Locality Results for Certain Extensions of Theories with Bridging Functions

Viorica Sofronie-Stokkermans

Max-Planck-Institut für Informatik, Campus E 1.4, Saarbrücken, Germany

Abstract. We study possibilities of reasoning about extensions of base theories with functions which satisfy certain recursion (or homomorphism) properties. Our focus is on emphasizing possibilities of hierarchical and modular reasoning in such extensions and combinations thereof. We present practical applications in verification and cryptography.

1 Introduction

In this paper we study possibilities of reasoning in extensions of theories with functions which satisfy certain recursion (or homomorphism) axioms. This type of axioms is very important in verification – for instance in situations in which we need to reason about functions defined by certain forms of primitive recursion – and in cryptography, where one may need to model homomorphism axioms of the form $\forall x, y, z(\mathsf{encode}_z(x * y) = \mathsf{encode}_z(x) * \mathsf{encode}_z(y))$. Decision procedures for recursive data structures exist. In [13], Oppen gave a PTIME decision procedure for absolutely free data structures based on bidirectional closure; methods which use rewriting and/or basic equational reasoning were given e.g. by Barrett et al. [2] and Bonacina and Echenim [3]. Some extensions of theories with recursively defined functions and homomorphisms have also been studied. In [1], Armando, Rusinowitch, and Ranise give a decision procedure for a theory of homomorphisms. In [18], Zhang, Manna and Sipma give a decision procedure for the extension of a theory of term structures with a recursively defined length function. In [8] tail recursive definitions are studied. It is proved that tail recursive definitions can be expressed by shallow axioms and therefore define so-called "stably local extensions". Locality properties have also been studied in a series of papers on the analysis of cryptographic protocols (cf. e.g. [4,5,6]).

In this paper we show that many extensions with recursive definitions (or with generalized homomorphism properties) satisfy locality conditions. This allows us to significantly extend existing results on reasoning about functions defined using certain forms of recursion, or satisfying homomorphism properties [1,8,18], and at the same time shows how powerful and widely applicable the concept of local theory (extension) is in automated reasoning. As a by-product, the methods we use provide a possibility of presenting in a different light (and in a different form) locality phenomena studied in cryptography in [4,5,6]; we believe that they will allow to better separate rewriting from proving, and thus to give simpler proofs. The main results are summarized below:

R.A. Schmidt (Ed.): CADE 2009, LNAI 5663, pp. 67–83, 2009.

- We show that the theory of absolutely free constructors is local, and locality is preserved also in the presence of selectors. These results are consistent with existing decision procedures for this theory [13] which use a variant of bi-directional closure in a graph formed starting from the subterms of the set of clauses whose satisfiability is being checked.
- We show that, under certain assumptions, extensions of the theory of absolutely free constructors with functions satisfying a certain type of recursion axioms satisfy locality properties, and show that for functions with values in an ordered domain we can combine recursive definitions with boundedness axioms without sacrificing locality. We also address the problem of only considering models whose data part is the *initial* term algebra of such theories.
- We analyze conditions which ensure that similar results can be obtained if we relax some assumptions about the absolute freeness of the underlying theory of data types, and illustrate the ideas on an example from cryptography.

The locality results we establish allow us to reduce the task of reasoning about the class of recursive functions we consider to reasoning in the underlying theory of data structures (possibly combined with the theories associated with the co-domains of the recursive functions).

Structure of the paper. In Section 2 we present the results on local theory extensions and hierarchical reasoning in local theory extensions needed in the paper. We start Section 3 by considering theories of absolutely free data structures, and extensions of such theories with selectors. We then consider additional functions defined using a certain type of recursion axioms (possibly having values in a different – e.g. numeric – domain). We show that in these cases locality results can be established. In Section 4 we show that similar results can be obtained if we relax some assumptions about the absolute freeness of the underlying theory of data types, and illustrate the results on a simple example from cryptography.

2 Preliminaries

We will consider theories over possibly many-sorted signatures $\Pi = (S, \Sigma, \mathsf{Pred})$, where S is a set of sorts, Σ a set of function symbols, and Pred a set of predicate symbols. For each function $f \in \Sigma$ (resp. predicate $P \in \mathsf{Pred}$), we denote by $a(f) = s_1, \ldots, s_n \rightarrow s$ (resp. $a(P) = s_1, \ldots, s_n$) its arity, where $s_1, \ldots, s_n, s \in S$, and $n \geq 0$. In the one-sorted case we will simply write $a(f) = n$ (resp. $a(P) = n$).

First-order theories are sets of formulae (closed under logical consequence), typically the set of all consequences of a set of axioms. When referring to a theory, we can also consider the set of all its models. We here consider theories specified by their sets of axioms, but – usually when talking about local extensions of a theory – we will refer to a theory, and mean the set of all its models.

The notion of *local theory* was introduced by Givan and McAllester [9,10]. They studied sets $\mathcal{K}$ of Horn clauses with the property that, for any ground Horn clause C, $\mathcal{K} \models C$ only if already $\mathcal{K}[C] \models C$ (where $\mathcal{K}[C]$ is the set of instances of $\mathcal{K}$ in which all terms are subterms of ground terms in $\mathcal{K}$ or C).

Theory Extensions. We here also consider *extensions of theories*, in which the signature is extended by new *function symbols* (i.e. we assume that the set of predicate symbols remains unchanged in the extension). Let $\mathcal{T}_0$ be an arbitrary theory with signature $\Pi_0 = (S, \Sigma_0, \mathsf{Pred})$. We consider extensions $\mathcal{T}_1$ of $\mathcal{T}_0$ with signature $\Pi = (S, \Sigma, \mathsf{Pred})$, where the set of function symbols is $\Sigma = \Sigma_0 \cup \Sigma_1$. We assume that $\mathcal{T}_1$ is obtained from $\mathcal{T}_0$ by adding a set $\mathcal{K}$ of (universally quantified) clauses in the signature Π.

Partial Models. Let $\Pi = (S, \Sigma, \mathsf{Pred})$. A *partial Π-structure* is a structure $(\{A_s\}_{s\in S}, \{f_A\}_{f\in\Sigma}, \{P_A\}_{P\in\mathsf{Pred}})$ in which for every $f\in\Sigma$, with $a(f)=s_1,\ldots, s_n\to s$, f_A is a (possibly partially defined) function from $A_{s_1}\times\cdots\times A_{s_n}$ to A_s, and for every $P\in\mathsf{Pred}$ with arity $a(P)=s_1\ldots s_n$, $P_A\subseteq A_{s_1}\times\cdots\times A_{s_n}$. A *weak Π-embedding* between partial structures $A=(\{A_s\}_{s\in S}, \{f_A\}_{f\in\Sigma}, \{P_A\}_{P\in\mathsf{Pred}})$ and $B=(\{B_s\}_{s\in S}, \{f_B\}_{f\in\Sigma}, \{P_B\}_{P\in\mathsf{Pred}})$ is an S-sorted family $i=(i_s)_{s\in S}$ of injective maps $i_s : A_s \to B_s$ which is an embedding w.r.t. Pred, s.t. if $a(f)=s_1,\ldots,s_n\to s$ and $f_A(a_1,\ldots,a_n)$ is defined then $f_B(i_{s_1}(a_1),\ldots,i_{s_n}(a_n))$ is defined and $i_s(f_A(a_1,\ldots,a_n))=f_B(i_{s_1}(a_1),\ldots,i_{s_n}(a_n))$.

We now define truth and satisfiability in partial structures of Π-literals and (sets of) clauses with variables in a set X. If A is a partial structure, $\beta : X\to A$ is a valuation[1] and $L=(\neg)P(t_1,\ldots,t_n)$ is a literal (with $P\in\mathsf{Pred}\cup\{=\}$) we say that $(A,\beta)\models_w L$ if (i) either $\beta(t_i)$ are all defined and $(\neg)P_A(\beta(t_1),\ldots,\beta(t_n))$ is true in A, or (ii) $\beta(t_i)$ is not defined for some argument t_i of P. Weak satisfaction of clauses ($(A,\beta)\models_w C$) is defined in the usual way. A is a *weak partial model* of a set $\mathcal{K}$ of clauses if $(A,\beta)\models_w C$ for every $\beta : X\to A$ and every clause $C\in\mathcal{K}$. A *weak partial model* of $\mathcal{T}_0\cup\mathcal{K}$ is a weak partial model of $\mathcal{K}$ whose reduct to Π_0 is a total model of $\mathcal{T}_0$.

Local Theory Extensions. Consider the following condition (in what follows we refer to sets G of ground clauses and assume that they are in the signature $\Pi^c=(S,\Sigma\cup\Sigma_c,\mathsf{Pred})$, where Σ_c is a set of new constants):

(Loc) For every finite set G of ground clauses $\mathcal{T}_1\cup G\models\bot$ iff $\mathcal{T}_0\cup\mathcal{K}[G]\cup G$ has no weak partial model with all terms in $\mathsf{st}(\mathcal{K},G)$ defined

where if T is a set of terms, $\mathcal{K}[T]$ is the set of instances of $\mathcal{K}$ in which all terms starting with a symbol in Σ_1 are in T, and $\mathcal{K}[G]:=\mathcal{K}[\mathsf{st}(\mathcal{K},G)]$, where $\mathsf{st}(\mathcal{K},G)$ is the family of all subterms of ground terms in $\mathcal{K}$ or G.

