. $\mathcal{S} = \{A,B,C,D\}$, with $A \leq B \leq D$ and $A \leq C \leq D$. $\Sigma_{\langle\rangle,A} = a$, $\Sigma_{B,B} = f$, $\Sigma_{C,C} = f$. Not regular since there is no least (w,S) with $A \leq w$ and $f \in \Sigma_{w,S}$.

See Example 3, Example 6, and Theorem 5 below for motivation for the restriction to coherent signatures.

The restriction to finite vocabulary is reasonable given the fact that our main concern in this paper is finite model theorems. In our setting, when $\mathcal{S}$ is finite, local filtering is equivalent to the requirement that each component have a maximum element.

It may be that one would like to specify that certain pairs of sorts are to be disjoint (*e.g.* Alloy, with "extends"). For simplicity we do not consider such a mechanism in our notion of signature, since we can get the same effect by explicitly writing disjointness sentences and conjoining hem with any sentences under consideration (as long as they are in the same connected component!). For future reference we note that such sentences involve no existential quantifiers.

Models Fix an OS 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

$$A \leq A' \text{ implies } \mathcal{M}_A \subseteq \mathcal{M}_{A'},$$

- for each $R \in \Sigma_w$ a relation $R^{\mathcal{M}_w} \subseteq \mathcal{M}_w$, such that

$$R \in \Sigma_{w_1} \cap \Sigma_{w_2} \quad \text{and} \quad w_1 \leq w_2 \quad \text{imply} \quad R^{\mathcal{M}_{w_1}} = R^{\mathcal{M}_{w_2}} \cap \mathcal{M}_{w_1}$$

- for each $f \in \Sigma_{w,A}$ a function $f^{\mathcal{M}_{w,A}} : \mathcal{M}_w \to \mathcal{M}_A$, such that

$$f \in \Sigma_{w_1,A_1} \cap \Sigma_{w_2,A_2} \quad \text{and} \quad w_1 \leq w_2 \quad \text{imply} \quad f^{\mathcal{M}_{w_1,A_1}} \text{ is } f^{\mathcal{M}_{w_2,A_2}} \text{ restricted to } \mathcal{M}_{w_1}$$

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.

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).

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

Motivating (local) filtering Two cautionary examples from [14].

Example 2. $A \leq B$, $A < C$, $f \in \Sigma_{B,B} \cap \Sigma_{C,C}$. Let $\mathcal{M}$ have $\mathcal{M}_A = \{a\}$, $\mathcal{M}_B = \{a, b\}$, $\mathcal{M}_C = \{a, c\}$, let $f^{\mathcal{M}_{B,B}}$ be the constant function b; let $f^{\mathcal{M}_{C,C}}$ be the constant function c. Thus these two functions, each interpreting f, do not agree on a.

Filtering and the monotonicity condition on models will preclude this.

Example 3. [14]] $B \leq A, B \leq C$; $a : A, b : B, c : C$; postulate $a = c$.

The term algebra does not satisfy $a = c$. But the algebra that interprets A as $\{d, e\}$, C as $\{d, e\}$, B as $\{d\}$ with constant b interpreted as d and constants a, c both interpreted as e, does satisfy it — and this alg is isomorphic to the term algebra! So equations are not preserved under isomorphism.

Assumption of local filtering precludes this phenomenon.

The Term Model When the set of relation symbols in $\mathcal{L}$ is empty then the set of closed terms forms the universe of a model for $\mathcal{L}$, the *term algebra* [14]. We may view this as a model for an arbitrary order-sorted signature, as follows.

Definition 4. Fix $\mathcal{L} = (\mathcal{S}, \leq, \Sigma)$. The family $\mathcal{T}^{\mathcal{L}} = \{\mathcal{T}^{\mathcal{L},A} \mid A \in \mathcal{S}\}$ of *closed terms* over $\mathcal{L}$ is the $\subseteq$-least family satisfying

- $\Sigma_{\langle\rangle,A} \subseteq \mathcal{T}^{\mathcal{L},A}$;
- if $f \in \Sigma_{w,s}$ with $w = A_1 \ldots A_n$ and for each i, $t_i \in \mathcal{T}^{\mathcal{L},A_i}$ then $f(t_1, \ldots, t_n) \in \mathcal{T}^{\mathcal{L},A}$;

– is $A \leq A'$ then $\mathcal{T}^{\mathcal{L},A} \subseteq \mathcal{T}^{\mathcal{L},A'}$.

The family $\{\mathcal{T}^{\mathcal{L},A} \mid A \in \mathcal{S}\}$ determines a model $\mathcal{T}^{\mathcal{L}}$ of $\mathcal{L}$, the *term model*, by taking the interpretation $f^{\mathcal{T}^{\mathcal{L}}}$ 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.

The following result is an easy consequence of the initiality of term models in order-sorted algebra, but it is crucial to our development.

Theorem 5. *Suppose $\mathcal{L} = (\mathcal{S}, \leq, \Sigma)$ is a regular signature such that Σ has no relation symbols. Then the term model $\mathcal{T}^{\mathcal{L}}$ is initial. That is, for any model $\mathcal{M}$ of $\mathcal{L}$ there is a unique homomorphism from $\mathcal{T}^{\mathcal{L}}$ to $\mathcal{M}$.*

Proof. Initiality of $\mathcal{T}^{\mathcal{L}}$ in the category of *algebras* was shown by Goguen and Meseguer [14]. Now, given an $\mathcal{L}$-model $\mathcal{M}$, we let $\mathcal{M}'$ be the reduct of $\mathcal{M}$ to $\mathcal{L}'$, is the reduct of $\mathcal{L}$ obtained by removing the relation symbols: $\mathcal{M}'$ is a $\mathcal{L}$'-algebra. 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.

Here's why we want a cross-domain equality predicate (rather than each sort having its own equality predicate).

Example 6 ([14]). $A \leq B, C \leq B, C \leq D$, with $a : A$ etc. Equations $a = b$, $b = c$, $c = d$ want to imply $a = d$ by transitivity but A and D are incomparable. Leads to allowing equations between terms in same connected component.

