On Local Reasoning in Verification

Carsten Ihlemann, Swen Jacobs, and Viorica Sofronie-Stokkermans

Max-Planck-Institut für Informatik, Campus E1 4, Saarbrücken, Germany
{ihlemann,sjacobs,sofronie}@mpi-inf.mpg.de

Abstract. We present a general framework which allows to identify complex theories important in verification for which efficient reasoning methods exist. The framework we present is based on a general notion of locality. We show that locality considerations allow us to obtain parameterized decidability and complexity results for many (combinations of) theories important in verification in general and in the verification of parametric systems in particular. We give numerous examples; in particular we show that several theories of data structures studied in the verification literature are local extensions of a base theory. The general framework we use allows us to identify situations in which some of the syntactical restrictions imposed in previous papers can be relaxed.

1 Introduction

Many problems in verification can be reduced to proving the satisfiability of conjunctions of literals in a background theory (which can be a standard theory, the extension of a theory with additional functions – free, monotone, or recursively defined – or a combination of theories). It is very important to identify situations where the search space can be controlled without losing completeness. Solutions to this problem were proposed in proof theory, algebra and verification: In [8,11], McAllester and Givan studied the proof-theoretical notion of "local inference systems" – where for proving/disproving a goal only ground instances of the inference rules are needed which contain ground terms which appear in the goal to be proved. In universal algebra, Burris [3] established a link between PTIME decidability of the uniform word problem in quasi-varieties of algebras and embeddability of partial into total models. A link to the notion of locality was established by Ganzinger [5]. In the verification literature, locality properties were investigated in the context of reasoning in pointer data structures by McPeak, Necula [12] and in the study of fragments of the theory of arrays by Bradley, Manna and Sipma [1] and Ghilardi, Nicolini, Ranise and Zucchelli [7]. The applications in verification usually require reasoning in complex domains. In [6,13] we study *local extensions of theories* and show that in such extensions proof tasks can be reduced, hierarchically, to proof tasks in the base theory.

The main contributions of this paper can be described as follows:

(1) We introduce generalized notions of locality and stable locality and show that theories important in verification (e.g. the theory of arrays in [1] and the theory of pointer structures in [12]) satisfy such locality conditions.

C.R. Ramakrishnan and J. Rehof (Eds.): TACAS 2008, LNCS 4963, pp. 265–281, 2008.

(2) We present a general framework which allows to identify local theories important in verification. This allows us to also handle fragments which do not satisfy all syntactical restrictions imposed in previous papers. In particular, the axiom sets which we consider may contain alternations of quantifiers.
(3) We use these results to give new examples of local theories of data types.
(4) We discuss the experiments we made with an implementation.

The paper is structured as follows. We start (Sect. 1.1 and 1.2) by discussing the application domains we consider and illustrating our main idea. Section 2 contains basic definitions. In Sect. 3 local extensions are defined, results on hierarchical reasoning, parameterized decidability and complexity results, and possibilities of recognizing local extensions are summarized. Section 4 contains a large number of examples, ranging from extensions with monotonicity, injectivity and (guarded) boundedness properties to theories of data structures (pointers, arrays). A general framework for recognizing locality in verification is presented in Sect. 5. We describe our implementation and some experiments in Sect. 6.

1.1 Application Domains

The application domains we consider are mainly related to the verification of parametric systems (parametric either w.r.t. the number of subsystems involved, or w.r.t. some data used to describe the states and their updates).

We model systems using transition constraint systems $T = (V, \Sigma, \mathsf{Init}, \mathsf{Update})$ which specify: the variables (V) and function symbols (Σ) whose values change over time; a formula Init specifying the properties of initial states; a formula Update with variables in $V \cup V'$ and function symbols in $\Sigma \cup \Sigma'$ (where V' and Σ' are copies of V resp. Σ, denoting the variables resp. functions after the transition) which specifies the relationship between the values of variables x and function symbols f before a transition and their values (x', f') after the transition. Such descriptions can be obtained from system specifications (for an example cf. [4]). With every specification, a *background theory* $\mathcal{T}_S$ – describing the data types used in the specification and their properties – is associated. The verification problems we consider are *invariant checking* and *bounded model checking*.

Invariant checking. We can check whether a formula Ψ is an inductive invariant of a transition constraint system $T=(V, \Sigma, \mathsf{Init}, \mathsf{Update})$ in two steps: (1) prove that $\mathcal{T}_S, \mathsf{Init} \models \Psi$; (2) prove that $\mathcal{T}_S, \Psi, \mathsf{Update} \models \Psi'$, where Ψ' results from Ψ by replacing each $x \in \mathcal{V}$ by x' and each $f \in \Sigma$ by f'. Failure to prove (2) means that Ψ is not an invariant, or Ψ is not inductive w.r.t. T.[1]

Bounded model checking. We check whether, for a fixed k, unsafe states are reachable in at most k steps. Formally, we check whether:

$$\mathcal{T}_S \wedge \mathsf{Init}_0 \wedge \bigwedge_{i=1}^{j} \mathsf{Update}_i \wedge \neg \Psi_j \models \bot \qquad \text{for all } 0 \leq j \leq k,$$

[1] Proving that Ψ is an invariant of the system in general requires to find a stronger formula Γ (i.e., $\mathcal{T}_S \models \Gamma \rightarrow \Psi$) and prove that Γ is an inductive invariant.

where Update_i is obtained from Update by replacing all variables $x \in V$ by x_i and any $f \in \Sigma$ by f_i, and all $x' \in V'$, $f' \in \Sigma'$ by x_{i+1}, f_{i+1}; Init_0 is Init with x_0 replacing $x \in V$ and f_0 replacing $f \in \Sigma$; Ψ_i is obtained from Ψ similarly.

We are interested in checking whether a safety property (expressed by a suitable formula) is an invariant, or holds for paths of bounded length, *for given instances of the parameters*, or *under given constraints on parameters*. We aim at identifying situations in which decision procedures exist. We will show that this is often the case, by investigating locality phenomena in verification. As a by-product, this will allow us to consider problems more general than usual tasks in verification, namely to *derive constraints between parameters* which guarantee safety. These constraints may also be used to solve optimization problems (maximize/minimize some of the parameters) such that safety is guaranteed.