We say that an extension $\mathcal{T}_0\subseteq\mathcal{T}_1$ is *local* if it satisfies condition (Loc). We say that it is *local for clauses with a property P* if it satisfies the locality conditions for all ground clauses G with property P. A more general locality condition (ELoc) refers to situations when $\mathcal{K}$ consists of formulae $(\Phi(x_1,\ldots,x_n)\vee C(x_1,\ldots,x_n))$, where $\Phi(x_1,\ldots,x_n)$ is a *first-order Π_0-formula* with free variables $x_1,\ldots,x_n$, and $C(x_1,\ldots,x_n)$ is a *clause* in the signature Π. The free variables $x_1,\ldots,x_n$ of such an axiom are considered to be universally quantified [14].

[1] We denote the canonical extension to terms of a valuation $\beta : X\to A$ again by β.

(ELoc) For every formula $\Gamma = \Gamma_0 \cup G$, where Γ_0 is a Π_0^c-sentence and G is a finite set of ground Π^c-clauses, $\mathcal{T}_1 \cup \Gamma \models \bot$ iff $\mathcal{T}_0 \cup \mathcal{K}[G] \cup \Gamma$ has no weak partial model in which all terms in $\mathsf{st}(\mathcal{K}, G)$ are defined.

A more general notion, namely Ψ-locality of a theory extension (in which the instances to be considered are described by a closure operation Ψ) is introduced in [11]. Let $\mathcal{K}$ be a set of clauses. Let $\Psi_{\mathcal{K}}$ be a closure operation associating with any set T of ground terms a set $\Psi_{\mathcal{K}}(T)$ of ground terms such that all ground subterms in $\mathcal{K}$ and T are in $\Psi_{\mathcal{K}}(T)$. Let $\Psi_{\mathcal{K}}(G) := \Psi_{\mathcal{K}}(\mathsf{st}(\mathcal{K}, G))$. We say that the extension $\mathcal{T}_0 \subseteq \mathcal{T}_0 \cup \mathcal{K}$ is Ψ-local if it satisfies:

(Loc^{Ψ}) for every finite set G of ground clauses, $\mathcal{T}_0 \cup \mathcal{K} \cup G \models \bot$ iff $\mathcal{T}_0 \cup \mathcal{K}[\Psi_{\mathcal{K}}(G)] \cup G$ has no weak partial model in which all terms in $\Psi_{\mathcal{K}}(G)$ are defined.

(ELoc^{Ψ}) is defined analogously. In (Ψ-)local theories and extensions satisfying (ELoc^{Ψ}), hierarchical reasoning is possible.

Theorem 1 ([14,11]). *Let $\mathcal{K}$ be a set of clauses. Assume that $\mathcal{T}_0 \subseteq \mathcal{T}_1 = \mathcal{T}_0 \cup \mathcal{K}$ is a Ψ-local theory extension, and that for every finite set T of terms $\Psi_{\mathcal{K}}(T)$ is finite. For any set G of ground clauses, let $\mathcal{K}_0 \cup G_0 \cup \mathsf{Def}$ be obtained from $\mathcal{K}[\Psi_{\mathcal{K}}(G)] \cup G$ by flattening and purification[2]. Then the following are equivalent:*

(1) G is satisfiable w.r.t. $\mathcal{T}_1$.
(2) $\mathcal{T}_0 \cup \mathcal{K}[\Psi_{\mathcal{K}}(G)] \cup G$ has a partial model with all terms in $\mathsf{st}(\mathcal{K}, G)$ defined.
(3) $\mathcal{T}_0 \cup \mathcal{K}_0 \cup G_0 \cup \mathsf{Con}[G]_0$ has a (total) model, where

$$\mathsf{Con}[G]_0 = \{ \bigwedge_{i=1}^{n} c_i = d_i \rightarrow c = d \mid f(c_1, \ldots, c_n) = c, f(d_1, \ldots, d_n) = d \in \mathsf{Def} \}.$$

Theorem 1 allows us to transfer decidability and complexity results from the theory $\mathcal{T}_0$ to the theory $\mathcal{T}_1$:

Theorem 2 ([14]). *Assume that the extension $\mathcal{T}_0 \subseteq \mathcal{T}_1$ satisfies condition (Loc^{Ψ}) – where Ψ has the property that $\Psi(T)$ is finite for every finite T – and that every variable in any clause of $\mathcal{K}$ occurs below some function symbol from Σ_1. If testing satisfiability of ground clauses in $\mathcal{T}_0$ is decidable, then so is testing satisfiability of ground clauses in $\mathcal{T}_1$. Assume that the complexity of testing the satisfiability w.r.t. $\mathcal{T}_0$ of a set of ground clauses of size m can be described by a function $g(m)$. Let G be a set of $\mathcal{T}_1$-clauses such that $\Psi_{\mathcal{K}}(G)$ has size n. Then the complexity of checking the satisfiability of G w.r.t. $\mathcal{T}_1$ is of order $g(n^k)$, where k is the maximum number of free variables in a clause in $\mathcal{K}$ (but at least 2).*

[2] $\mathcal{K}[\Psi_{\mathcal{K}}(G)] \cup G$ can be flattened and purified by introducing, in a bottom-up manner, new constants c_t for subterms $t = f(g_1, \ldots, g_n)$ with $f \in \Sigma_1$, g_i ground $\Sigma_0 \cup \Sigma_c$-terms (where Σ_c is a set of constants which contains the constants introduced by flattening, resp. purification), together with corresponding definitions $c_t = t$. We obtain a set of clauses $\mathcal{K}_0 \cup G_0 \cup \mathsf{Def}$, where Def consists of ground unit clauses of the form $f(g_1, \ldots, g_n) = c$, where $f \in \Sigma_1$, c is a constant, $g_1, \ldots, g_n$ are ground $\Sigma_0 \cup \Sigma_c$-terms, and $\mathcal{K}_0$ and G_0 are $\Sigma_0 \cup \Sigma_c$-clauses. Flattening and purification preserve satisfiability and unsatisfiability w.r.t. total algebras, and w.r.t. partial algebras in which all ground subterms which are flattened are defined [14]. In what follows, we explicitly indicate the sorts of the constraints in Def by using indices, i.e. $\mathsf{Def} = \bigcup_{s \in S} \mathsf{Def}_s$.

Examples of Local Extensions. The locality of an extension can either be proved directly, or by proving embeddability of partial into total models.

Theorem 3 ([14,16,11,17]). *The following theory extensions are local:*

(1) Any extension of a theory with free function symbols;
(2) Extensions of any base theory $\mathcal{T}_0$ with functions satisfying axioms of the form

$$\mathsf{GBounded}(f) \qquad \bigwedge_{i=1}^{n}(\phi_i(\overline{x}) \rightarrow s_i \leq f(\overline{x}) \leq t_i)$$

where Π_0 contains a sort s for which a reflexive binary relation $\leq$ exists, s_i, t_i are Σ_0-terms of sort s and ϕ_i are Π_0-formulae s.t. for $i \neq j$, $\phi_i \wedge \phi_j \models_{\mathcal{T}_0} \bot$, and $\mathcal{T}_0 \models \forall \overline{x}(\phi_i(\overline{x}) \rightarrow s_i(\overline{x}) \leq t_i(\overline{x}))$.

3 Functions on Absolutely Free Data Structures

Let $\mathsf{AbsFree}_{\Sigma_0} = (\bigcup_{c\in\Sigma_0}(\mathsf{Inj}_c) \cup (\mathsf{Acyc}_c)) \cup \bigcup_{\substack{c,d\in\Sigma \\ c\neq d}} \mathsf{Disjoint}(c,d)$, where:

$$(\mathsf{Inj}_c) \qquad c(x_1,\ldots,x_n) = c(y_1,\ldots,y_n) \rightarrow \bigwedge_{i=1}^{n} x_i = y_i$$

$$(\mathsf{Acyc}_c) \qquad c(t_1,\ldots,t_n) \neq x \text{ if } x \text{ occurs in some } t_i$$

$$\mathsf{Disjoint}(c,d) \qquad c(x_1,\ldots,x_n) \neq d(y_1,\ldots,y_k) \qquad \text{if } c \neq d$$

Note that (Acyc_c) is an axiom schema (representing an infinite set of axioms).