Summary: we allow equations between terms in the same connected component (in order to support transitivity). But we may assume terms in an equation have the same sort (Definition 8) by local filtering.

3.1 Formulas and truth

Henceforth we assume that our signatures are coherent: regularity ensures that the term model is initial, and local filtering allows us to assume that terms in an equation have the same type (even though we have "cross-sort" equality, terms in the same connected component of the sort poset do have a common type).

Definition 7 (Open terms). Fix a signature $\mathcal{L} = (\mathcal{S}, \leq, \Sigma)$. Let X be an $\mathcal{S}$-sorted set $\{X_A \mid A \in \mathcal{S}\}$ of variables, with the X_A mutually disjoint and disjoint from Σ. The set $\mathcal{T}^{\mathcal{L}}(X)$ of *(open) terms over X* is, intuitively, obtained by adjoining the variables in X_A to the term model at type A. Formally we proceed as follows [14]. Define Σ^X to be the family of symbols with ${\Sigma^X}_{\langle\rangle,A} = \Sigma_{\langle\rangle,A} \cup X_A$ and ${\Sigma^X}_{w,A} = \Sigma_{w,A}$ for $w \neq \langle\rangle$. Then $\mathcal{L}^X$ is the signature $(\mathcal{S}, \leq, \Sigma^X)$. Then the family $\mathcal{T}^{\mathcal{L}}(X)$ of open terms over X is the family $\mathcal{T}^{\mathcal{L}^X}$ as defined in Definition **??**, that is the terms of $\mathcal{T}^{\mathcal{L}}(X)$ are the closed terms over $\mathcal{L}^X$.

As noted by Goguen and Meseguer [14] the fact that the open term algebra is initial in the category of $\mathcal{L}^X$-algebras entails the fact that this algebra is *free* over the generators X in the category of $\mathcal{L}$-algebras. In particular, if $\eta : X \to \mathcal{M}$ is an $\mathcal{S}$-sorted assignment of values in $\mathcal{M}$ to variables from X then there is a canonical way to extend η to map $\mathcal{T}^{\mathcal{L}}(X)$ to $\mathcal{M}$.

Definition 8 (Formulas). An *atomic formula* is one of

- $\bot$ (to be interpreted as falsehood)
- $P(t_1,...,t_n)$ where $P \in \Sigma_{\langle A_1,...A_n\rangle}$ and $t_i \in \mathcal{T}^{\Sigma,A_i}(X)$ for all $1 \leq i \leq n$.
- $t_1 = t_2$ where for some A, for all i, $t_i \in \mathcal{T}^{\Sigma,A}(X)$.

The formula $\bot$ will be convenient in Section 4. 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 $\forall x^A$.

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.

We don't need to introduce a constant for "true", as the negation of falsehood, since any sentence of the form $\forall x^A \,.\, x = x$ will serve (as will be seen, existential-free sentences will play a key role in the following, and $\bot$ provides an existential-free falsehood).

Definition 9. An *environment* η over a model $\mathcal{M}$ is an $\mathcal{S}$-indexed family of partial functions $\{\eta_s : X_s \to \mathcal{M}_A \mid A \in \mathcal{S}\}$ such that $\eta_A = (\eta_{A'})\restriction_{X_A}$ whenever $A \leq A'$. As usual the notation $\eta[x^A \mapsto e]$ refers to the environment agreeing with η except that it maps variable $x \in X_A$ to e. An environment η can be extended to terms in $\mathcal{T}^{\Sigma}(X)$ in the usual way.

Definition 10 (Truth in a model). Let $\mathcal{M}$ be 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 induction over φ as follows.

- If φ is $\bot$ then $\mathcal{M} \models_\eta \varphi$ fails.
- If $\varphi \equiv P(t_1,...,t_n)$ then $\mathcal{M} \models_\eta \varphi$ if and only if $P^{\mathcal{M}}(\eta(t_1),...,\eta(t_n))$ holds.
- If $\varphi \equiv t_1 = t_2$ then $\mathcal{M} \models_\eta \varphi$ if and only if $\eta(t_1) = \eta(t_2)$ holds.
- If $\varphi \equiv \neg\alpha$ then $\mathcal{M} \models_\eta \varphi$ if and only if $\mathcal{M} \not\models_\eta \alpha$.
- If $\varphi \equiv \alpha \wedge \beta$ then $\mathcal{M} \models_\eta \varphi$ iff $\mathcal{M} \models_\eta \alpha$ and $\mathcal{M} \models_\eta \beta$.
- If $\varphi \equiv \alpha \vee \beta$ then $\mathcal{M} \models_\eta \varphi$ iff $\mathcal{M} \models_\eta \alpha$ or $\mathcal{M} \models_\eta \beta$.
- If $\varphi \equiv \exists x^A \alpha$ then $\mathcal{M} \models_\eta \varphi$ iff there is some element e in $\mathcal{M}_A$ such that $\mathcal{M} \models_{\eta[x\mapsto e]} \alpha$.
- If $\varphi \equiv \forall x^A \alpha$ then $\mathcal{M} \models_\eta \varphi$ iff for each element e in $\mathcal{M}_A$ it holds that $\mathcal{M} \models_{\eta[x\mapsto e]} \alpha$.

We allow empty sorts in our models, indeed we even allow all sorts to be empty, and it is well-known that the possibility of empty sorts introduces subtleties into formal *deduction* [27]. But we are not interested in formal deduction *per se* and, given the fact that we require that the domain of an environment include all the free variables of a formula under consideration, there are no semantic difficulties arising from empty sorts. For example, if sort A is empty in a model then any environment η must have η_A be the empty function; as a consequence any formula $\exists x^A.\varphi$ will be false under η, and any formula $\forall x^A.\varphi$ will be true under η.