1.2 Illustration

We illustrate the problems as well as our solution on the following example.[2] Consider a parametric number m of processes. The priorities associated with the processes (non-negative real numbers) are stored in an array p. The states of the processes – enabled (1) or disabled (0) are stored in an array a. At each step only the process with maximal priority is enabled, its priority is set to x and the priorities of the waiting processes are increased by y. This can be expressed with the following set of axioms which we denote by $\mathsf{Update}(a, p, a', p')$

$$\begin{array}{l}
\forall i(1 \le i \le m \wedge \quad (\forall j(1 \le j \le m \wedge j \ne i \rightarrow p(i) > p(j))) \longrightarrow a'(i) = 1) \\
\forall i(1 \le i \le m \wedge \quad (\forall j(1 \le j \le m \wedge j \ne i \rightarrow p(i) > p(j))) \longrightarrow p'(i) = x) \\
\forall i(1 \le i \le m \wedge \neg(\forall j(1 \le j \le m \wedge j \ne i \rightarrow p(i) > p(j))) \longrightarrow a'(i) = 0) \\
\forall i(1 \le i \le m \wedge \neg(\forall j(1 \le j \le m \wedge j \ne i \rightarrow p(i) > p(j))) \longrightarrow p'(i) = p(i) + y)
\end{array}$$

where x and y are considered to be parameters. We may need to check whether if at the beginning the priority list is injective, i.e. formula $(\mathsf{Inj})(p)$ holds:

$$\mathsf{Inj}(p) \quad \forall i, j(1 \le i \le m \wedge 1 \le j \le m \wedge i \ne j \rightarrow p(i) \ne p(j))$$

then it remains injective after the update, i.e. check the satisfiability of:

$$(\mathbb{Z} \cup \mathbb{R}_+ \cup \{0, 1\}) \wedge \mathsf{Inj}(p) \wedge \mathsf{Update}(a, p, a', p') \wedge 1 \le c \le m \wedge 1 \le d \le m \wedge c \ne d \wedge p'(c) = p'(d).$$

We may need to check satisfiability of the formula under certain assumptions on the values of x and y (for instance if $x = 0$ and $y = 1$), or to determine constraints on x and y for which the formula is (un)satisfiable.

Problem. The problem above is a satisfiability problem for a formula with (alternations of) quantifiers in a combination of theories. SMT provers heuristically compute ground instances of the problems, and return $\mathsf{unsatisfiable}$ if a contradiction is found, and $\mathsf{unknown}$ if no contradiction can be derived from these instances. It is important to find a set of ground instances which are sufficient for deriving a contradiction if one exists. [1] presents a fragment of the theory

[2] All the examples in this paper will address invariant checking only. Bounded model checking problems can be handled in a similar way.

of arrays for which this is possible. The formula above does not belong to this fragment: $\mathsf{Inj}(p)$ contains the premise $i \neq j$; $\mathsf{Update}(a, p, a', p')$ contains $\forall\exists$ axioms.

Idea. Let $\mathcal{T}_0$ be the many-sorted combination of the theory of integers (for indices), of real numbers (priorities), and $\{0, 1\}$ (enabled/disabled). We consider:

(i) The extension $\mathcal{T}_1$ of $\mathcal{T}_0$ with the functions $a : \mathbb{Z} \to \{0, 1\}$ (a free function) and $p : \mathbb{Z} \to \mathbb{R}_+$ satisfying $\mathsf{Inj}(p)$;
(ii) The extension $\mathcal{T}_2$ of $\mathcal{T}_1$ with the functions $a' : \mathbb{Z} \to \{0, 1\}$, $p' : \mathbb{Z} \to \mathbb{R}_+$ satisfying the update axioms $\mathsf{Update}(a, p, a', p')$.

We show that both extensions have a locality property which allows us to use determined instances of the axioms without loss of completeness; the satisfiability problem w.r.t. $\mathcal{T}_2$ can be hierarchically reduced to a satisfiability problem w.r.t. $\mathcal{T}_1$ and then to a satisfiability problem w.r.t. $\mathcal{T}_0$. The purpose of this paper is to show that we can do this in a systematic way in a large number of situations.

2 Preliminaries

We assume known standard definitions from first-order logic. (Logical) theories can be regarded as collections of formulae (i.e. can be described as the consequences of a set of axioms), as collections of models (the set of all models of a set of axioms, or concrete models such as $\mathbb{Z}$ or $\mathbb{R}$), or both. If $\mathcal{T}$ is a theory and ϕ, ψ are formulae, we say that $\mathcal{T} \wedge \phi \models \psi$ (written also $\phi \models_{\mathcal{T}} \psi$) if ψ is true in all models of $\mathcal{T}$ which satisfy ϕ. If $\mathcal{T} \wedge \phi \models \bot$ (where $\bot$ is false), there are no models of $\mathcal{T}$ which satisfy ϕ, i.e. ϕ is unsatisfiable w.r.t. $\mathcal{T}$. For the verification tasks mentioned above, efficient reasoning in certain theories, which depend on the specification of the systems under consideration, is extremely important.

Local theory extensions. We consider extensions $\mathcal{T}_0 \cup \mathcal{K}$ of a theory $\mathcal{T}_0$ with new sorts and new function symbols (called *extension functions*) satisfying a set $\mathcal{K}$ of (universally quantified) clauses. An extension $\mathcal{T}_0 \subseteq \mathcal{T}_0 \cup \mathcal{K}$ is *local* if satisfiability of a set G of clauses w.r.t. $\mathcal{T}_0 \cup \mathcal{K}$ only depends on $\mathcal{T}_0$ and those instances $\mathcal{K}[G]$ of $\mathcal{K}$ in which the terms starting with *extension functions* are in the set $\mathsf{st}(\mathcal{K}, G)$ of ground terms which already occur in G or $\mathcal{K}$ [13]. A weaker locality notion, namely *stable locality*, exists; it allows to restrict the search to the instances $\mathcal{K}^{[G]}$ of $\mathcal{K}$ in which the variables below extension functions are instantiated with Σ_0-terms generated from $\mathsf{st}(\mathcal{K}, G)$. These generalize the notion of *local theories* introduced by [8,11,9] resp. of locality and stable locality studied in [5]. In such extensions hierarchical reasoning is possible (cf. also Sect. 3.1).

Partial and total models. Local and stably local theory extensions can be recognized by proving embeddability of partial into total models [13,16]. Let $\Pi = (S, \Sigma, \mathsf{Pred})$ be an S-sorted signature where Σ is a set of function symbols and Pred a set of predicate symbols. In a *partial Π-structure* the function symbols may be partial (for definitions cf. [2]). If A is a partial structure and $\beta : X \to \mathcal{A}$ is a valuation we say that $(A, \beta) \models_w (\neg)P(t_1, \ldots, t_n)$ iff (a) $\beta(t_i)$ are all defined and their values are in the relationship $(\neg)P_A$; or (b) at least one of $\beta(t_i)$ is undefined.

This holds in particular for the equality relation. (A, β) *weakly satisfies a clause* C (notation: $(A, \beta) \models_w C$) if it satisfies at least one literal in C. A is a *weak partial model* of a set of clauses $\mathcal{K}$ if $(A, \beta) \models_w C$ for every valuation β and every clause C in $\mathcal{K}$. *(Evans) partial models* are defined similarly, with the following difference: $(A, \beta) \models t \approx s$ iff (a) $\beta(t)$ and $\beta(s)$ are both defined and equal; or (b) $\beta(s)$ is defined, $t = f(t_1, \ldots, t_n)$ and $\beta(t_i)$ is undefined for at least one of the direct subterms of t; or (c) both $\beta(s)$ and $\beta(t)$ are undefined.