Theorem 4. *The following theories are local:*

(a) The theory $\mathsf{AbsFree}_{\Sigma_0}$ of absolutely free constructors in Σ_0.
(b) Any theory $\mathsf{AbsFree}_{\Sigma_0\setminus\Sigma}$ obtained from $\mathsf{AbsFree}_{\Sigma_0}$ by dropping the acyclicity condition for a set $\Sigma \subseteq \Sigma_0$ of constructors.
(c) $\mathcal{T} \cup \mathsf{Sel}(\Sigma')$, where $\mathcal{T}$ is one of the theories in (a) or (b), and $\mathsf{Sel}(\Sigma') = \bigcup_{c\in\Sigma'}\bigcup_{i=1}^{n}\mathsf{Sel}(s_i^c, c)$ axiomatizes a family of selectors $s_1^c,\ldots,s_n^c$, where $n = a(c)$, corresponding to constructors $c \in \Sigma' \subseteq \Sigma_0$. Here,

$$\mathsf{Sel}(s_i, c) \qquad \forall x, x_1,\ldots,x_n \quad x = c(x_1,\ldots,x_n) \rightarrow s_i(x) = x_i.$$

In addition, $\mathcal{K} = \mathsf{AbsFree}_{\Sigma_0} \cup \mathsf{Sel}(\Sigma_0) \cup \mathsf{IsC}$, where

$$(\mathsf{IsC}) \quad \forall x \bigvee_{c\in\Sigma_0} x = c(s_1^c(x),\ldots,s_{a(c)}^c(x))$$

has the property that for every set G of ground $\Sigma_0 \cup \mathsf{Sel} \cup \Sigma_c$-clauses (where Σ_c is a set of additional constants), $\mathcal{K} \wedge G \models \bot$ iff $\mathcal{K}[\Psi(G)] \wedge G \models \bot$, where $\Psi(G) = \mathsf{st}(G) \cup \bigcup_{a\in\Sigma_c\cap\mathsf{st}(G)}\bigcup_{c\in\Sigma_0}(\{s_i^c(a) \mid 1\leq i\leq a(c)\}\cup\{c(s_1^c(a),\ldots,s_n^c(a))\})$.

Proof: This is proved by showing that every weak partial model of the axioms for (a)–(c) weakly embeds into a total model of the axioms. The locality then follows from the link between embeddability and locality established in [7]. □

The reduction to the pure theory of equality made possible by Theorem 4 is very similar to Oppen's method [13] for deciding satisfiability of ground formulae for

free recursive data structures by bi-directional closure. Quantifier elimination (cf. [13]) followed by the reduction enabled by Theorem 4 can be used to obtain a decision procedure for the first-order theory of absolutely free constructors axiomatized by $\mathsf{AbsFree}_{\Sigma_0} \cup \mathsf{Sel}(\Sigma_0) \cup \mathsf{IsC}$.

We consider extensions of $\mathsf{AbsFree}_{\Sigma_0}$ with new function symbols, possibly with codomain of a different sort, i.e. theories over the signature $S=\{d, s_1, \ldots, s_n\}$, where d is the "data" sort; we do not impose any restriction on the nature of the sorts in s_i (some may be equal to d). The function symbols are:

- constructors $c \in \Sigma$ (arity $d^n \rightarrow d$), and corresponding selectors s_i^c (arity $d \rightarrow d$);
- all functions Σ_{s_i} in the signature of the theory of sort s_i, for $i = 1, \ldots, n$;
- for every $1 \leq i \leq n$, a set Σ_i of functions of sort $d \rightarrow s_i$.

In what follows we will analyze certain such extensions for which decision procedures for ground satisfiability exist[3]. We assume for simplicity that $S = \{d, s\}$.

3.1 A Class of Recursively Defined Functions

Let $S = \{d, s\}$, where d is the "data" sort and s is a different sort (output sort for some of the recursively defined functions).

Let $\mathcal{T}_s$ be a theory of sort s. We consider extensions of the disjoint combination of $\mathsf{AbsFree}_{\Sigma_0}$ and $\mathcal{T}_s$ with functions in a set $\Sigma = \Sigma_1 \cup \Sigma_2$, where the functions in Σ_1 have arity $d \rightarrow d$ and those in Σ_2 have arity $d \rightarrow s$. If f has sort $d \rightarrow b$, with $b \in S$, we denote its output sort b by $o(f)$. Let $\Sigma_{o(f)}$ be Σ_0 if $o(f) = d$, or Σ_s if $o(f) = s$, and $\mathcal{T}_{o(f)}$ be the theory $\mathsf{AbsFree}_{\Sigma_0}$ if $o(f) = d$, or $\mathcal{T}_s$ if $o(f) = s$. For every $f \in \Sigma$ we assume that a subset $\Sigma_r(f) \subseteq \Sigma_0$ is specified (a set of constructors for which recursion axioms for f exist).

We consider theories of the form $\mathcal{T} = \mathsf{AbsFree}_{\Sigma_0} \cup \mathcal{T}_s \cup \mathsf{Rec}_\Sigma$, where $\mathsf{Rec}_\Sigma = \bigcup_{f \in \Sigma} \mathsf{Rec}_f$ is a set of axioms of the form:

$$\mathsf{Rec}_f \quad \left\{ \begin{array}{c} f(k) = k_f \\ f(c(x_1, \ldots, x_n)) = g^{c,f}(f(x_1), \ldots, f(x_n)) \end{array} \right.$$

where k, c range over all constructors in $\Sigma_r(f) \subseteq \Sigma_0$, with $a(k) = 0, a(c) = n$, k_f are ground $\Sigma_{o(f)}$-terms and the functions $g^{c,f}$ are expressible by $\Sigma_{o(f)}$-terms.

We also consider extensions with a new set of functions satisfying definitions by guarded recursion of the form $\mathsf{Rec}_\Sigma^g = \bigcup_{f \in \Sigma} \mathsf{Rec}_f^g$:

$$\mathsf{Rec}_f^g \quad \left\{ \begin{array}{rl} f(k) = k_f & \\ f(c(x_1, \ldots, x_n)) = & \left\{ \begin{array}{ll} g_1^{c,f}(f(x_1), \ldots, f(x_n)) & \text{if } \phi_1(f(x_1), \ldots, f(x_n)) \\ \ldots & \\ g_k^{c,f}(f(x_1), \ldots, f(x_n)) & \text{if } \phi_k(f(x_1), \ldots, f(x_n)) \end{array} \right. \end{array} \right.$$

[3] In this paper we only focus on the problem of checking the satisfiability of sets of ground clauses, although it appears that when adding axiom IsC decision procedures for larger fragments can be obtained using arguments similar to those used in [18].

where k, c range over all constructors in $\Sigma_r(f) \subseteq \Sigma_0$, with $a(k) = 0, a(c) = n$, k_f are ground $\Sigma_{o(f)}$-terms and the functions $g_i^{c,f}$ are expressible by $\Sigma_{o(f)}$-terms, and $\phi_i(x_1, \ldots, x_n)$ are $\Sigma_{o(f)}$-formulae with free variables $x_1, \ldots, x_n$, where $\phi_i \wedge \phi_j \models_{\mathcal{T}_{o(f)}} \bot$ for $i \neq j$.

Definition 1. *A definition of type* Rec_f *is* exhaustive *if* $\Sigma_r(f) = \Sigma_0$ *(i.e.* Rec_f *contains recursive definitions for terms starting with any* $c \in \Sigma_0$*). A definition of type* Rec_f^g *is* exhaustive *if* $\Sigma_r(f) = \Sigma_0$ *and for every definition, the disjoint guards* $\phi_1, \ldots, \phi_n$ *are exhaustive, i.e.* $\mathcal{T}_{o(f)} \models \forall \overline{x} \phi_1(\overline{x}) \vee \ldots \vee \phi_n(\overline{x})$. Quasi-exhaustive *definitions are defined similarly, by allowing that* $\Sigma_0 \setminus \Sigma_r(f)$ *may consist of constants.*

Example 5. *Let* $\Sigma_0 = \{c_0, c\}$ *with* $a(c_0) = 0, a(c) = n$*. Let* $\mathcal{T}_0 = \mathsf{AbsFree}_{\Sigma_0} \cup \mathcal{T}_s$ *be the disjoint, many-sorted combination of the theory* $\mathsf{AbsFree}_{\Sigma_0}$ *(sort* d*) and* $\mathcal{T}_{\mathsf{num}}$*, the theory of natural numbers with addition (sort* num*).*

(1) A size *function can be axiomatized by* $\mathsf{Rec}_{\mathsf{size}}$*:*

$$\begin{cases} \mathsf{size}(c_0) = 1 \\ \mathsf{size}(c(x_1, \ldots, x_n)) = 1 + \mathsf{size}(x_1) + \cdots + \mathsf{size}(x_n) \end{cases}$$

(2) A depth *function can be axiomatized by the following definition* $\mathsf{Rec}_{\mathsf{depth}}^g$ *(of type* Rec^g *due to* max*):*

$$\begin{cases} \mathsf{depth}(c_0) = 1 \\ \mathsf{depth}(c(x_1, \ldots, x_n)) = 1 + \mathsf{max}\{\mathsf{depth}(x_1), \ldots, \mathsf{depth}(x_n)\} \end{cases}$$