On reduction to unsorted logic It is not unusual for treatments of many-sorted logic to "encode" sorts as unary predicates and to view many-sorted logic as a particular syntactic discipline over standard one-sorted logic. For example one can reasonably view the sorted quantification $\exists x^A \,.\, \varphi$ as shorthand for $\exists x \,.\, (A(x) \wedge \varphi)$. This is the traditional approach taken in mathematical logic [8]. To handle subsorting is is natural to introduce *coercion functions:* to capture $A \leq B$ in the order-sorted setting we can add to the signature a function symbol $c_{AB} : A \to B$ in the flat setting. There are some natural axioms imposed on the coercion functions: that they are injective, that they compose properly to reflect transitivity, and so forth.

Definition 11. Let $(\mathcal{S}, \leq, \Sigma)$ be a signature. The *flat many-sorted encoding* of $\mathcal{L}$ is the signature $\mathcal{L}^+ = (\mathcal{S}, \emptyset, \Sigma^+)$ where

$$\Sigma^+ = \Sigma \cup \{c_{A,A'} \in \Sigma^+_{A,A'} \mid A \leq A'\}.$$

There are some natural axioms on the coercion functions.

1. $c_{A,A'}(x) = x$
2. $c_{A,A'}$ is injective
3. if $A \leq A' \leq A''$ then $c_{A',A''} \circ c_{A,A'} = c_{A,A''}$
4. each c commutes with functions in Σ.

The Reduction Theorem of [14]:

Theorem 12 ([14]). *When $(\mathcal{S}, \leq, \Sigma)$ is regular and locally filtered then order-sorted algebra can be reduced to ordinary many-sorted algebras satisfying the axioms above via an equivalence of categories.*

This theorem lifts, of course, to our setting of order-sorted *logic*. But, for a variety of reasons, we resist such an encoding into unsorted logic. First, it would make counting terms, our central concern, more difficult. This is because the closed terms would involve the coercion functions and we would have to count *modulo* the axioms governing them, otherwise we would overestimate the size of the true set of closed terms in the original signature. Second, introducing coercion functions would clutter the treatment of results such as Herbrand's Theorem. Finally, we want our results to provide clear information to users of tools—like Alloy—that are explicitly order-sorted, and an encoding into another formalism would be an obstacle to these users. So we prefer to work with order-sorted logic directly.

4 Skolemization

4.1 Negation-normal form

A formula is in *negation-normal* form if the negation sign is applied only to atomic formulas.

Lemma 13. *Every formula is logically equivalent to a formula in negation normal form.*

Proof. 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. ///

Failure of prenex-normal form The fact that models can have empty sorts changes the rules for how quantifiers may be moved within a formula. In particular the classical equivalence

$$((\exists x^A \alpha) \vee \beta) \text{ is equivalent to } \exists x^A (\alpha \vee \beta) \quad [x \text{ not free in } \beta]$$

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 14 (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 $\exists x^A.\varphi(x^A, x_1^{A_1}, \ldots, x_n^{A_n})$ 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 $\exists x^A.\varphi(x, x_1, \ldots, x_n)$ by $\varphi(f(x_1, \ldots, x_n), x_1, \ldots, x_n)$. 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.

It is not hard to see that any two Skolemizations of a sentence will differ only in the names of the new function symbols used. We do not need this result here and so will not prove it. But in order to unambiguously speak of *the* Skolemization of a sentence σ let us agree that we will eliminate existential formulas from left-to-right and use a canonical well-ordering of the universe of potential vocabulary symbols. With this understanding, if σ is a sentence over $\mathcal{L}$ we we will speak of "the Skolemization" of σ, and denote it σ_{sk}.

Lemma 15. *For any σ we have $\sigma_{sk} \models \sigma$.*

Proof. Note that the signature of σ_{sk} is an expansion of the signature of σ so the entailment claim makes sense. It suffices to show that the result of a single Skolem-step on σ entails σ; this is very easy to see from the definition. ///

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 16. 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.

We first note that the phenomenon in Example 16 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.

Definition 17. A model $\mathcal{M}$ is *safe for* formula φ if for every occurrence of a subformula $\exists x^A \,.\, \alpha$ in φ we have $\mathcal{M}_A \neq \emptyset$.

Lemma 18. *If $\mathcal{M} \models \sigma$ and $\mathcal{M}$ is safe for σ then there is an expansion $\mathcal{M}^*$ of $\mathcal{M}$ to the signature of σ_{sk} such that $\mathcal{M}^* \models \sigma_{sk}$.*

Proof. It suffices to show that the corresponding result holds for a single Skolem-step on σ entails σ, so suppose σ and σ' are as in Definition 14. The argument is just as for classical one-sorted logic: we can expand the model $\mathcal{M}$ to interpret the new function symbol f precisely because $\mathcal{M}$ satisfies the the original σ, but we must know that A is non-empty in case the truth of $\exists x^A.\sigma(x, x_1, \ldots, x_n)$ is needed for this. ///

This points the way to recovering a weak version of the classical equi-satisfiability result which will be good enough for our present purposes.

Definition 19. Let φ be a formula. An *approximation* of φ is a formula obtained by replacing some (zero or more) subformulas $\exists x^A \,.\, \alpha$ of φ by $\bot$.

Lemma 20. *If $\sigma_\perp$ is an approximation of σ then $\sigma_\perp \models \sigma$.*

Proof. We prove by induction over arbitrary formulas φ and approximations $\varphi_\perp$, and for arbitrary models $\mathcal{M}$ and environments η, if $\mathcal{M} \models_\eta \varphi_\perp$ then $\mathcal{M} \models_\eta \varphi$. Suppose $\mathcal{M} \models_\eta \varphi_\perp$.