3 Locality

As seen in Section 1.2, the axioms occurring in applications may contain alternations of quantifiers. To address this, we study the notion of *extended (stable) locality* (cf. also [13]). Let $\mathcal{T}_0$ be a theory with signature $\Pi_0 = (S_0, \Sigma_0, \mathsf{Pred})$, where S_0 is a set of sorts, Σ_0 a set of function symbols, and Pred a set of predicate symbols. We consider extensions $\mathcal{T}_1$ of $\mathcal{T}_0$ with new sorts and function symbols (i.e. with signature $\Pi = (S_0 \cup S_1, \Sigma_0 \cup \Sigma_1, \mathsf{Pred})$), satisfying a set $\mathcal{K}$ of axioms of the form $(\Phi(x_1, \ldots, x_n) \vee C(x_1, \ldots, x_n))$, where $\Phi(x_1, \ldots, x_n)$ is an *arbitrary first-order formula* in the base signature Π_0 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. We are interested in disproving closed formulae Σ in the extension Π^c of Π with new constants Σ_c.

Example 1. *Consider the example in Sect. 1.2. In modeling this problem we start from the disjoint combination $\mathcal{T}_0$ of integers, reals and Booleans with signature $\Pi_0 = (S_0, \Sigma_0, \mathsf{Pred})$, where $S_0 = \{\mathsf{int}, \mathsf{real}, \mathsf{bool}\}$ and Σ_0, Pred consist of the (many-sorted) combination of the signatures of the corresponding theories. In a first step, $\mathcal{T}_0$ is extended to $\mathcal{T}_1 = \mathcal{T}_0 \cup \mathsf{Inj}(p)$, with signature $\Pi_1 = (S_0, \Sigma_0 \cup \{a, p\}, \mathsf{Pred})$. $\mathsf{Inj}(p)$ is a clause. In a second step, $\mathcal{T}_1$ is extended to a theory $\mathcal{T}_2 = \mathcal{T}_1 \cup \mathsf{Update}(a, p, a', p')$ with signature $(S_0, \Sigma_0 \cup \{a, p\} \cup \{a', p'\}, \mathsf{Pred})$. The axioms in $\mathsf{Update}(a, p, a', p')$ are of the form $\phi(i) \vee C(i)$ and $\neg\phi(i) \vee D(i)$, where $\phi(i) = \forall j(1 \leq j \leq m \wedge j \neq i \rightarrow p(i) > p(j))$. (Thus it can be seen that the first two axioms in $\mathsf{Update}(a, p, a', p')$ contain a $\forall\exists$ quantifier alternation.)*

We can extend the notion of locality accordingly. We study extensions $\mathcal{T}_0 \subseteq \mathcal{T}_0 \cup \mathcal{K}$ as above satisfying the locality and stable locality conditions (ELoc, ESLoc):

(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 $\mathcal{T}_0 \cup \mathcal{K}[\Gamma] \cup \Gamma$ has no weak partial model in which all terms in $\mathsf{st}(\mathcal{K}, G)$ are defined.

Here $\mathcal{K}[\Gamma]$ consists of all instances of $\mathcal{K}$ in which the terms starting with extension functions are in the set $\mathsf{st}(\mathcal{K}, G)$ (defined in Sect. 2).

(ESLoc) 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 $\mathcal{T}_0 \cup \mathcal{K}^{[\Gamma]} \cup \Gamma$ has no partial model in which all terms in $\mathsf{st}(\mathcal{K}, G)$ are defined.

Here $\mathcal{K}^{[\Gamma]}$ consists of all instances of $\mathcal{K}$ in which the variables below a Σ_1-symbol are instantiated with Σ_0-terms generated from $\mathsf{st}(\mathcal{K}, G)$.

The problem with (ESLoc) is that the number of instances in $\mathcal{K}^{[\Gamma]}$ is finite only if the number of Σ_0-terms generated from $\mathsf{st}(\mathcal{K}, G)$ can be guaranteed to be finite, i.e. when $\Sigma_0 = \emptyset$ (in which case the size of $\mathcal{K}^{[\Gamma]}$ is polynomial in the size of $\mathsf{st}(\mathcal{K}, G)$) or when only finitely many non-equivalent Σ_0-terms (modulo $\mathcal{T}_0$) can be generated from a finite set of generators (then the size of $\mathcal{K}^{[\Gamma]}$ is polynomial in the number of such non-equivalent terms). To overcome these problems, we identify a family of conditions in between locality and stable locality.

Let Ψ be a function associating with a set $\mathcal{K}$ of axioms and a set of ground terms T a set $\Psi_{\mathcal{K}}(T)$ of ground terms such that (i) all ground subterms in $\mathcal{K}$ and T are in $\Psi_{\mathcal{K}}(T)$; (ii) for all sets of ground terms T, T' if $T \subseteq T'$ then $\Psi_{\mathcal{K}}(T) \subseteq \Psi_{\mathcal{K}}(T')$; (iii) Ψ is a closure operation, i.e. for all sets of ground terms T, $\Psi_{\mathcal{K}}(\Psi_{\mathcal{K}}(T)) \subseteq \Psi_{\mathcal{K}}(T)$; (iv) Ψ is compatible with any map h between constants, i.e. for any map $h : C \to C$, $\Psi_{\mathcal{K}}(\overline{h}(T)) = \overline{h}(\Psi_{\mathcal{K}}(T))$, where $\overline{h}$ is the unique extension of h to terms. Let $\mathcal{K}[\Psi_{\mathcal{K}}(G)]$ be the set of instances of $\mathcal{K}$ in which the extension terms are in $\Psi_{\mathcal{K}}(\mathsf{st}(\mathcal{K}, G))$, which here will be denoted by $\Psi_{\mathcal{K}}(G)$. We say that an extension $\mathcal{T}_0 \subseteq \mathcal{T}_0 \cup \mathcal{K}$ is Ψ-local if it satisfies condition (ELoc^{Ψ}):

(ELoc^{Ψ}) for every formula $\Gamma = \Gamma_0 \cup G$, where Γ_0 is a Π^c_0-sentence and G a finite set of ground Π^c-clauses, $\mathcal{T}_1 \cup \Gamma \models \bot$ iff $\mathcal{T}_0 \cup \mathcal{K}[\Psi_{\mathcal{K}}(G)] \cup \Gamma$ has no weak partial model in which all terms in $\Psi_{\mathcal{K}}(G)$ are defined.

If $\mathcal{K}$ consists of clauses and only satisfiability of sets G of ground clauses is considered we obtain a condition (Loc^{Ψ}) extending the notion (Loc) of locality in [13]. Ψ-stable locality (ESLoc^{Ψ}) can be defined replacing $\mathcal{K}[\Psi_{\mathcal{K}}(G)]$ by $\mathcal{K}^{[\Psi_{\mathcal{K}}(G)]}$.