Example 6. *Let* $\Sigma_0 = \{c_0, d_0, c\}$ *with* $a(c_0) = a(d_0) = 0, a(c) = n$*, and let* $\mathcal{T}_0 = \mathsf{AbsFree}_{\Sigma_0} \cup \mathsf{Bool}$ *be the disjoint combination of the theories* $\mathsf{AbsFree}_{\Sigma_0}$ *(sort* d*) and* Bool*, having as model the two-element Boolean algebra* $\mathsf{B}_2 = (\{\mathsf{t}, \mathsf{f}\}, \sqcap, \sqcup, \neg)$ *(sort* bool*) with a function* has_{c_0} *with output of sort* bool*, defined by* $\mathsf{Rec}_{\mathsf{has}_{c_0}}$*:*

$$\begin{cases} \mathsf{has}_{c_0}(c_0) = \mathsf{t} \\ \mathsf{has}_{c_0}(d_0) = \mathsf{f} \\ \mathsf{has}_{c_0}(c(x_1, \ldots, x_n)) = \bigsqcup_{i=1}^n \mathsf{has}_{c_0}(x_i) \end{cases} \quad (\textstyle\bigsqcup \textit{ is the supremum operation in } \mathsf{B}_2).$$

Problem. We analyze the problem of testing satisfiability of conjunctions G of ground unit $\Sigma_0 \cup \Sigma_1 \cup \Sigma_2 \cup \Sigma_c$-clauses, where Σ_c is a set of new constants:

$$(\mathsf{AbsFree}_{\Sigma_0} \cup \mathcal{T}_s \cup \mathsf{Rec}_{\Sigma_1}^{[g]} \cup \mathsf{Rec}_{\Sigma_2}^{[g]}) \wedge G \models \bot$$

(If $\Sigma_2 = \emptyset$, $\mathcal{T}_s$ can be omitted.) In what follows we use the abbreviations $\Sigma = \Sigma_1 \cup \Sigma_2$, $\mathsf{Rec}_\Sigma^g = \mathsf{Rec}_{\Sigma_1}^g \cup \mathsf{Rec}_{\Sigma_2}^g$, and $\mathsf{Rec}_\Sigma = \mathsf{Rec}_{\Sigma_1} \cup \mathsf{Rec}_{\Sigma_2}$.

The form of the ground formulae to be considered can be simplified as follows:

Lemma 7. *For every set* G *of ground unit* $\Sigma_0 \cup \Sigma \cup \Sigma_c$*-clauses there exists a set* G' *of* Σ*-flat ground unit* $\Sigma_0 \cup \Sigma \cup \Sigma_c'$*-clauses (where* $\Sigma_c \subseteq \Sigma_c'$*) of the form*

$$G' = C_s \wedge C_{\Sigma_0} \wedge C_\Sigma \wedge NC_{\Sigma_c'},$$

where C_s *is a set of (unit)* Σ_s*-clauses (if* $\Sigma_2 \neq \emptyset$*) and* $C_{\Sigma_0}, C_\Sigma, NC_{\Sigma_c'}$ *are (possibly empty) conjunctions of literals of the form:*

C_{Σ_0}: $c = c'$ *and* $c \neq c$, *where* $c, c' \in \Sigma_0$, *nullary;*
C_{Σ}: $(\neg)f(t_d) = t'$, *where* $f \in \Sigma_1 \cup \Sigma_2$, t_d *is a* $\Sigma_0 \cup \Sigma'_c$*-term,* t' *a* $\Sigma_{o(f)} \cup \Sigma'_c$*-term;*
$(\neg)f(t_d) = f'(t'_d)$, *where* $f, g \in \Sigma_2$, *and* t_d, t'_d *are* $\Sigma_0 \cup \Sigma'_c$*-terms;*
$NC_{\Sigma'_c}$: $t_d \neq t'_d$, *where* t_d, t'_d *are* $\Sigma_0 \cup \Sigma'_c$*-terms;*

such that G *and* G' *are equisatisfiable w.r.t.* $\mathsf{AbsFree}_{\Sigma_0} \cup \mathcal{T}_s \cup \mathcal{K}$ *for any set of clauses* $\mathcal{K}$ *axiomatizing the properties of the functions in* Σ.

Remark 8. *If* $\mathcal{K} = \mathsf{Rec}_\Sigma$ *we can ensure that, for every literal in* C_Σ, t_d (t'_d) *either starts with a constructor* $c \notin \Sigma_r(f)$ (*resp.* $c \notin \Sigma_r(f')$) *or is equal to some* $a \in \Sigma'_c$. *If the definition of* $f \in \Sigma$ *is exhaustive (resp. quasi-exhaustive), we can ensure that the only occurrence of* f *in* G' *is at the root of a term, in terms of the form* $f(a)$, *where* $a \in \Sigma_c$ (*resp., if* Rec_f *is quasi-exhaustive,* $a \in \Sigma_c \cup (\Sigma_0 \backslash \Sigma_r(f))$). *We can ensure that each such* $f(a)$ *occurs in at most one positive clause by replacing any conjunction* $f(a) = t_1 \wedge f(a) = t_2$ *with* $f(a) = t_1 \wedge t_1 = t_2$. $f(a) = t_1 \wedge f(a) \neq t_2$ *can also be replaced with the (equisatisfiable) conjunction:* $f(a) = t_1 \wedge t_1 \neq t_2$.

We make the following assumptions:

Assumption 1: Either $\Sigma_1 = \emptyset$, or else $\Sigma_1 \neq \emptyset$ and Rec_{Σ_1} is quasi-exhaustive.
Assumption 2: G is a set of ground unit clauses with the property that any occurrence of a function symbol in Σ_1 is in positive unit clauses of G of the form $f(a) = t$, with $a \in \Sigma_c \cup (\Sigma_0 \backslash \Sigma_r(f))$, and G does not contain any equalities between $\Sigma_0 \cup \Sigma_c$-terms. (By Remark 8, we can assume w.l.o.g. that for all $f \in \Sigma_1$ and $a \in \Sigma_c \cup (\Sigma_0 \backslash \Sigma_r(f))$, $f(a)$ occurs in at most one positive unit clause of G of the form $f(a) = t$.)

Theorem 9. *If Assumption 1 holds, then:*

(1) $\mathsf{AbsFree}_{\Sigma_0} \cup \mathcal{T}_s \cup \mathsf{Rec}_{\Sigma_2}$ *is a* Ψ*-local extension of* $\mathsf{AbsFree}_{\Sigma_0} \cup \mathcal{T}_s$;
(2) If Rec_{Σ_1} *is quasi-exhaustive, then* $\mathsf{AbsFree}_{\Sigma_0} \cup \mathcal{T}_s \cup \mathsf{Rec}_{\Sigma_1} \cup \mathsf{Rec}_{\Sigma_2}$ *satisfies the* Ψ*-locality conditions of an extension of* $\mathsf{AbsFree}_{\Sigma_0} \cup \mathcal{T}_s$ *for every set* G *of unit clauses which satisfy Assumption 2;*

where Ψ *associates with any set* T *of ground terms the smallest set which contains* T *and if* $f(c(t_1, \ldots, t_n)) \in \Psi(T)$ *and* $c \in \Sigma_r(f)$ *then* $f(t_i) \in \Psi(T)$ *for* $i = 1, \ldots, n$.

Similar results hold for extensions with Rec^g_Σ *(under similar assumptions) provided the guards* ϕ_i *in the recursive definitions of functions in* Σ_1 *are positive. The results can even be extended to recursive definitions of the form* $\mathsf{ERec}^{[g]}_f$:

$$\begin{cases} f(k, x) = k_f(x) \\ f(c(x_1, \ldots, x_n), x) = \begin{cases} g_1^{c,f}(f(x_1, x), \ldots, f(x_n, x), x) & \text{if } \phi_1(f(x_1), \ldots, f(x_n)) \\ \ldots \\ g_k^{c,f}(f(x_1, x), \ldots, f(x_n, x), x) & \text{if } \phi_k(f(x_1), \ldots, f(x_n)) \end{cases} \end{cases}$$

where k, c *range over* $\Sigma_r(f)$, $a(k) = 0, a(c) = n$, $k_f(x)$ *are* $\Sigma_{o(f)}$*-terms with free variable* x, $g_i^{c,f}$ *are functions expressible as* $\Sigma_{o(f)}$*-terms, and* $\phi_i(x_1, \ldots, x_n)$ *are* $\Sigma_{o(f)}$*-formulae with free variables* $x_1, \ldots, x_n$, *s.t.* $\phi_i \wedge \phi_j \models_{\mathcal{T}_{o(f)}} \bot$ *for* $i \neq j$.

Note: We can actually prove a variant of ELoc^{Ψ}, in which we can allow first-order Σ_s-constraints in $(\mathsf{E})\mathsf{Rec}^g_\Sigma$ and in G.