- φ is a literal: then $\varphi_\perp = \varphi$. So certainly $\mathcal{M} \models_\eta \varphi$.
- $\varphi \equiv \alpha \vee \beta$: then $\varphi_\perp = \alpha_\perp \vee \beta_\perp$, where $\alpha_\perp$ and $\beta_\perp$, are approximations of α and β. Then $\mathcal{M} \models_\eta \alpha_\perp$ or $\mathcal{M} \models_\eta \beta_\perp$ so by induction $\mathcal{M} \models_\eta \alpha$ or $\mathcal{M} \models_\eta \beta$, as desired.
- $\varphi \equiv \alpha \wedge \beta$: then $\varphi_\perp = \alpha_\perp \wedge \beta_\perp$, where $\alpha_\perp$ and $\beta_\perp$, are approximations of α and β. Then $\mathcal{M} \models_\eta \alpha_\perp$ and $\mathcal{M} \models_\eta \beta_\perp$ so by induction $\mathcal{M} \models_\eta \alpha$ and $\mathcal{M} \models_\eta \beta$, as desired.
- $\varphi \equiv \forall x^A \alpha$: then $\varphi_\perp = \forall x^A \,.\, \alpha_\perp$, where $\alpha_\perp$ is an approximation of α. For every $e \in \mathcal{M}_A$ we have $\mathcal{M} \models_{\eta[x^A \mapsto e]} \alpha_\perp$, so by induction at each such e we have $\mathcal{M} \models_{\eta[x^A \mapsto e]} \alpha$, so $\mathcal{M} \models_\eta \forall x^A \,.\, \alpha$.
- $\varphi \equiv \exists x^A \alpha$: then either $\varphi_\perp = \exists x^A . \alpha_\perp$ or $\varphi_\perp = \perp$. In the former case we have, for some $e \in \mathcal{M}_A$, $\mathcal{M} \models_{\eta[x^A \mapsto e]} \alpha_\perp$, so by induction $\mathcal{M} \models_{\eta[x^A \mapsto e]} \alpha$, so $\mathcal{M} \models_\eta \exists x^A \,.\, \alpha$. The latter case cannot arise under the hypothesis that $\mathcal{M} \models_\eta \varphi_\perp$.

///

We can now prove a slightly weaker version of the traditional result on preservation of satisfiability.

Lemma 21. *If $\mathcal{M} \models \sigma$ then there is an approximation $(\sigma_\perp^{\mathcal{M}})$ of σ such that $\mathcal{M} \models \sigma_\perp^{\mathcal{M}}$ and $\mathcal{M}$ is safe for $\sigma_\perp^{\mathcal{M}}$.*

Proof. The sentence $\sigma_\perp^{\mathcal{M}}$ is obtained by replacing $\exists x^A \,.\, \alpha$ by $\perp$ precisely when $\mathcal{M}_A = \emptyset$. Formally we inductively define approximations $\varphi_\perp^{\mathcal{M}}$ for arbitrary formulas φ as follows.

- φ is a literal: then $\varphi_\perp^{\mathcal{M}} = \varphi$.
- $\varphi \equiv \exists x^A \alpha$ and $\mathcal{M}_A = \emptyset$: then $\varphi_\perp^{\mathcal{M}} = \perp$
- $\varphi \equiv \exists x^A \alpha$ and $\mathcal{M}_A \neq \emptyset$: then $\varphi_\perp^{\mathcal{M}} = \exists x^A . \alpha_\perp^{\mathcal{M}}$.
- $\varphi \equiv \forall x^A \alpha$: then $\varphi_\perp^{\mathcal{M}} = \forall x^A \,.\, \alpha_\perp^{\mathcal{M}}$.
- $\varphi \equiv \alpha \vee \beta$: then $\varphi_\perp^{\mathcal{M}} = \alpha_\perp^{\mathcal{M}} \vee \beta_\perp^{\mathcal{M}}$.
- $\varphi \equiv \alpha \wedge \beta$: then $\varphi_\perp^{\mathcal{M}} = \alpha_\perp^{\mathcal{M}} \wedge \beta_\perp^{\mathcal{M}}$.

It is clear from the construction that $\mathcal{M}$ is safe for $\varphi_\perp^{\mathcal{M}}$. We now claim that for an arbitrary environment η, if $\mathcal{M} \models_\eta \varphi$ then $\mathcal{M} \models_\eta \varphi_\perp^{\mathcal{M}}$. But this is a straightforward induction over formulas φ. The lemma follows by taking φ to be σ. ///

Lemma 22. *If σ is satisfiable then there exists an approximation $\sigma_\perp$ of σ such that $\sigma_{\perp sk}$ is satisfiable.*

Proof. By Lemma 21 and Lemma 18. ///

5 A Finite Model Theorem for Order-Sorted Logic

Model $\mathcal{M}$ is a *submodel* of model $\mathcal{N}$ if for each A, $\mathcal{M}_A \subseteq \mathcal{N}_A$ and each $f^{\mathcal{M}}$ and $R^{\mathcal{M}}$ are the restrictions of $f^{\mathcal{N}}$ and $R^{\mathcal{N}}$ to $\mathcal{M}$ and $\mathcal{M}$, respectively. 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).

5.1 Homomorphisms and Submodels

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 $(a_1, \ldots, a_n) \in X_1 \times \cdots \times X_n$ 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 23. *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. ///

We will denote the submodel identified in Lemma 23 as $h(\mathcal{M})$.

Remark 24. For future reference we observe that if $\mathcal{P}$ is a submodel of $\mathcal{M}$ and $e \in \mathcal{M}_B$ then it need not be the case that $e \in \mathcal{P}_B$ even if $e \in \bigcup\{\mathcal{P}_A \mid A \in \mathcal{S}\}$. Indeed this can happen even when $\mathcal{P}$ is obtained as the image of a homomorphism into $\mathcal{M}$. This has important consequences for the use of sorts as predicates, as we will discuss in Section 7.

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

Theorem 25. *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 $\mathcal{M}' \models \sigma$.*

Proof. We prove the following for arbitrary formulas φ: *for any environment η over $\mathcal{M}'$, if $\mathcal{M} \models_\eta \varphi$ then $\mathcal{M}' \models_\eta \varphi$.* Suppose φ is $P(t_1, ..., t_n)$ or $\neg P(t_1, ..., t_n)$, where P is a relation symbol from Σ or the equality symbol. Each $\eta(t_i)$ is in the domain of $\mathcal{M}'$, so we have that $\mathcal{M} \models_\eta P(t_1, ..., t_n)$ iff $\mathcal{M}' \models_\eta P(t_1, ..., t_n)$ by definition of submodel.