3.1 Hierarchical Reasoning in Local Theory Extensions

Let $\mathcal{T}_0 \subseteq \mathcal{T}_1 = \mathcal{T}_0 \cup \mathcal{K}$ be a theory extension satisfying condition ($\mathsf{E(S)Loc}$) or ($\mathsf{E(S)Loc}^{\Psi}$). To check the satisfiability w.r.t. $\mathcal{T}_1$ of a formula $\Gamma = \Gamma_0 \cup G$, where Γ_0 is a Π^c_0-sentence and G is a set of ground Π^c-clauses, we proceed as follows:

Step 1: By the locality assumption, $\mathcal{T}_1 \cup \Gamma_0 \cup G$ is satisfiable iff $\mathcal{T}_0 \cup \mathcal{K}*[G] \cup \Gamma_0 \cup G$ has a (weak) partial model with corresponding properties, where, depending on the type of locality, $\mathcal{K}*[G]$ is $\mathcal{K}[G], \mathcal{K}^{[G]}, \mathcal{K}[\Psi_{\mathcal{K}}(G)]$ or $\mathcal{K}^{[\Psi_{\mathcal{K}}(G)]}$.

Step 2: Purification. We purify $\mathcal{K}*[G] \cup G$ by introducing, in a bottom-up manner, new constants c_t (from a set Σ_c of constants) for subterms $t = f(g_1, \ldots, g_n)$ with $f \in \Sigma_1$, g_i ground $\Sigma_0 \cup \Sigma_c$-terms, together with their definitions $c_t \approx t$. The set of formulae thus obtained has the form $\mathcal{K}_0 \cup G_0 \cup \Gamma_0 \cup D$, where D consists of definitions of the form $f(g_1, \ldots, g_n) \approx c$, where $f \in \Sigma_1$, c is a constant, $g_1, \ldots, g_n$ are ground $\Sigma_0 \cup \Sigma_c$-terms, and $\mathcal{K}_0, G_0, \Gamma_0$ are Π^c_0-formulae.

Step 3: Reduction to testing satisfiability in $\mathcal{T}_0$. We reduce the problem to testing satisfiability in $\mathcal{T}_0$ by replacing D with the following set of clauses:

$$N_0 = \{ \bigwedge_{i=1}^{n} c_i \approx d_i \rightarrow c = d \mid f(c_1, \ldots, c_n) \approx c, f(d_1, \ldots, d_n) \approx d \in D \}.$$

This yields a sound and complete hierarchical reduction to a satisfiability problem in the base theory $\mathcal{T}_0$ (for ($\mathsf{E(S)Loc}^{\Psi}$) the proof is similar to that in [13]):

Theorem 1. *Let $\mathcal{K}$ and $\Gamma = \Gamma_0 \wedge G$ be as specified above. Assume that $\mathcal{T}_0 \subseteq \mathcal{T}_0 \cup \mathcal{K}$ satisfies condition* $(\mathsf{E(S)Loc})$ *or* $(\mathsf{E(S)Loc}^{\Psi})$*. Let $\mathcal{K}_0 \cup G_0 \cup \Gamma_0 \cup D$ be obtained from $\mathcal{K} * [G] \cup \Gamma_0 \cup G$ by purification, as explained above. The following are equivalent:*

(1) $\mathcal{T}_0 \cup \mathcal{K} * [G] \cup \Gamma_0 \cup G$ has a partial model with all terms in $\mathsf{st}(\mathcal{K}, G)$ defined.
(2) $\mathcal{T}_0 \cup \mathcal{K}_0 \cup G_0 \cup \Gamma_0 \cup D$ has a partial model with all extension terms in D defined.
(3) $\mathcal{T}_0 \cup \mathcal{K}_0 \cup G_0 \cup \Gamma_0 \cup N_0$ has a (total) model.

Alternatively, if $\mathcal{K}$ consists only of clauses and all variables occur below an extension function and if Γ is a set of ground clauses then $\mathcal{K} * [G] \wedge \Gamma$ consists of ground clauses, so locality also allows us to reduce reasoning in $\mathcal{T}_1$ to reasoning in an extension of $\mathcal{T}_0$ with free function symbols; an SMT procedure can be used. If Γ_0 contains quantifiers or $\mathcal{K} * [G]$ contains free variables it is problematic to use SMT provers without loss of completeness.

3.2 Decidability, Parameterized Complexity

Assume that $\mathcal{K}$ consists of axioms of the form $\overline{C} = (\Phi_C(\overline{x}) \vee C(\overline{x}))$, where $\Phi_C(\overline{x})$ is in a fragment (class of formulae) $\mathcal{F}$ of $\mathcal{T}_0$ and $C(\overline{x})$ is a Π-clause, and $\Gamma = \Gamma_0 \wedge G$, where Γ_0 is a formula in $\mathcal{F}$ without free variables, and G is a set of ground Π^c-clauses, both containing constants in Σ_c.

Theorem 2. *Assume that the theory extension $\mathcal{T}_0 \subseteq \mathcal{T}_1$ satisfies* $(\mathsf{E(S)Loc})$*, or* $(\mathsf{E(S)Loc}^{\Psi})$*. Satisfiability of goals $\Gamma_0 \cup G$ as above w.r.t. $\mathcal{T}_1$ is decidable provided $\mathcal{K} * [G]$ is finite and $\mathcal{K}_0 \cup G_0 \cup \Gamma_0 \cup N_0$ belongs to a decidable fragment of $\mathcal{T}_0$.*

Locality allows us to obtain parameterized decidability and complexity results:

Case 1: If for each $\overline{C} = \Phi_C(\overline{x}) \vee C(\overline{x}) \in \mathcal{K}$ all free variables occur below some extension symbol, then $\mathcal{K} * [G]$ contains only formulae of the form $\Phi_C(\overline{g}) \vee C(\overline{g})$, where $\overline{g}$ consists of ground Σ_0-terms, so $\mathcal{K}_0 \cup G_0 \cup \Gamma_0 \cup N_0 \in \mathcal{F}_g$, the class obtained by instantiating all free variables of formulae in $\mathcal{F}$ with ground Σ_0-terms.

Decidability and complexity: If checking satisfiability for the class $\mathcal{F}_g$ w.r.t. $\mathcal{T}_0$ is decidable, then checking satisfiability of goals of the form above w.r.t. $\mathcal{T}_1$ is decidable. Assume that the complexity of a decision procedure for the fragment $\mathcal{F}_g$ of $\mathcal{T}_0$ is $g(n)$ for an input of size n. Let m be the size of $\mathcal{K}_0 \cup G_0 \cup \Gamma_0 \cup N_0$. Then the complexity of proving satisfiability of $\Gamma_0 \cup G$ w.r.t. $\mathcal{T}_1$ is of order $g(m)$.

(i) For local extensions, $\mathcal{K} * [G] = \mathcal{K}[G]$; the size m of $\mathcal{K}_0 \cup G_0 \cup \Gamma_0 \cup N_0$ is of order $|G|^k$ for some $2 \leq k \in \mathbb{Z}$ for a fixed $\mathcal{K}$ (at least quadratic because of N_0).
(ii) For stably local extensions, the size of $\mathcal{K} * [G] = \mathcal{K}^{[G]}$ is polynomial in the size s of the model of $\mathcal{T}_0$ freely generated by $|\mathsf{st}(\mathcal{K}, G)|$ generators.