Example 10. *Let $\Sigma_0 = \{c_0, d_0, c\}$, where c is a binary constructor and c_0, d_0 are nullary. Consider the recursive definition $\mathsf{Rec}_{\mathsf{has}_{c_0}}$ of the function has_{c_0} in Example 6. We want to show that $\mathsf{AbsFree}_{\Sigma_0} \cup \mathsf{Bool} \cup \mathsf{Rec}_{\mathsf{has}_{c_0}} \models G_1$ where*

$$G_1 = \forall \overline{x}(\mathsf{has}_{c_0}(x){=}\mathsf{t} \wedge z_1{=}c(y_1, c(x_1, x)) \wedge z_1{=}c(y_2, y_3) \rightarrow \mathsf{has}_{c_0}(y_3){=}\mathsf{t})$$
$$G = \neg G_1 = (\mathsf{has}_{c_0}(a){=}\mathsf{t} \wedge c_1{=}c(b_1, c(a_1, a)) \wedge c_1{=}c(b_2, b_3) \wedge \mathsf{has}_{c_0}(b_3){=}\mathsf{f}),$$

where $\Sigma_c = \{a, a_1, b_1, b_2, b_3, c_1\}$. We transform G as explained in Lemma 7 by inferring all equalities entailed by the equalities between constructor terms in G; if $a_i = a_j$ (resp. $a_i = c(a_1, \ldots, a_n)$) is entailed we replace a_i with a_j (resp. with $c(a_1, \ldots, a_n)$). We obtain the equisatisfiable set of ground clauses:

$$G' = (\mathsf{has}_{c_0}(a){=}\mathsf{t} \wedge \mathsf{has}_{c_0}(c(a_1, a)){=}\mathsf{f}).$$

$(\mathsf{AbsFree}_{\Sigma_0} \cup \mathsf{Bool} \cup \mathsf{Rec}_{\mathsf{has}_{c_0}}) \cup G' \models \bot$ *iff* $(\mathsf{AbsFree}_{\Sigma_0} \cup \mathsf{Bool}) \cup \mathsf{Rec}_{\mathsf{has}_{c_0}}[\Psi(G')] \cup G' \models \bot$, *where* $\Psi(G') = \{\mathsf{has}_{c_0}(c(a_1, a)), \mathsf{has}_{c_0}(a_1), \mathsf{has}_{c_0}(a)\}$ *by Theorem 9. After purification we obtain:*

$\mathsf{Def}_{\mathsf{bool}}$	$G_0 \wedge \mathsf{Rec}_{\mathsf{has}_{c_0}}[\Psi(G)]_0$
$\mathsf{has}_{c_0}(a_1) = h_1 \wedge \mathsf{has}_{c_0}(a) = h_2 \wedge \mathsf{has}_{c_0}(c(a_1, a)) = h_3$	$h_2 = \mathsf{t} \wedge h_3 = \mathsf{f} \wedge h_3 = h_1 \sqcup h_2$

We immediately obtain a contradiction in Bool*, without needing to consider* Con_0 *or a further reduction to a satisfiability test w.r.t.* $\mathsf{AbsFree}_{\Sigma_0}$.

Combining Recursive Definitions with Boundedness. We analyze the locality of combinations of $\mathsf{Rec}^{[g]}_\Sigma$ with boundedness axioms, of the type:

$$\mathsf{Bounded}(f) \qquad \forall x(t_1 \leq f(x) \leq t_2)$$

Theorem 11. *Assume that $\leq$ is a partial order in all models of $\mathcal{T}_s$, $a(f) = d \rightarrow s$, t_1, t_2 are Σ_s-terms with $\mathcal{T}_s \models t_1 \leq t_2$, and all functions $g_i^{c,f}$ used in the definition of f have the property:*

$$\forall x_1, \ldots, x_n (\bigwedge_{i=1}^{n} t_1 \leq x_i \leq t_2 \rightarrow t_1 \leq g_i^{c,f}(x_1, \ldots, x_n) \leq t_2), \text{ where } n = a(c).$$

If Assumption 1 holds then $\mathsf{AbsFree}_{\Sigma_0} \cup \mathcal{T}_s \cup \mathsf{Rec}^{[g]}_f \cup \mathsf{Bounded}$ *is a Ψ-local extension of* $\mathsf{AbsFree}_{\Sigma_0} \cup \mathcal{T}_s$*, where Ψ is defined as in Theorem 9.*

Proof: The conditions on the functions $g_i^{c,f}$ ensure that in the completion process used in Theorem 9 the corresponding properties of f can be guaranteed. □

Example 12. *(1) We want to check whether* $\mathsf{AbsFree}_{\Sigma_0} \cup \mathbb{Z} \cup \mathsf{Rec}_{\mathsf{depth}}$ *entails*

$$G_1 = \forall x_1, x_2, x_3, x_4\ (\mathsf{depth}(x_1) \leq \mathsf{depth}(x_2) \wedge \mathsf{depth}(x_4) \leq \mathsf{depth}(x_3) \wedge x_4{=}c(x_2) \rightarrow \mathsf{depth}(d(x_1, e(x_2, c'))) \leq \mathsf{depth}(e(x_4, x_3))),$$

where Σ_0 contains the constructors c' (nullary), c (unary), and d, e (binary). By Ψ-locality, this can be reduced to testing the satisfiability of the following conjunction of ground clauses containing the additional constants:

$$\Sigma_c = \{a_1, a_2, a_3, a_4, d_1, d_2, d_3, d_4, e_1, e_2, e_3, g_1, g_2, g_3, c_2', d_2'\}$$

(below we present the flattened and purified form), where $G = \neg G_1$:

Def_d	$\mathsf{Def}_{\mathsf{num}}$	G_{0d}	$G_{0\mathsf{num}}$	$\mathsf{Rec}_{\mathsf{depth}}[\Psi(G)]_0$
$d(a_1, e_2) = e_1$	$\mathsf{depth}(a_i) = d_i (i = 1-4)$	$a_4 = c_2'$	$d_1 \leq d_2$	$g_1 = 1 + \mathsf{max}\{d_1, g_2\}$
$e(a_2, c') = e_2$	$\mathsf{depth}(e_i) = g_i (i = 1, 2, 3)$		$d_4 \leq d_3$	$g_2 = 1 + \mathsf{max}\{d_2, 1\}$
$e(a_4, a_3) = e_3$	$\mathsf{depth}(c_2') = d_2'$		$g_1 \not\leq g_3$	$g_3 = 1 + \mathsf{max}\{d_4, d_3\}$
$c(a_2) = c_2'$				$d_2' = 1 + d_2$

Let Con_0 consist of all the instances of congruence axioms for c, d, e and depth. $G_0 \cup \mathsf{Rec}_{\mathsf{depth}}[\Psi(G)]_0 \cup \mathsf{Con}_0$ is satisfiable in $\mathsf{AbsFree}_{\Sigma_0} \cup \mathbb{Z}$. A satisfying assignment is: $d_1 = d_2 = 0$ and $d_2' = d_4 = d_3 = 1$ (d_2' and d_4 need to be equal due to Con_0 because $c_2' = a_4$; and $d_4 \leq d_3$). $g_2 = 1 + \mathsf{max}\{0, 1\} = 2$, $g_1 = 1 + \mathsf{max}\{d_1, g_2\} = 3$ and $g_3 = 1 + \mathsf{max}\{d_4, d_3\} = 1 + d_4 = 2$. Thus, $\mathsf{AbsFree}_{\Sigma_0} \cup \mathbb{Z} \cup \mathsf{Rec}_{\mathsf{depth}} \not\models G_1$.

(2) We now show that $\mathsf{AbsFree}_{\Sigma_0} \cup \mathbb{Z} \cup \mathsf{Rec}_{\mathsf{depth}} \cup \mathsf{Bounded}(\mathsf{depth}) \models G_1$, where

$$\mathsf{Bounded}(\mathsf{depth}) \qquad \forall x(\mathsf{depth}(x) \geq 1).$$

By Theorem 11, we only need to consider the instances of $\mathsf{Bounded}(\mathsf{depth})$ containing terms in $\mathsf{Def}_{\mathsf{num}}$, i.e. the constraints $d_i \geq 1$ for $i \in \{1, \ldots, 4\}$; $g_i \geq 1$ for $i \in \{1, \ldots, 3\}$ and $d_2' \geq 1$. Con_0 can be used to derive $d_4 = d_2'$. We obtain:

$g_1 = 1 + \mathsf{max}\{d_1, g_2\} = 1 + \mathsf{max}\{d_1, 1 + \mathsf{max}\{d_2, 1\}\} = 1 + \mathsf{max}\{d_1, 1 + d_2\} = 2 + d_2$
$g_3 = 1 + \mathsf{max}\{d_4, d_3\} = 1 + d_3 \geq 1 + d_4 = 1 + d_2' = 2 + d_2.$
which together with $g_1 \not\leq g_3$ yields a contradiction.