Proceeding inductively: when φ is $\alpha \wedge \beta$ or φ is $\alpha \vee \beta$ the result is an immediate application of the induction hypothesis. Finally suppose that φ is $\forall x^A \alpha$. Then $\mathcal{M} \models \varphi$ implies that $\mathcal{M} \models_{\eta[x^A \mapsto e]} \alpha$ for all $e \in \mathcal{M}_A$. Since $\mathcal{M}_A \supseteq \mathcal{M}'_A$ we have $\mathcal{M} \models_{\eta[x^A \mapsto e]} \alpha$ for all $e \in \mathcal{M}'_A$, that is, $\mathcal{M}' \models_\eta \forall x^A \alpha$. ///

We pause here to point out that Theorem 25 *fails* if sort names are permitted to be used as unary predicates. This is simply because in the definition of submodel there is no requirement that, for example, if element e lies in sort A in a model then e is in sort A in the submodel. This issue is discussed in detail in Section 7.

5.2 The Kernel of a Model

Definition 26 (The kernel of a model). Let $\mathcal{M}$ be a model for the regular signature $\mathcal{L} = (\mathcal{S}, \leq, \Sigma)$. Let h be the unique homomorphism from $\mathcal{T}^{\mathcal{L}}$ to $\mathcal{M}$ (*c.f.* Theorem 5). The image of h is a submodel of $\mathcal{M}$ by Lemma 23; 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 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 Skolem expansion, *i.e.* 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 $H_{\mathcal{M}}(X)$ 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 27. 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 $\mathcal{K}_A = \{b_0\}$, $\mathcal{K}_B = \emptyset$, $f^{\mathcal{K}} = \emptyset$.

5.3 A Finite Model Theorem

Here we present our main theorem.

Theorem 28. *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 22 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 ${\mathcal{T}^{\mathcal{L}^{**}}}_A$, and thus no greater than the cardinality of ${\mathcal{T}^{\mathcal{L}^*}}_A$. 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 29. *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.

In Section 6 we will show 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 28, this establishes a decision procedure for satisfiability of OS-EPL sentences.

5.4 Herbrand's Theorem

As a brief digression, we address Herbrand's Theorem for order-sorted logic. The standard model-theoretic proof of Herbrand's Theorem in first-order logic uses in an essential way the fact that every element in the Skolem hull of a model is named by a term. As we have noted this fails when sorts may overlap. There are proof-theoretic approaches to Herbrand's Theorem of course but proof theory of order-sorted logic, especially in the presence of empty sorts, is delicate, so a model-theoretic proof would be nice to have. We are in a position to give such a proof here.

Theorem 30 (Herbrand's Theorem for Order-Sorted Logic). *Let* $\tau \equiv \forall y_1, \ldots, y_n \,.\, \alpha$ *be a sentence with* α *quantifier-free. Then* τ *is unsatisfiable iff there is a set* $\{\alpha_1, \ldots, \alpha_k\}$ *of closed instances of* α *such that* $(\alpha_1 \wedge \cdots \wedge \alpha_k)$ *is unsatisfiable.*

Proof. It is convenient to prove the following expanded version of the theorem. The following are equivalent.

1. $\tau = \forall y_1 \ldots y_n \,.\, \alpha$ is satisfiable.
2. The set $T = \{\alpha_* \mid \alpha_* \text{ is a closed instance of } \alpha\}$ is satisfiable.
3. For every finite subset $\{\alpha_1, \ldots, \alpha_n\}$ of closed instances of α, $\quad (\alpha_1 \wedge \cdots \wedge \alpha_n)$ is satisfiable.

The implication 1 to 2 is immediate, since any model of τ will satisfy T. To see that 2 implies 1: let $\mathcal{M} \models T$. Let $\mathcal{M}_0$ be the kernel of $\mathcal{M}$. So $\mathcal{M}_0 \models T$ by Theorem 25. Then $\mathcal{M}_0 \models \tau$ because the universal quantifier just ranges over closed terms when interpreted in $\mathcal{M}_0$.

The equivalence of (2) and (3) is just the Compactness Theorem *for ordinary propositional logic.* ///

It is worth noting that we do not in the above proof, require the Compactness Theorem for order-sorted logic *per se*.

6 Algorithms

In this section we present an algorithm to determine, given a signature $\mathcal{L} = (\mathcal{S}, \leq .\Sigma)$ which sorts of $\mathcal{S}$ are inhabited by only finitely many closed terms, and an algorithm to count the number of closed terms inhabiting a sort.

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

6.1 Testing OS-EPL membership

Definition 31. Let $\mathcal{L} = (\mathcal{S}, \Sigma)$ be a many-sorted signature. The grammar $G_{\mathcal{L}}$ is defined 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:

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

Recall that a non-terminal X in a CFG 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 32. *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* $\mathcal{T}^{\mathcal{L},A}$ *if and only if there is a derivation* $A \Rightarrow^* u$ *in* $G_{\mathcal{L}}$. *A sort* A *is inhabited by closed 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* $(\mathcal{T}^{\mathcal{L}})_A$ *is finite if and only if* A *generates only finitely many terms in 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 33. *There is an algorithm that, given an order-sorted signature $\mathcal{L}$, determines (uniformly) for each sort A, whether ${\mathcal{T}^{\mathcal{L}}}_A$ is finite. The algorithm runs in time linear in the total size of $\mathcal{L}$.*

Proof. By Lemma 32, ${\mathcal{T}^{\mathcal{L}}}_A$ 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 [18]. Transliterated into our setting the algorithm is as follows. First restrict attention to those sorts A that are inhabited by closed strictly-typed terms (*i.e.* eliminate "useless symbols" from the grammar $G_{\mathcal{L}}$): this can be done in linear time with a judicious choice of data structures (see for example [18]). Next, 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 $A \leq B$. Having ensured in the previous step that each sort named by a non-terminal in $G_{\mathcal{L}}$ is inhabited, it is the case that A generates infinitely many terminal strings if and only if there is a path from A to a cycle. The set of such sorts can be checked in linear time by a depth-first search. 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 34. Return to Example 1 from the introduction. 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 35. *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 33, we can decide whether all sorts of $\mathcal{L}^*$ are finitary in time linear in σ. ///

6.2 Computing the number of terms in a sort

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.