Similarly for Ψ-(stably) local extensions (with $\mathsf{st}(\mathcal{K}, G)$ replaced by $\Psi_{\mathcal{K}}(G)$).

Case 2: If not all free variables in $\mathcal{K}$ occur below an extension symbol, then the instances in $\mathcal{K} * [G]$ contain free variables, so $\mathcal{K}_0 \cup G_0 \cup \Gamma_0 \cup N_0$ is in the universal closure $\forall \mathcal{F}$ of $\mathcal{F}$. The decidability and complexity remarks above here apply relative to the complexity of checking satisfiability of formulae in the fragment $\forall \mathcal{F}$ of $\mathcal{T}_0$ with constants in Σ_c (regarded as existentially quantified variables).

3.3 Recognizing Generalized Locality

Theory extensions $\mathcal{T}_0 \subseteq \mathcal{T}_1$ satisfying $(\mathsf{E(S)Loc}), (\mathsf{E(S)Loc}^{\Psi})$ can be recognized by showing that certain partial models of $\mathcal{T}_1$ can be completed to total models. We consider the following completability conditions:

($\mathsf{Comp_w}$) Every weak partial model A of $\mathcal{T}_1$ with totally defined Σ_0-functions and extension functions with a finite definition domain weakly embeds into a total model B of $\mathcal{T}_1$ s.t. $A_{|\Pi_0}$ and $B_{|\Pi_0}$ are isomorphic.

($\mathsf{Comp}_{\mathsf{w}}^{\Psi}$) Every weak partial model A of $\mathcal{T}_1$ with totally defined Σ_0-functions and such that $\{f(a_1,\ldots,a_n) \mid a_i \in A, f \in \Sigma_1, f_A(a_1,\ldots,a_n) \text{ defined}\}$ is finite and closed under $\Psi_{\mathcal{K}}$ weakly embeds into a total model B of $\mathcal{T}_1$ s.t. $A_{|\Pi_0}$ and $B_{|\Pi_0}$ are elementarily equivalent.

Conditions $(\mathsf{Comp}), (\mathsf{Comp}^{\Psi})$ can be defined by replacing "weak partial model" with "Evans partial model". Assume Ψ satisfies conditions (i)–(iv) in Sect.3:

Theorem 3. *(1) If all terms of $\mathcal{K}$ starting with a Σ_1-function are flat and linear and the extension $\mathcal{T}_0 \subseteq \mathcal{T}_1$ satisfies* $(\mathsf{Comp_w})$ *(resp.* $(\mathsf{Comp}_{\mathsf{w}}^{\Psi})$*) then it satisfies* (ELoc) *[13] (resp.* (ELoc^{Ψ})*).*
(2) If $\mathcal{T}_0$ is a universal theory and the extension $\mathcal{T}_0 \subseteq \mathcal{T}_1$ satisfies (Comp) *(resp.* (Comp^{Ψ})*) then it satisfies* (ESLoc) *[13] (resp.* (ESLoc^{Ψ})*).*

Theorem 3 allows us to identify many examples of local extensions (see Sect. 4). A combination of extensions of a theory $\mathcal{T}_0$ which satisfy condition Comp ($\mathsf{Comp_w}$) also satisfies condition Comp ($\mathsf{Comp_w}$) and hence also condition ESLoc (ELoc).

Theorem 4 ([15]). *Let $\mathcal{T}_0$ be a first order theory with signature $\Pi_0 = (\Sigma_0, \mathsf{Pred})$ and (for $i \in \{1,2\}$) $\mathcal{T}_i = \mathcal{T}_0 \cup \mathcal{K}_i$ be an extension of $\mathcal{T}_0$ with signature $\Pi_i = (\Sigma_0 \cup \Sigma_i, \mathsf{Pred})$. Assume that both extensions $\mathcal{T}_0 \subseteq \mathcal{T}_1$ and $\mathcal{T}_0 \subseteq \mathcal{T}_2$ satisfy condition* $(\mathsf{Comp_w})$*, and that $\Sigma_1 \cap \Sigma_2 = \emptyset$. Then the extension $\mathcal{T}_0 \subseteq \mathcal{T} = \mathcal{T}_0 \cup \mathcal{K}_1 \cup \mathcal{K}_2$ satisfies condition* $(\mathsf{Comp_w})$*. If, additionally, in $\mathcal{K}_i$ all terms starting with a function symbol in Σ_i are flat and linear, for $i = 1, 2$, then the extension is local.*

4 Examples

4.1 Extensions with Free, (Strictly) Monotone, Injective Functions

Any extension $\mathcal{T}_0 \cup \mathsf{Free}(\Sigma)$ of a theory $\mathcal{T}_0$ with a set Σ of *free function symbols* satisfies condition $(\mathsf{Comp_w})$. We also consider monotonicity/antitonicity conditions[3] for an n-ary function f w.r.t. a subset I of its arguments:

$$\mathsf{Mon}^{\sigma}(f) \qquad \bigwedge_{i \in I} x_i \leq_i^{\sigma_i} y_i \wedge \bigwedge_{i \notin I} x_i = y_i \rightarrow f(x_1,..,x_n) \leq f(y_1,..,y_n),$$

where for $i \in I$, $\sigma_i \in \{-,+\}$, and for $i \notin I$, $\sigma_i = 0$, and $\leq^+ = \leq$ and $\leq^- = \geq$.

[3] If $I = \{1,\ldots,n\}$ we speak of monotonicity in all arguments; we denote $\mathsf{Mon}^I(f)$ by $\mathsf{Mon}(f)$. If $I = \emptyset$, $\mathsf{Mon}^{\emptyset}(f)$ is equivalent to the congruence axiom for f.

We showed [13,16] that the extensions of any (possibly many-sorted) theory whose models are posets with functions satisfying the axioms $\mathsf{Mon}^{\sigma}(f)$ satisfy condition ($\mathsf{Comp_w}$) if the codomains of the functions have a bounded semilattice reduct or are totally ordered. In particular, any extension of the theory of reals, rationals or integers with functions satisfying $\mathsf{Mon}^{\sigma}(f)$ into an numeric domain (reals, rationals, integers or a subset thereof) is local, since ($\mathsf{Comp_w}$) holds.

Example 2. The sortedness property $\mathsf{Sorted}(a)$ of the array a can be expressed as a monotonicity axiom: $\forall i, j(1 \leq i \leq j \leq m \rightarrow a(i) \leq a(j))$. An extension of the theory of integers with a function a of arity $\mathsf{i} \rightarrow \mathsf{e}$ satisfying $\mathsf{Sorted}(a)$ (where e is a new or old sort and the theory of sort e is totally ordered) is local.