3.2 Restricting to Term-Generated Algebras

The apparent paradox in the first part of Example 12 is due to the fact that the axiomatization of $\mathsf{AbsFree}_{\Sigma_0}$ makes it possible to consider models in which the constants in Σ_c are not interpreted as ground Σ_0-terms. We would like to consider only models for which the support A_d of sort d is the set $T_{\Sigma_0}(\emptyset)$ of ground Σ_0-terms (we will refer to them as *term generated models*)[4]. We will assume that the axiomatization of the recursive functions contains a family of constraints $\{C(a) \mid a \in \Sigma_c\}$ expressed in first order logic on the values the function needs to take on any element in Σ_c with the property:

(**TG**) $C(a)$ iff there exists $t \in T_{\Sigma_0}(\emptyset)$ such that for all $f \in \Sigma_2$, $f(a) = f(t)$.

[4] For expressing this, we can use axiom IsC (cf. Theorem 4) or the axiom used in [18]: $(\mathsf{IsConstr}) \forall x \bigvee_{c \in \Sigma_0} \mathsf{Is}_c(x)$ where $\mathsf{Is}_c(x) = \exists x_1, \ldots, x_n : x = c(x_1, \ldots, x_n)$.

Example 13. *Some examples are presented below:*

(1) Assume $\Sigma_2 = \{\mathsf{size}\}$ *(the size function over absolutely free algebras with set of constructors* $\{c_i \mid 1 \leq i \leq n\}$ *with arities* $a(c_i)$*). The following size constraints have the desired property (cf. also [18]):*

$$C(a) = \exists x_1, \ldots, x_n(\mathsf{size}(a) = (\sum_{i=1}^{n} a(c_i) * x_i) + 1).$$

To prove this, note that for every term t, $\mathsf{size}(t) = (\sum_{i=1}^{n} a(c_i) * n(c_i, t) + 1)$, *where* $n(c_i, t)$ *is the number of times* c_i *occurs in* t*. Thus, if there exists* t *such that* $\mathsf{size}(t) = \mathsf{size}(a)$*, then* $C(a)$ *is true. Conversely, if* $C(a)$ *is true* $\mathsf{size}(a) = \mathsf{size}(t)$ *for every term with* x_i *occurrences of the constructor* c_i *for* $i = 1, ..., n$.

(2) Consider the depth function (with output sort int*) over absolutely free algebras with set of constructors* $\{c_i \mid 1 \leq i \leq n\}$*. Then* $C(a) := \mathsf{depth}(a) \geq 1$.

In what follows we will assume that $\Sigma_1 = \emptyset$.

Theorem 14. *Assume that for every* $a \in \Sigma_c$*, a set* $C(a)$ *of constraints satisfying condition* **(TG)** *exists. Then* $\mathsf{AbsFree}_{\Sigma_0} \cup \mathcal{T}_s \cup \mathsf{Rec}^{[g]}_{\Sigma_2} \cup \bigcup_{a \in \Sigma_c} C(a)$ *is a* Ψ*-local extension of* $\mathsf{AbsFree}_c \cup \mathcal{T}_s$*, where* Ψ *is defined as in Theorem 9.*

Note: As in Theorem 9, we can prove, in fact, ELoc^{Ψ}-locality. Hence, the possibility that $C(a)$ may be a first-order formula of sort s is not a problem.

In order to guarantee that we test satisfiability w.r.t. term generated models, in general we have to add, in addition to the constraints $C(a)$, for every function symbol $f \in \Sigma_2$, additional counting constraints describing, for every $x \in A_s$, the maximal number of distinct terms t in $T_{\Sigma_0}(\emptyset)$ with $f(t) = x$. If Σ_0 contains infinitely many nullary constructors the number of distinct terms t in $T_{\Sigma_0}(\emptyset)$ with $f(t) = x$ is infinite, so no counting constraints need to be imposed.

Counting constraints are important if Σ_0 contains only finitely many nullary constructors and if the set G of ground unit clauses we consider contains negative (unit) $\Sigma_0 \cup \Sigma_c$-clauses. For the sake of simplicity, we here only consider sets G of unit ground clauses which contain only negative (unit) clauses of sort s.

Lemma 15. *Assume that* $\Sigma_1 = \emptyset$ *and for every* $a \in \Sigma_c$ *there exists a set* $C(a)$ *of constraints such that condition* **(TG)** *holds. The following are equivalent for any set* G *of unit* $\Sigma_0 \cup \Sigma_2 \cup \Sigma_c$*-clauses in which all negative literals have all sort* s*.*

(1) There exists a term-generated model $A = (T_{\Sigma_0}(\emptyset), A_s, \{f_A\}_{f \in \Sigma_2}, \{a_A\}_{a \in \Sigma_c})$ *of* $\mathsf{AbsFree}_{\Sigma_0} \cup \mathcal{T}_s \cup \mathsf{Rec}^{[g]}_{\Sigma_2}$ *and* G.

(2) There exists a model $F = (T_{\Sigma_0}(\Sigma_c), A_s, \{f_F\}_{f \in \Sigma_2}, \{a_F\}_{a \in \Sigma_c})$ *of* $\mathsf{AbsFree}_{\Sigma_0} \cup \mathcal{T}_s \cup \mathsf{Rec}^{[g]}_{\Sigma_2} \cup \bigcup_{a \in \Sigma_c} C(a)$ *and* G*, where for every* $a \in \Sigma_c$*,* $a_F = a$.

(3) There exists a model $A = (A_d, A_s, \{f_A\}_{f \in \Sigma_2}, \{a_A\}_{a \in \Sigma_c})$ *of* $\mathsf{AbsFree}_{\Sigma_0} \cup \mathcal{T}_s \cup \mathsf{Rec}^{[g]}_{\Sigma_2} \cup \bigcup_{a \in \Sigma_c} C(a)$ *and* G.

From Theorem 14 and Lemma 15 it follows that for every set G of ground unit clauses in which all negative (unit) clauses consist of literals of sort s, testing whether there exists a term-generated model of $\mathsf{AbsFree}_{\Sigma_0} \cup \mathcal{T}_s \cup \mathsf{Rec}^{[g]}_{\Sigma_2}$ and G can be done by computing $\mathsf{Rec}^{[g]}_{\Sigma_2}[\Psi(G)]$ and then reducing the problem hierarchically to a satisfiability test w.r.t. $\mathsf{AbsFree}_{\Sigma_0} \cup \mathcal{T}_s$.

Example 16. *Example 12 provides an example of a ground clause G for which $\mathsf{AbsFree}_{\Sigma_0} \cup \mathbb{Z} \cup \mathsf{Rec}_{\mathsf{depth}} \not\models G$, and $\mathsf{AbsFree}_{\Sigma_0} \cup \mathbb{Z} \cup \mathsf{Rec}_{\mathsf{depth}} \wedge \mathsf{Bounded}(\mathsf{depth}) \models G$. Example 12(2) shows that $\mathsf{AbsFree}_{\Sigma_0} \cup \mathbb{Z} \cup \mathsf{Rec}_{\mathsf{depth}} \cup \bigcup_{a \in Const(G)} C(a) \models G$, i.e. (by Lemma 15), G is true in every term-generated model of $\mathsf{AbsFree}_{\Sigma_0} \cup \mathbb{Z} \cup \mathsf{Rec}_{\mathsf{depth}}$.*

Similar results can be obtained if we relax the restriction on occurrences of negative clauses in G. If the set of nullary constructors in Σ_0 is infinite the extension is easy; otherwise we need to use equality completion and add counting constraints as done e.g. in [18] (assuming that there exist counting constraints expressible in first-order logic for the recursive definitions we consider).

4 More General Data Structures

We will now extend the results above to more general data structures. Consider a signature consisting of a set Σ_0 of constructors (including a set C of constants). Let E be an additional set of identities between Σ_0-terms.

Example 17. *Let $\Sigma_0 = \{c, c_0\}$, where c is a binary constructor and c_0 is a constant. We can impose that E includes one or more of the following equations:*

(A) $c(c(x,y),z) = c(x,c(y,z))$ *(associativity)*
(C) $c(x,y) = c(y,x)$ *(commutativity)*
(I) $c(x,x) = x$ *(idempotence)*
(N) $c(x,x) = c_0$ *(nilpotence)*

We consider many-sorted extensions of the theory defined by E with functions in $\Sigma = \Sigma_1 \cup \Sigma_2$, and sorts $S = \{d, s\}$, where the functions in Σ_1 have sort $d \to d$, those in Σ_2 have sort $d \to s$, and the functions in Σ satisfy additional axioms of the form Rec_Σ and ERec_Σ as defined in Section 3.1.[5] We therefore consider two-sorted theories of the form $E \cup \mathcal{T}_s \cup (\mathsf{E})\mathsf{Rec}_\Sigma$, where $\mathcal{T}_s$ is a theory of sort s. We make the following assumptions:

Assumption 3: We assume that:

(a) The equations in E only contain constructors c with $c \in \bigcap_{f \in \Sigma} \Sigma_r(f)$.
(b) For every $\forall \overline{x} \;\; t(\overline{x}) = s(\overline{x}) \in E$ and every $f \in \Sigma_1 \cup \Sigma_2$ let $t'(\overline{x})$ (resp. $s'(\overline{x})$) be the $\Sigma_{o(f)}$-term obtained by replacing every constructor $c \in \Sigma_0$ with the term-generated function[6] $g^{c,f}$. Then for every $f \in \Sigma_1$, $E \models \forall \overline{x} \;\; t'(\overline{x}) = s'(\overline{x})$, and for every $f \in \Sigma_2$, $\mathcal{T}_s \models \forall \overline{x} \;\; t'(\overline{x}) = s'(\overline{x})$.