The intuition behind Algorithm 1 is as follows. 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* $h-1$.

Theorem 36. *There is an algorithm that, given a regular signature $\mathfrak{L}$, computes, in time cubic in the size of $\mathfrak{L}$, the size of $\mathcal{T}^{\mathfrak{L},A}$ for each finitary sort A (returning "∞" for the non-finitary sorts).*

Proof. The algorithm is given as Algorithm 1 below.

Proof of correctness Since the algorithm uses only FOR loops, it is easy to see that it terminates. Furthermore we claim that after termination, the totals of each column $\mathsf{Tbl}[\sum][A]$ contain exactly $|\mathcal{T}^{\mathfrak{L},A}|$ for each finitary sort A.

First observe that by the pigeonhole principle, all terms in $\mathcal{T}^{\mathfrak{L},A}$ (A finitary) must have *height* $\leq \mathsf{n_f}$. Therefore, when counting terms in finitary sorts it suffices to count only terms of *height* $\leq \mathsf{n_f}$, and thus we need only prove that the algorithm populates the table correctly: that each $\mathsf{Tbl}[h][A]$ contains exactly the number of terms having height h within $\mathcal{T}^{\mathfrak{L},A}$.

Proof: After a row is computed by our algorithm, it is never again modified. So we proceed by induction on h.

Base: If $h = 0$ then we are concerned only with constant terms. The first block of our algorithm counts every constant $c : S$ exactly once in each $\mathsf{Tbl}[0][S']$ such that $S \leq S'$. So we can conclude that $\mathsf{Tbl}[0]$ contains a faithful count of height 0 terms for all sorts.

Induction: Suppose $h > 0$ and that each $\mathsf{Tbl}[x]$, $0 \leq x < h$ is correct. A non-constant term t is in $\mathcal{T}^{\mathfrak{L},A}$ if (by definition!):

1. t has a function at its head with result sort A
2. t has a function at its head with some result sort B with $B \leq A, B \neq A$.

The algorithm increments a table cell according to case (2) if and only if it has already incremented a cell according to case (1).

Each ground term of height $h > 0$ has one distinct function f at its head, and (with respect to the ordering in f's arity) exactly one left-most subterm of height $h-1$ at index lm, $1 \leq \mathsf{leftmost} \leq n$. So we only need to show that we correctly calculate the number of terms in $\mathcal{T}^{\mathfrak{L},A}$ having height h, head function f with result sort A and left-most $h-1$ subterm at index $\mathsf{leftmost}$

The number of such terms depends only on the number of subterms available to fill each index of f's arity. The number of usable subterms for A_i is:

- If $i = \mathsf{leftmost}$, terms of sort A_i having height exactly $h-1$ are admissible.
- If $i < \mathsf{leftmost}$, terms of sort A_i having height up to $h-2$ are admissible.
- If $i > \mathsf{leftmost}$, terms of sort A_i having height up to $h-1$ are admissible.

(The usable heights differ by index since index $\mathsf{leftmost}$ is the *leftmost* appearance of a height $h-1$ subterm.)

But this is exactly the calculation that the algorithm makes, and by our induction hypothesis, these subterm rows have been calculated correctly.

Algorithm .1: The Counting Algorithm

Input: A signature $\mathcal{L} = (\mathcal{S}, \leq, \Sigma)$
Output: For each sort s for which Σ in finitary, the number of terms of sort s.

let $\mathsf{n_s}$ be the number of sorts ;
let $\mathsf{n_f}$ be the number of functions ;
let Tbl be a 2-dimensional vector of size $[\mathsf{n_f}+1][\mathsf{n_s}]$;
initialize Tbl with 0s ;

```
// Fill the first row of the table with the number of constant terms of
   each sort.
```

for *each constant symbol* $c : S$ **do**
    // Populate
    Increment Tbl [0][S] by 1;
    // Propagate to supersorts
    for *each sort S' such that* $S < S'$ **do**
        Increment Tbl [0][S'] by 1;

for h = *1 to* $\mathsf{n_f}$ **do**
    // This row begins initialized to 0
    foundTermsOfThisHeight := false;
    for *each function* $f : (A_1, ..., A_n) \rightarrow B$ **do**
        // Compute the number of ways to construct a term of height h via
           f. (To be this height, must use at least one term of height
           h-1.)
        ways = 0 ;
        // How many ways to make a term where the left-most height h-1
           component term is...
        for leftmost= *1 to n* **do**
            // Start with the number of h-1 height terms having this sort
            $\mathsf{ways}_n = \mathsf{Tbl}[h-1][\mathsf{leftmost}]$;
            // And multiply by the number of available subterms of the
               other sorts in f's arity
            for component= *1 to* $(\mathsf{leftmost}-1)$ **do**
                $\mathsf{ways}_n* = \sum\{\mathsf{Tbl}[k][\mathsf{component}] \mid 0 \leq k \leq h-2\}$
            for component= $(\mathsf{leftmost}+1)$ *to n* **do**
                $\mathsf{ways}_n* = \sum\{\mathsf{Tbl}[k][\mathsf{component}] \mid 0 \leq k \leq h-1\}$
            $\mathsf{ways} += \mathsf{ways}_n$
        // ways contains the number of new height h terms produced by
           function f. Add those terms to the table...
        $\mathsf{Tbl}[h][B] += \mathsf{ways}$;
        if ways > 0 **then**
            foundTermsOfThisHeight := true
        // ... and propagate to supersorts
        for *each sort S' such that* $B < S'$ **do**
            $\mathsf{Tbl}[h][S'] += \mathsf{ways}$
    // If no height h terms were found, there aren't any larger terms
       either.
    if *not foundTermsOfThisHeight* **then**
        break;

return a 1-dimensional vector of size $\mathsf{n_s}$which contains the column sums of Tbl

Complexity Note that we could optimize the counting algorithm by memoizing column totals, saving us the trouble of repeatedly summing up $\sum\{\mathsf{Tbl}[k][\mathsf{component}] \mid 0 \leq k \leq h-1\}$ and $\sum\{\mathsf{Tbl}[k][\mathsf{component}] \mid 0 \leq k \leq h-1\}$. The code for this is omitted for clarity, but we assume it when calculating the complexity bounds below.