Consider now the following conditions:

$\mathsf{SMon}(f)$ $\forall i, j(i < j \rightarrow f(i) < f(j))$ and $\mathsf{Inj}(f)$ $\forall i, j(i \neq j \rightarrow f(i) \neq f(j))$

Theorem 5. *Assume that in all models of $\mathcal{T}_0$ the support of sort i has an underlying strict total order relation $<$. Let $\mathcal{T}_1 = \mathcal{T}_0 \cup \mathsf{SMon}(f)$, where f is a new function of arity $\mathsf{i} \rightarrow \mathsf{e}$ (e may be a new or an old sort), in all models of $\mathcal{T}_1$ the support of sort e has an underlying strict total order $<$, and there exist injective order-preserving maps from any interval of the support of sort i to any interval of the support e. Then the extension $\mathcal{T}_0 \subseteq \mathcal{T}_1$ satisfies ($\mathsf{Comp_w}$), hence it is local.*

Example 3. Let $\mathcal{T}_0$ be the (many-sorted) combination of $\mathcal{T}_0^{\mathsf{i}}$ (the theory of linear integer arithmetic, sort i) and $\mathcal{T}_0^{\mathsf{num}}$ (the theory of real numbers, sort num). The extension $\mathcal{T}_1$ of $\mathcal{T}_0$ with a function f of arity $\mathsf{i} \rightarrow \mathsf{num}$ satisfying $\mathsf{SMon}(f)$ is local.

Theorem 6. *A theory extension $\mathcal{T}_0 \subseteq \mathcal{T}_1 = \mathcal{T}_0 \cup \mathsf{Inj}(f)$ with a function f of arity $\mathsf{i} \rightarrow \mathsf{e}$ satisfying $\mathsf{Inj}(f)$ is local provided that in all models of $\mathcal{T}_1$ the cardinality of the support of sort i is lower or equal to the cardinality of the support of sort e.*

4.2 Extensions with Definitions and Boundedness Conditions

Let $\mathcal{T}_0$ be a theory containing a binary predicate $\leq$ which is reflexive, and $f \notin \Sigma_0$.

Guarded boundedness. Let $m \in \mathbb{N}$. For $1 \leq i \leq m$ let $t_i(x_1, \ldots, x_n)$ and $s_i(x_1, \ldots, x_n)$ be terms in the signature Π_0 with variables among $x_1, \ldots, x_n$, and let $\phi_i(x_1, \ldots, x_n)$, $i \in \{1, \ldots, m\}$ be Π_0-formulae with free variables among $x_1, \ldots, x_n$, such that (i) for every $i \neq j$, $\phi_i \wedge \phi_j \models_{\mathcal{T}_0} \bot$, and (ii) for every i, $\mathcal{T}_0 \models \forall \overline{x}(\phi_i(\overline{x}) \rightarrow s_i(\overline{x}) \leq t_i(\overline{x}))$. Let $\mathsf{GBound}(f) = \bigwedge_{i=1}^{m} \mathsf{GBound}^{\phi_i}(f)$, where:

$\mathsf{GBound}^{\phi_i}(f)$ $\forall \overline{x}(\phi_i(\overline{x}) \rightarrow s_i(\overline{x}) \leq f(\overline{x}) \leq t_i(\overline{x}))$.

The extension $\mathcal{T}_0 \subseteq \mathcal{T}_0 \cup \mathsf{GBound}(f)$ is local.

Boundedness for (strictly) monotone and injective functions. Any extension of a theory for which $\leq$ is a partial order (or at least reflexive) with functions satisfying $\mathsf{Mon}^{\sigma}(f)$ and boundedness $\mathsf{Bound}^{t}(f)$ conditions is local [14,16].

$\mathsf{Bound}^{t}(f)$ $\forall x_1, \ldots, x_n(f(x_1, \ldots, x_n) \leq t(x_1, \ldots, x_n))$

where $t(x_1, \ldots, x_n)$ is a Π_0-term with variables among $x_1, \ldots, x_n$ whose associated function has the same monotonicity as f in any model. Similar results hold for strictly monotone/injective functions (under the conditions in Thm. 5, 6).

4.3 Pointer Data Structures à la McPeak and Necula

In [12], McPeak and Necula investigate reasoning in pointer data structures. The language used has sorts p (pointer) and s (scalar). Sets Σ_p and Σ_s of pointer resp. scalar fields are modeled by functions of sort $\mathsf{p} \to \mathsf{p}$ and $\mathsf{p} \to \mathsf{s}$, respectively. A constant null of sort p exists. The only predicate of sort p is equality; predicates of sort s can have any arity. The axioms considered in [12] are of the form

$$\forall p \;\; \mathcal{E} \vee \mathcal{C} \tag{1}$$

where $\mathcal{E}$ contains disjunctions of pointer equalities and $\mathcal{C}$ contains scalar constraints (sets of both positive and negative literals). It is assumed that for all terms $f_1(f_2(\dots f_n(p)))$ occurring in the body of an axiom, the axiom also contains the disjunction $p = \mathsf{null} \vee f_n(p) = \mathsf{null} \vee \dots \vee f_2(\dots f_n(p)) = \mathsf{null}$.[4] Examples of axioms (for doubly linked data structures with priorities) considered there are:

$$\forall p \;\; p \neq \mathsf{null} \wedge \mathsf{next}(p) \neq \mathsf{null} \to \mathsf{prev}(\mathsf{next}(p)) = p \tag{2}$$
$$\forall p \;\; p \neq \mathsf{null} \wedge \mathsf{prev}(p) \neq \mathsf{null} \to \mathsf{next}(\mathsf{prev}(p)) = p \tag{3}$$
$$\forall p \;\; p \neq \mathsf{null} \wedge \mathsf{next}(p) \neq \mathsf{null} \to \mathsf{priority}(p) \geq \mathsf{priority}(\mathsf{next}(p)) \tag{4}$$

(the first two axioms state that prev is a left inverse for next, the third axiom is a monotonicity condition on the function $\mathsf{priority}$). Let $\Psi_{\mathcal{K}}(T) = \mathsf{st}(\mathcal{K}) \cup T \cup \{f(t) \mid t \in \mathsf{st}(\mathcal{K}) \cup T, f \in \Sigma_s\}$ for any set of ground terms T.

Theorem 7. *Let $\mathcal{T}_0$ be a Π_0-theory, where $S_0 = \{\mathsf{s}\}$, and $\mathcal{T}_1 = \mathcal{T}_0 \cup \mathcal{K}$ be the extension of $\mathcal{T}_0$ with signature $\Pi = (\{\mathsf{p}, \mathsf{s}\}, \Sigma, \mathsf{Pred})$ – where $\Sigma = \Sigma_p \cup \Sigma_s \cup \Sigma_0$, and $\mathcal{K}$ is a set of axioms $\forall p(\mathcal{E} \vee \mathcal{C})$ of type (1). Then every partial model A of $\mathcal{K}$ with total Σ_0 functions such that the definition domain of A is closed under $\Psi_{\mathcal{K}}$ (i.e. if $f \in \Sigma_s$ and the p-term t is defined in A then $f(t)$ is defined in A) weakly embeds into a total model of $\mathcal{K}$. Hence $\mathcal{T}_0 \subseteq \mathcal{T}_1$ is a Ψ-stably local extension.*

Ψ-stable locality is not harmful in this case, since all universally quantified variables in the axioms in $\mathcal{K}$ are of sort p, and the number of instances of these variables with subterms in $\Psi_{\mathcal{K}}(G)$ which need to be considered is polynomial in the size of $\mathsf{st}(\mathcal{K}, G)$ (no operations with output sort s generate such terms).