[5] We restrict to unguarded recursive definitions of type Rec_Σ and ERec_Σ to simplify the presentation. Similar results can be obtained for definitions of the type Rec^g_Σ and ERec^g_Σ, with minor changes in Assumption 3.

[6] $g^{c,f}$ is the function (expressible as a $\Sigma_{o(f)}$-term) from the definition $f(c(x_1, \ldots, x_n)) = g^{c,f}(f(x_1), \ldots, f(x_n))$ in Rec_f.

Example 18. *Consider the extension of the theory of one binary associative and/or commutative function c with the size function defined as in Example 5(1). Then*

$$\mathsf{size}(c(x,y)) = g^c_{\mathsf{size}}(\mathsf{size}(x),\mathsf{size}(y)), \text{ where } g^c_{\mathsf{size}}(x,y) = 1+x+y.$$

Note that g^c_{size} is associative and commutative, so Assumption 3 holds.

$$g^c_{\mathsf{size}}(g^c_{\mathsf{size}}(x,y),z) = 1+(1+x+y)+z = 1+x+(1+y+z) = g^c_{\mathsf{size}}(x,g^c_{\mathsf{size}}(y,z));$$
$$g^c_{\mathsf{size}}(x,y) = 1+x+y = 1+y+x = g^c_{\mathsf{size}}(y,x).$$

Example 19. *Assume that Σ_0 only contains the binary constructor c satisfying a set E of axioms containing some of the axioms $\{(\mathbf{A}),(\mathbf{C}),(\mathbf{I})\}$ in Example 17. Let enc_k be a new function symbol (modeling encoding with key k) satisfying*

$$\mathsf{Rec}_{\mathsf{enc}} \qquad \mathsf{enc}_k(c(x,y)) = c(\mathsf{enc}_k(x),\mathsf{enc}_k(y)).$$

It is easy to see that $g^c_{\mathsf{enc}} = c$ and hence Assumption 3 is satisfied.

In what follows we assume that Assumption 3 holds, and that Rec_{Σ_1} is quasi-exhaustive. Note that in the presence of axioms such as associativity, the universal (Horn) theory of E itself may be undecidable. We will therefore only consider the simpler proof task of checking whether

$$E \cup [\mathsf{E}]\mathsf{Rec}^{[\mathsf{g}]}_{\Sigma_1} \cup [\mathsf{E}]\mathsf{Rec}^{[\mathsf{g}]}_{\Sigma_2} \models G_1,$$

where G_1 is a ground $\Sigma \cup \Sigma_1 \cup \Sigma_2$-clause of the form

$$\bigwedge_{k=1}^{l} g_k(c_k) = t^d_k \wedge \bigwedge_{i=1}^{n} f_i(t^d_i) = t^s_i \wedge \bigwedge_{j=1}^{m} f_j(t^d_j) = f'_j(t'^d_j) \rightarrow f(t_d) = t_s \qquad (1)$$

where $g_k \in \Sigma_1$, $c_k \in \Sigma_0 \backslash \Sigma_r(g_k)$, f_i, f'_i, f are functions in Σ_2 (with output sort s different from d), t^d_k, t'^d_k, t_d are ground Σ_0-terms, and t^s_k, t'^s_k, t_s are Σ_s-terms. We additionally assume that for every $g \in \Sigma_1$ and every $c \in \Sigma_0 \backslash \Sigma_r(g)$, $g(c)$ occurs at most once in the premise of G.

Remark. If Rec_{Σ_2} is quasi-exhaustive, G is equisatisfiable with a clause in which every occurrence of $f \in \Sigma_2$ is in a term of the form $f(c)$, with $c \in \Sigma_0 \backslash \Sigma_r(f)$.

Theorem 20. *Assume that Rec_{Σ_1}, Rec_{Σ_2} are quasi-exhaustive and Assumption 3 holds. The following are equivalent for any set G of $\Sigma_0 \cup \Sigma$-clauses of form (1):*

(1) $E \cup \mathsf{Rec}_{\Sigma_1} \cup \mathcal{T}_s \cup \mathsf{Rec}_{\Sigma_2} \models G$.
(2) G is true in all models $A = (A_d, A_s, \{f_A\}_{f\in\Sigma})$ of $E \cup \mathsf{Rec}_{\Sigma_1} \cup \mathcal{T}_s \cup \mathsf{Rec}_{\Sigma_2}$.
(3) G is true in all models $F = (T_{\Sigma_0}(\emptyset)/\equiv_E, A_s, \{f_A\}_{f\in\Sigma})$ of $E \cup \mathcal{T}_s \cup \mathsf{Rec}_{\Sigma_1} \cup \mathsf{Rec}_{\Sigma_2}$.
(4) G is true in all weak partial models $F = (T_{\Sigma_0}(\emptyset)/\equiv_E, A_s, \{f_A\}_{f\in\Sigma})$ of $E \cup \mathcal{T}_s \cup (\mathsf{Rec}_{\Sigma_1} \cup \mathsf{Rec}_{\Sigma_1})[\Psi(G)]$ in which all terms in $\Psi(G)$ are defined.

Similar results can also be obtained for definitions of type Rec^q_Σ or $\mathsf{ERec}^{[g]}_\Sigma$.

Note: We can impose boundedness conditions on the recursively defined functions without affecting locality (as for absolutely free constructors). [7]

4.1 An Example Inspired from Cryptography

In this section we illustrate the ideas on an example inspired by the treatment of a Dolev-Yao security protocol considered in [4] (cf. also Examples 17 and 19). Let $\Sigma_0 = \{c\} \cup C$, where c is a binary constructor, and let enc be a binary function. We analyze the following situations:

(1) c satisfies a set E of axioms and enc is a free binary function. By Theorem 3, the extension of E with the free function enc is a local extension of E.
(2) c is an absolutely free constructor, and enc satisfies the recursive definition:

$$(\mathsf{ERec}_{\mathsf{enc}}) \quad \forall x, y, z \;\; \mathsf{enc}(c(x, y), z) = c(\mathsf{enc}(x, z), \mathsf{enc}(y, z)).$$

By Theorem 9, the extension $\mathsf{AbsFree}_c \subseteq \mathsf{AbsFree}_c \cup \mathsf{ERec}_{\mathsf{enc}}$ satisfies the Ψ-locality condition for all clauses satisfying Assumption 2 (with Ψ as in Theorem 9).
(3) If c is associative (resp. commutative) and enc satisfies axiom $\mathsf{ERec}_{\mathsf{enc}}$ then Assumption 3 is satisfied, so, by Theorem 20, $E \cup \mathsf{ERec}_{\mathsf{enc}}$ satisfies the condition of a Ψ-local extension of E for all clauses of type (1).

Formalizing the Intruder Deduction Problem. We now formalize the version of the deduction system of the Dolev and Yao protocol given in [4]. Let E be the set of identities which specify the properties of the constructors in Σ_0. We use the following chain of successive theory extensions:

$$E \subseteq E \cup \mathsf{ERec}_{\mathsf{enc}} \subseteq E \cup \mathsf{ERec}_{\mathsf{enc}} \cup \mathsf{Bool} \cup \mathsf{Rec}^g_{\mathsf{known}},$$

where known has sort $d \to \mathsf{bool}$ and $\mathsf{Rec}^g_{\mathsf{known}}$ consists of the following axioms:

$$\begin{array}{ll} \forall x, y & \mathsf{known}(c(x, y)) = \mathsf{known}(x) \sqcap \mathsf{known}(y) \\ \forall x, y & \mathsf{known}(y) = \mathsf{t} \to \mathsf{known}(\mathsf{enc}(x, y)) = \mathsf{known}(x) \end{array}$$

Intruder deduction problem. The general statement of the intruder deduction problem is: "Given a finite set T of messages and a message m, is it possible to retrieve m from T?".

Encoding the intruder deduction problem. The finite set of known messages, $T = \{t_1, \ldots, t_n\}$, where t_i are ground $\Sigma_0 \cup \{\mathsf{enc}\}$-terms, is encoded as $\bigwedge_{i=1}^n \mathsf{known}(t_i) = \mathsf{t}$. With this encoding, the intruder deduction problem becomes:

"Test whether $E \cup \mathsf{Rec}_{\mathsf{enc}} \cup \mathsf{Bool} \cup \mathsf{Rec}_{\mathsf{known}} \models \bigwedge_{i=1}^n \mathsf{known}(t_i) = \mathsf{t} \to \mathsf{known}(m) = \mathsf{t}$."