The initial pass for row 0 takes no more than $\mathsf{n_c}\mathsf{n_s}$ steps.

The main block (with memoization) has loop structure as follows:

(1) **for** *each* $h \leq \mathsf{n_f}$ **do**
  (2) **for** *each* f *(*$\leq \mathsf{n_f}$ *iterations)* **do**
    (3) **for** *each lm in f's arity* **do**
      (4) **for** *each component in f's arity not equal to lm* **do**
        (iterations bounded by maximum arity) ...
    (5) **for** *each supersort of f's result sort (*$\leq \mathsf{n_s}$ *iterations)* **do**
      ...

Together (2) and (3) make $|\Sigma|$, the size of the signature. Therefore we have a bound: $\mathsf{n_c}\mathsf{n_s} + \mathsf{n_f}(|\Sigma| * \mathit{maxarity} + \mathsf{n_f}\mathsf{n_s})$, which is cubic in the size of the signature.

///

We could use this algorithm to test for finitary signatures as well, since if we continue to iterate term height past $|\Sigma_F|$, there will be an increase in $\mathsf{Tbl}[h][S]$ if and only if S is infinitary. However, it benefits us to know *in advance* which sorts are finitary and which functions never produce ground terms, as that may greatly reduce the size of Tbl.

If we want to know the total number of terms across all sorts (without duplication), it is easy enough to add a counter and increment it on population (not propagation) in the algorithm above.

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 28. Since these bounds are computable we can effectively decide satisfiability for such sentences.

Remark 37. The results of the algorithm in Theorem 36 can be useful even if not all sorts are finitary. Fontaine and Gribomont [11] have implemented an instantiation-based algorithm that takes advantage of the information that certain sorts are guaranteed to have finitely many closed 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.

7 About Sorts-as-Predicates

Many formulations of sorted logic, and certain tools, allow sort names to be used as predicate names in formulas. We have not built this into our syntax; in this section we explain why. We start with an example, in which Herbrand's Theorem fails in a dramatic way, with obvious negative consequences for our finite model theorem.

Example 38. Consider a signature $\mathcal{L} = (\mathcal{S}, \leq, \Sigma)$ with sorts A and B, a constant $b \in \Sigma_{\langle\rangle,B}$ and a function $f \in \Sigma_{A,B}$. Let σ be the following sentence expressing the fact that f is one-to-one but not onto

$$(\forall x^A \,.\, B(x)) \wedge (\forall y^B \,.\, A(y)) \wedge (\forall x^A \,.\, f(x) \neq b) \wedge (\forall x_1^A x_2^A \,.\, (x_1 \neq x_2) \rightarrow (f(x_1) \neq f(x_2))).$$

Since the first two conjuncts force A and B to be equal, this sentence has only infinite models. But the Herbrand universe for $\mathcal{L}$ is the singleton set $\{b\}$.

What went wrong? The fundamental fact that the truth of existential-free sentences is preserved under submodel, Theorem 25, fails when sorts are allowed as predicates. This is because in the definition of submodel there is no requirement that elements remain in each sort they inhabit in the original model: if $\mathcal{P}$ is a submodel of $\mathcal{M}$ and element e happens to be in $\mathcal{M}_A \cap \mathcal{M}_B$ then it is possible that, say, $\mathcal{P}_A$ but not in $\mathcal{P}_B$. So Theorem 25, crucial to Herbrand's Theorem, as well as to the soundness of our decision procedure, fails at the base case.

A natural response to this might be: if sorts are to used as predicates then the notion of submodel should be refined to reflect this. In particular we might refine the definition of submodel and insist that an element in the universe of a submodel retain all of the "sort-memberships" it had in the original model. Unfortunately, if we do this this something else breaks: the fact that the image of a homomorphism makes a submodel (Lemma 23). Recall that closure under under the functions of a vocabulary is a property of a *family* of sets (*e.g.* the family of images of sorts in the source model) and not the union of this family. When we put a sort-structure on the union, in the target model, of the images of sorts in the source by retaining the sort-memberships in the target the resulting family of sets can fail to be closed under the interpretations of the function symbols.

Example 39. Refer to Example 27; consider the unique homomorphism $h : \mathcal{K} \rightarrow \mathcal{M}$, for which h_A maps b_0 to b_0 and $h_B = \emptyset$. The submodel of $\mathcal{M}$ generated by h interprets sort A as $\{b_0\}$, interprets B as $\emptyset$, and is the universe of a submodel of $\mathcal{M}$ (in which f is interpreted as the empty function). But if we were to insist that element b_0 inhabit sort B in the "submodel" induced by h, then we cannot interpret f: it is supposed to be the restriction of $f^{\mathcal{M}}$ to the universe of our submodel, but $f^{\mathcal{M}}(b_0) = b_1$, and b_1 is not in the range of h.

We stress that these observations are not bound up with our project of trying to build finite models, they are general foundational problems with using sorts as predicates. If we permit sorts to be used as predicates, we must either give up the notion of models being closed under homomorphisms or give up the intuitively compelling model-theoretic result that universal sentences are preserved under submodel.

The solution is to view the use of sorts as predicates as syntactic sugar for formulas in the core language. The construction for de-sugaring is the following. Given σ with subterm $A(t)$ for a sort A, rewrite this to replace $A(t)$ with $(\exists z^A \,.\, z = t)$ (where z is a fresh variable). This is done before passing to negation-normal form, so as a consequence a subformula $\neg A(t)$ will be replaced with $(\forall z^A \,.\, z \neq t)$.