4.4 The Theory of Arrays à la Bradley, Manna and Sipma

In [1] the *array property fragment* is studied, a fragment of the theory of arrays with Presburger arithmetic as index theory and parametric element theories. Consider the extension of the combination $\mathcal{T}_0$ of the index and element theories with functions read, write and axioms:

$$\mathrm{read}(\mathrm{write}(a, i, e), i) = e \qquad j \neq i \to \mathrm{read}(\mathrm{write}(a, i, e), j) = \mathrm{read}(a, j).$$

The array property fragment is defined as follows[5]:

[4] This has the rôle of excluding null pointer errors.

[5] The considerations below are for arrays of dimension 1, the general case is similar.

An *index guard* is a positive Boolean combination of atoms of the form $t \leq u$ or $t = u$ where t and u are either a variable of index sort or a ground term (of index sort) constructed from (Skolem) constants and integer numbers using addition and multiplication with integers. A formula of the form $(\forall i)(\varphi_I(i) \rightarrow \varphi_V(i))$ is an *array property* if φ_I is an index guard and if any universally quantified variable of index sort i only occurs in a direct array read $\mathsf{read}(a, x)$ in φ_V. Array reads may not be nested. The *array property fragment* consists of all existentially-closed Boolean combinations of array property formulae and quantifier-free formulae.

The decision procedure proposed in [1] decides satisfiability of formulae in negation normal form in the array property fragment in the following steps.

1. Replace all existentially quantified array variables with Skolem constants; replace all terms of the form $\mathsf{read}(a, i)$ with $a(i)$; eliminate all terms of the form $\mathsf{write}(a, i, e)$ by replacing the formula $\phi(\mathsf{write}(a, i, e))$ with the conjunction of the formula $\phi(b)$ (obtained by introducing a fresh array name b for $\mathsf{write}(a, i, e)$) with $(b(i) = e) \wedge \forall j(j \leq i - 1 \vee i + 1 \leq j \rightarrow b(j) = a(j))$.[6]
2. Existentially quantified index variables are replaced with Skolem constants.
3. Universal quantification over index variables is replaced by conjunction of suitably chosen instances of the variables.

For determining the set of ground instances to be used in Step 3, the authors prove that certain partial "minimal" models can be completed to total ones.

Theorem 8 (cf. also [1]). *Let $\mathcal{K}$ be the clause part and G the ground part (after the transformation steps (1)–(3)), and $\mathcal{I}$ be the set of index terms defined in [1]. Let $\Psi_{\mathcal{K}}(G) = \{f(i_1, \ldots, i_n) \mid f$ array name $, i_1, \ldots, i_n \in \mathcal{I}\}$. Every partial model of $\mathcal{T}_0 \cup \mathcal{K}[\Psi_{\mathcal{K}}(G)] \cup G$ in which all terms in $\Psi_{\mathcal{K}}(G)$ are defined can be transformed into a (total) model of $\mathcal{T}_0 \cup \mathcal{K} \cup G$. This criterion entails* (ELoc^{Ψ}).

5 A General Framework for Obtaining Locality Results

In Section 4 we identified a large number of theory extensions which can be proved to be local and arise in a natural way in invariant checking and bounded model checking. We distinguish several aspects:

- Programs usually handle complex data structures; it may be necessary to reason about various data types such as lists, arrays, records, etc. We presented classes of such theories for which locality properties hold. Theorem 4 identifies cases in which locality is preserved when combining theories.
- The transition constraint systems we consider define updates of the values of variables and functions which are guarded by formulae which describe a partition of the state space, and therefore define local theory extensions.
- In invariant checking and bounded model checking, the paths to be verified (consisting of successive updates) can be used to identify *chains of extensions* to be considered in the deduction process. These extensions are often (combinations) of various extensions with guarded boundedness conditions.

[6] Note that, by the definition of array property formulae, if a term $\mathsf{write}(a, i, e)$ occurs in the array property fragment then i is an existentially quantified index variable.

Thus, results in Sect. 4.2 and 3.3 allow us to extend the classes of theories from verification for which instantiation-based complete decision procedures exist.

Extensions of the fragment of Necula and McPeak. We are interested in pointer structures which can be changed during execution of a program (a cell of a list can be removed, or a new subtree added into a tree structure). The general remarks above also apply for such situations.

Theorem 9. *Assume that the update axioms* $\mathsf{Update}(\Sigma, \Sigma')$ *describe how the values of the* Σ*-functions change, depending on a finite set* $\{\phi_i \mid i \in I\}$ *of mutually exclusive conditions, expressed as formulae over the base signature and the* Σ*-functions (axioms of type (5) below represent precise ways of defining the updated functions, whereas axioms of type (6) represent boundedness properties on the updated scalar fields, assuming the scalar domains are partially ordered):*

$$\forall \overline{x}(\phi_i(\overline{x}) \rightarrow f_i'(\overline{x}) = s_i(\overline{x})) \;\; i \in I, \text{ where } \phi_i(\overline{x}) \wedge \phi_j(\overline{x}) \models_{\mathcal{T}_0} \bot \text{ for } i \neq j \quad (5)$$

$$\forall \overline{x}(\phi_i(\overline{x}) \rightarrow t_i(\overline{x}) \leq f_i'(\overline{x}) \leq s_i(\overline{x})) \;\; i \in I, \text{ where } \phi_i(\overline{x}) \wedge \phi_j(\overline{x}) \models_{\mathcal{T}_0} \bot \text{ for } i \neq j \quad (6)$$

where s_i, t_i *are terms over the signature* Σ *such that* $\mathcal{T}_0 \models \forall \overline{x}(\phi_i(\overline{x}) \rightarrow t_i(\overline{x}) \leq s_i(\overline{x}))$ *for all* $i \in I$*. They define local theory extensions. This holds for any extensions of disjoint combinations of various pointer structures with such update axioms.*

Example 4. Consider the following algorithm for inserting an element c with priority field $c.\mathsf{prio} = x$ into a doubly-linked list sorted w.r.t. the priority fields.