[7] We can also consider axioms which link the values of functions $f_2 \in \Sigma_2$ and $f_1 \in \Sigma_1$ on the constants, such as e.g. "$f_2(f_1(c)) = t_s$" if we consider clauses G in which if $f_1(c) = t$ occurs then $t = c'$, where c' is a constant constructor not in $\Sigma_r(f_2)$. In the case of Σ_1-functions defined by ERec we can consider additional axioms of the form: $\phi(f_2(x)) \to f_2(f_1(c, x)) = t'_s$, where t'_s is a ground term of sort s either containing f_2 (and of the form $f_2(c')$) or a pure Σ_s-term.

Example 21. *We illustrate the hierarchical reasoning method we propose on the following example: Assume that* $E = \{(\mathbf{C})\}$ *and the intruder knows the messages* $c(a,b)$ *and* $\mathsf{enc}(c(c(e,f),e),c(b,a))$*. We check if he can retrieve* $c(f,e)$*, i.e. if*

$$G : (\mathsf{known}(c(a,b))=\mathsf{t}) \wedge (\mathsf{known}(\mathsf{enc}(c(c(e,f),e),c(b,a)))=\mathsf{t}) \wedge (\mathsf{known}(c(f,e))=\mathsf{f})$$

is unsatisfiable w.r.t. $E \cup \mathsf{Bool} \cup \mathsf{ERec}_{\mathsf{enc}} \cup \mathsf{Rec}^{g}_{\mathsf{known}}$*. G is equisatisfiable with a set* G' *of clauses obtained by applying all the definitions in* $\mathsf{ERec}_{\mathsf{enc}}$ *and* $\mathsf{Rec}^{g}_{\mathsf{known}}$*:*

$$G' : \begin{array}{l}(\mathsf{known}(\mathsf{enc}(e,c(b,a))) \sqcap \mathsf{known}(\mathsf{enc}(f,c(b,a))) \sqcap \mathsf{known}(\mathsf{enc}(e,c(b,a)))=\mathsf{t}) \\ \wedge\ (\mathsf{known}(a) \sqcap \mathsf{known}(b)=\mathsf{t}) \ \wedge\ (\mathsf{known}(f) \sqcap \mathsf{known}(e)=\mathsf{f}).\end{array}$$

By Theorem 20, we know that $E \cup \mathsf{Rec}_{\mathsf{enc}} \cup \mathsf{Bool} \cup \mathsf{Rec}_{\mathsf{known}} \wedge G' \models \bot$ *iff* $E \cup \mathsf{Rec}_{\mathsf{enc}} \cup \mathsf{Bool} \cup \mathsf{Rec}_{\mathsf{known}}[\Psi(G')] \wedge G' \models \bot$ *. The reduction is illustrated below:*

$\mathsf{Def}_{\mathsf{bool}}$		$G'_0 \quad \wedge$	$\mathsf{Rec}_{\mathsf{known}}[\Psi(G')]_0$
$k_1 = \mathsf{known}(a)$	$k_5 = \mathsf{known}(\mathsf{enc}(e,c(b,a)))$	$k_1 \sqcap k_2 = \mathsf{t}$	$k_7 = k_2 \sqcap k_1$
$k_2 = \mathsf{known}(b)$	$k_6 = \mathsf{known}(\mathsf{enc}(f,c(b,a)))$	$k_3 \sqcap k_4 = \mathsf{f}$	$k_7 = \mathsf{t} \rightarrow k_5 = k_3$
$k_3 = \mathsf{known}(e)$	$k_7 = \mathsf{known}(c(b,a))$	$k_5 \sqcap k_6 \sqcap k_5 = \mathsf{t}$	$k_7 = \mathsf{t} \rightarrow k_6 = k_4$
$k_4 = \mathsf{known}(f)$			

(We ignored Con_0*.) The contradiction in* Bool *can be detected immediately.*

5 Conclusion

We showed that many extensions with recursive definitions (which can be seen as generalized homomorphism properties) satisfy locality conditions. This allows us to reduce the task of reasoning about the class of recursive functions we consider to reasoning in the underlying theory of data structures (possibly combined with the theories attached to the co-domains of the additional functions). We illustrated the ideas on several examples (including one inspired from cryptography). The main advantage of the method we use consists in the fact that it has the potential of completely separating the task of reasoning about the recursive definitions from the task of reasoning about the underlying data structures. We believe that these ideas will make the automatic verification of certain properties of recursive programs or of cryptographic protocols much easier, and we plan to make a detailed study of applications to cryptography in future work. An implementation of the method for hierarchical reasoning in local theory extensions is available at `www.mpi-inf.mpg.de/~ihlemann/software/index.html` (cf. also [12]). In various test runs it turned out to be extremely efficient, and can be used as a decision procedure for local theory extensions. We plan to extend the program to handle the theory extensions considered in this paper; we expect that this will not pose any problems. There are other classes of bridging functions – such as, for instance, cardinality functions for finite sets and measure functions for subsets of $\mathbb{R}$ (for instance intervals) – which turn out to satisfy similar locality properties. We plan to present such phenomena in a separate paper.

Acknowledgments. Many thanks to the referees for their helpful comments. This work was partly supported by the German Research Council (DFG) as part of the Transregional Collaborative Research Center "Automatic Verification and Analysis of Complex Systems" (SFB/TR 14 AVACS, `www.avacs.org`).

References

1. Armando, A., Ranise, S., Rusinowitch, M.: A rewriting approach to satisfiability procedures. Information and Computation 183(2), 140–164 (2003)
2. Barrett, C., Shikanian, I., Tinelli, C.: An abstract decision procedure for satisfiability in the theory of inductive data types. Journal on Satisfiability, Boolean Modeling and Computation 3, 1–17 (2007)
3. Bonacina, M.P., Echenim, M.: Rewrite-based decision procedures. Electronic Notes in Theoretical Computer Science 174(11), 27–45 (2007)
4. Comon-Lundh, H., Treinen, R.: Easy intruder deductions. In: Dershowitz, N. (ed.) Verification: Theory and Practice. LNCS, vol. 2772, pp. 225–242. Springer, Heidelberg (2004)
5. Comon-Lundh, H.: Challenges in the automated verification of security protocols. In: Armando, A., Baumgartner, P., Dowek, G. (eds.) IJCAR 2008. LNCS, vol. 5195, pp. 396–409. Springer, Heidelberg (2008)
6. Delaune, S.: Easy intruder deduction problems with homomorphisms. Information Processing Letters 97(6), 213–218 (2006)
7. Ganzinger, H.: Relating semantic and proof-theoretic concepts for polynomial time decidability of uniform word problems. In: Sixteenth Annual IEEE Symposium on Logic in Computer Science, Boston, MA, USA, pp. 81–90. IEEE Computer Society, Los Alamitos (2001)
8. Ganzinger, H., Sofronie-Stokkermans, V., Waldmann, U.: Modular proof systems for partial functions with Evans equality. Information and Computation 204(10), 1453–1492 (2006)
9. Givan, R., McAllester, D.: New results on local inference relations. In: Principles of Knowledge Representation and reasoning: Proceedings of the Third International Conference (KR 1992), pp. 403–412. Morgan Kaufmann Press, San Francisco (1992)
10. Givan, R., McAllester, D.A.: Polynomial-time computation via local inference relations. ACM Transactions on Computational Logic 3(4), 521–541 (2002)
11. Ihlemann, C., Jacobs, S., Sofronie-Stokkermans, V.: On local reasoning in verification. In: Ramakrishnan, C.R., Rehof, J. (eds.) TACAS 2008. LNCS, vol. 4963, pp. 265–281. Springer, Heidelberg (2008)
12. Ihlemann, C., Sofronie-Stokkermans, V.: System description. H-PiLOT. In: Schmidt, R.A. (ed.) CADE 2009. LNCS (LNAI), vol. 5663, pp. 131–139. Springer, Heidelberg (2009)
13. Oppen, D.C.: Reasoning about recursively defined data structures. Journal of the ACM 27(3), 403–411 (1980)
14. Sofronie-Stokkermans, V.: Hierarchic reasoning in local theory extensions. In: Nieuwenhuis, R. (ed.) CADE 2005. LNCS (LNAI), vol. 3632, pp. 219–234. Springer, Heidelberg (2005)
15. Sofronie-Stokkermans, V.: Hierarchical and modular reasoning in complex theories: The case of local theory extensions. In: Konev, B., Wolter, F. (eds.) FroCos 2007. LNCS, vol. 4720, pp. 47–71. Springer, Heidelberg (2007)

16. Sofronie-Stokkermans, V., Ihlemann, C.: Automated reasoning in some local extensions of ordered structures. Journal of Multiple-Valued Logics and Soft Computing 13(4-6), 397–414 (2007)
17. Sofronie-Stokkermans, V.: Efficient hierarchical reasoning about functions over numerical domains. In: Dengel, A.R., Berns, K., Breuel, T.M., Bomarius, F., Roth-Berghofer, T.R. (eds.) KI 2008. LNCS (LNAI), vol. 5243, pp. 135–143. Springer, Heidelberg (2008)
18. Zhang, T., Sipma, H., Manna, Z.: Decision procedures for term algebras with integer constraints. Information and Computation 204(10), 1526–1574 (2006)