Lemma 40. *If σ' is obtained from σ by the process described in the previous paragraph then σ' and σ are logically equivalent.*

Proof. Recall that we define the truth of a formula φ in a model $\mathcal{M}$ in the context of environments η under the assumption that the domain of η includes all the free variables of φ. Noting that $A(t)$ has the same free variables as $(\exists z^A \,.\, z = t)$ when z is fresh variable, we must show that for any $\mathcal{M}$ and environment η such that the domain of η includes the free variables of t), $\mathcal{M} \models_\eta A(t)$ if and only if $\mathcal{M} \models_\eta (\exists z^A \,.\, z = t)$. The fact that $\mathcal{M}_A$ might be empty does not cause any problems: if $\mathcal{M} \models_\eta A(t)$ this means that $\eta(t) \in \mathcal{M}_A$ and so we can bind z to $\eta(t)$ to witness the truth of $(\exists z^A \,.\, z = t)$; while if $\mathcal{M} \not\models_\eta A(t)$ this means that $\eta(t) \notin \mathcal{M}_A$ and so $\mathcal{M} \models (\forall z^A \,.\, z \neq t)$. ///

Example 41. We revisit Example 38. When the sentence there is de-sugared according to the recipe above we arrive at

$$(\forall x^A \exists u^B \,.\, u = x) \wedge (\forall y^B \exists w^A \,.\, w = y) \wedge (\forall x^A \,.\, f(x) \neq b) \wedge (\forall x_1^A x_2^A \,.\, (x_1 \neq x_2) \rightarrow (f(x_1) \neq f(x_2))).$$

When this sentence is Skolemized we get a function from A to B and one from B to A; together with the constant b this obviously generates an infinite set of closed terms.

Summarizing: the use of sorts as predicates is not innocent, but it can be can be accommodated after translation into the core language.

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** (2010) 153 – 172 Automated Deduction: Decidability, Complexity, Tractability.
2. Bernays, P., Schönfinkel, M.: Zum entscheidungsproblem der mathematischen Logik. Mathematische Annalen **99** (1928) 342–372
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. Number 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. De Moura, L., Bjorner, N.: Z3: An efficient SMT solver. In: Tools and Algorithms for the Construction and Analysis of Systems. Lecture Notes in Computer Science, Vol. 4963. Springer (2008) 337
7. 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. Springer (2008) 410–425

8. Enderton, H.B.: A Mathematical Introduction to Logic. Academic Press (1972)
9. Feferman, S.: Many-Sorted Interpolation Theorems and Applications. In: Proceedings of the Tarski Symposium, AMS Proc. Symp. in Pure Math. Vol. 25. (1974) 205–223
10. Fisler, K., Krishnamurthi, S., Meyerovich, L.A., Tschantz, M.C.: Verification and change-impact analysis of access-control policies. In: International Conference on Software Engineering. (2005) 196–205
11. 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. Springer-Verlag (2003) 97–112
12. 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, Berlin, Heidelberg, Springer-Verlag (2009) 306–320
13. Goguen, J.A., Diaconescu, R.: An oxford survey of order sorted algebra. Mathematical Structures in Computer Science **4** (1994) 363–392
14. Goguen, J.A., Meseguer, J.: Order-Sorted Algebra I: Equational Deduction for Multiple Inheritance, Overloading, Exceptions and Partial Operations. Theor. Comput. Sci. **105** (1992) 217–273
15. Harrison, J.: Exploiting sorts in expansion-based proof procedures (Unpublished manuscript) http://www.cl.cam.ac.uk/~jrh13/papers/manysorted.pdf.
16. Herbrand, J.: Recherches sur la théorie de la démonstration. PhD thesis, Université de Paris, Paris, France (1930)
17. Hooker, J., Rago, G., Chandru, V., Shrivastava, A.: Partial instantiation methods for inference in first-order logic. Journal of Automated Reasoning **28** (2002) 371–396
18. Hopcroft, J.E., Motwani, R., Ullman, J.D.: Introduction to Automata Theory, Languages, and Computation. Third edn. Addison-Wesley, Reading, Massachusetts (2006)
19. Jackson, D.: Software Abstractions: Logic, Language, and Analysis. The MIT Press (2006)
20. Jereslow, R.G.: Computation-oriented reductions of predicate to propositional logic. Decision Support Systems **4** (1988) 183–197
21. Lahiri, S., Seshia, S.: The UCLID decision procedure. In: 16th International Conference, Computer-Aided Verification. Springer (2004) 475–478
22. Lewis, H.: Complexity results for classes of quantificational formulas. J. COMP. AND SYS. SCI. **21** (1980) 317–353
23. Momtahan, L.: Towards a small model theorem for data independent systems in Alloy. Electronic Notes in Theoretical Computer Science **128** (2005) 37 – 52 Proceedings of the Fouth International Workshop on Automated Verification of Critical Systems (AVoCS 2004).
24. Oberschelp, A.: Order sorted predicate logic. In: Workshop on Sorts and Types in Artificial Intelligence. Springer (1989) 1–17
25. Piskac, R., de Moura, L., Bjørner, N.: Deciding Effectively Propositional Logic with Equality. Technical Report MSR-TR-2008-181, Microsoft Research (2008)
26. Ramsey, F.P.: On a problem in formal logic. Proceedings of the London Mathematical Society **30** (1930) 264–286
27. Smolka, G., Nutt, W., Goguen, J.A., Meseguer, J.: Order-sorted equational computation. In: Aït-Kaci, H., Nivat, M. (eds.): Resolution of Equations in Algebraic Structures. Vol. 2: Rewriting Techniques. Academic Press, New York (1989) 299–369
28. Walther, C.: Many-sorted unification. Journal of the ACM (JACM) **35** (1988) 1–17
29. Weibel, T.: An order-sorted resolution in theory and practice. Theoretical computer science **185** (1997) 393–410