$c.\mathsf{prio} = x, c.\mathsf{next} = \mathsf{null}$
for all $p \neq c$ **do**
if $p.\mathsf{prio} \leq x$ **then if** $p.\mathsf{prev} = \mathsf{null}$ **then** $c.\mathsf{next}' = p$, **endif**; $p.\mathsf{next}' = p.\mathsf{next}$
$p.\mathsf{prio} > x$ **then** **case** $p.\mathsf{next} = \mathsf{null}$ **then** $p.\mathsf{next}' := c, c.\mathsf{next}' = \mathsf{null}$
$p.\mathsf{next} \neq \mathsf{null} \wedge p.\mathsf{next} > x$ **then** $p.\mathsf{next}' = p.\mathsf{next}$
$p.\mathsf{next} \neq \mathsf{null} \wedge p.\mathsf{next} \leq x$ **then** $p.\mathsf{next}' = c$, $c.\mathsf{next}' = p.\mathsf{next}$

The update rules $\mathsf{Update}(\mathsf{next}, \mathsf{next}')$ can be read from the program above:

$\forall p(p \neq \mathsf{null} \wedge p \neq c \wedge \mathsf{prio}(p) \leq x \wedge (\mathsf{prev}(p) = \mathsf{null}) \rightarrow \mathsf{next}'(c) = p \wedge \mathsf{next}'(p) = \mathsf{next}(p))$
$\forall p(p \neq \mathsf{null} \wedge p \neq c \wedge \mathsf{prio}(p) \leq x \wedge (\mathsf{prev}(p) \neq \mathsf{null}) \rightarrow \mathsf{next}'(p) = \mathsf{next}(p))$
$\forall p(p \neq \mathsf{null} \wedge p \neq c \wedge \mathsf{prio}(p) > x \wedge \mathsf{next}(p) = \mathsf{null} \rightarrow \mathsf{next}'(p) = c \wedge \mathsf{next}'(c) = \mathsf{null})$
$\forall p(p \neq \mathsf{null} \wedge p \neq c \wedge \mathsf{prio}(p) > x \wedge \mathsf{next}(p) \neq \mathsf{null} \wedge \mathsf{prio}(\mathsf{next}(p)) > x \rightarrow \mathsf{next}'(p) = \mathsf{next}(p))$
$\forall p(p \neq \mathsf{null} \wedge p \neq c \wedge \mathsf{prio}(p) > x \wedge \mathsf{next}(p) \neq \mathsf{null} \wedge \mathsf{prio}(\mathsf{next}(p)) \leq x \rightarrow \mathsf{next}'(p) = c \wedge \mathsf{next}'(c) = \mathsf{next}(p))$

We prove that if the list is sorted, it remains so after insertion, i.e. the formula:

$$d \neq \mathsf{null} \wedge \mathsf{next}'(d) \neq \mathsf{null} \wedge \neg \mathsf{prio}(d) \geq \mathsf{prio}(\mathsf{next}'(d))$$

is unsatisfiable in the extension $\mathcal{T}_1 = \mathcal{T}_0 \cup \mathsf{Update}(\mathsf{next}, \mathsf{next}')$ of the theory $\mathcal{T}_0$ of doubly linked lists with a monotone field prio. $\mathcal{T}_0$ is axiomatized by the axioms $\mathcal{K} = \{(2), (3), (4)\}$ in Sect. 4. The update rules are guarded boundedness axioms, so the extension $\mathcal{T}_0 \subseteq \mathcal{T}_1$ is local. Hence, the satisfiability task above w.r.t. $\mathcal{T}_1$ can be reduced to a satisfiability task w.r.t. $\mathcal{T}_0$ as follows:

Update_0	$d{\neq}\mathsf{null} \wedge d{\neq}c \wedge \mathsf{prio}(d){\leq}x \wedge \mathsf{prev}(d){=}\mathsf{null} \rightarrow c_1{=}d$
	$d{\neq}\mathsf{null} \wedge d{\neq}c \wedge \mathsf{prio}(d){\leq}x \wedge \mathsf{prev}(d){=}\mathsf{null} \rightarrow d_1{=}\mathsf{next}(d)$
	$d{\neq}\mathsf{null} \wedge d{\neq}c \wedge \mathsf{prio}(d){\leq}x \wedge \mathsf{prev}(d){\neq}\mathsf{null} \rightarrow d_1{=}\mathsf{next}(d)$
	$d{\neq}\mathsf{null} \wedge d{\neq}c \wedge \mathsf{prio}(d){>}x \wedge \mathsf{next}(d){=}\mathsf{null} \rightarrow d_1{=}c \wedge c_1{=}\mathsf{null}$
	$d{\neq}\mathsf{null} \wedge d{\neq}c \wedge \mathsf{prio}(d){>}x \wedge \mathsf{next}(d){\neq}\mathsf{null} \wedge \mathsf{prio}(\mathsf{next}(d)){<}x \rightarrow d_1{=}\mathsf{next}(d)$
	$d{\neq}\mathsf{null} \wedge d{\neq}c \wedge \mathsf{prio}(d){>}x \wedge \mathsf{next}(d){\neq}\mathsf{null} \wedge \mathsf{prio}(\mathsf{next}(d)){\leq}x \rightarrow d_1{=}c$
	$d{\neq}\mathsf{null} \wedge d{\neq}c \wedge \mathsf{prio}(d){>}x \wedge \mathsf{next}(d){\neq}\mathsf{null} \wedge \mathsf{prio}(\mathsf{next}(d)){\leq}x \rightarrow c_1{=}\mathsf{next}(d)$
G_0	$d \neq \mathsf{null} \wedge \mathsf{next}'(d) \neq \mathsf{null} \wedge \neg\mathsf{prio}(d) \geq \mathsf{priority}(\mathsf{next}'(d))$
N_0	$d{=}c \rightarrow d_1{=}c_1$ (corresponds to $\mathsf{Def}: \ \mathsf{next}'(d){=}d_1 \wedge \mathsf{next}'(c){=}c_1$)

To check the satisfiability of $G' = \mathsf{Update}_0 \wedge G_0 \wedge N_0$ w.r.t. $\mathcal{T}_0$ we use the Ψ-stable locality of the theory defined by the axioms $\mathcal{K} = \{(2), (3), (4)\}$ of doubly linked lists with decreasing priorities in Sect. 4 or the instantiation method in [12].

Extending the array property fragment. Let $\mathcal{T}_0$ be the array property fragment in [1] (set of arrays Σ_0). There are several ways of extending $\mathcal{T}_0$:

Theorem 10. *Let $\mathcal{T}_1{=}\mathcal{T}_0 \cup \mathcal{K}$ be an extension of $\mathcal{T}_0$ with new arrays in a set Σ_1.*

(1) If $\mathcal{K}$ consists of guarded boundedness axioms, or guarded definitions (cf. Sect.4.2) for the Σ_1-function symbols, then the extension $\mathcal{T}_0 \subseteq \mathcal{T}_1$ is local. [7]
(2) If $\mathcal{K}$ consists of injectivity or (strict) monotonicity (and possibly boundedness axioms) for the function symbols in Σ_1 then the extension $\mathcal{T}_0 \subseteq \mathcal{T}_1$ is local if the assumptions about the element theory specified in Sect. 4.1 hold.
(3) Any combination of extensions of $\mathcal{T}_0$ as those mentioned in (1),(2) with disjoint sets of new array constants leads to a local extension of $\mathcal{T}_0$.

If the guards ϕ_i of the axioms in $\mathcal{K}$ are clauses then the result of the hierarchical reasoning method in Thm. 1 is a formula in $\mathcal{T}_0$, hence satisfiability of ground clauses w.r.t. $\mathcal{T}_0 \cup \mathcal{K}$ is decidable. Similarly for chains of extensions. The same holds for testing satisfiability of goals $\Gamma_0 \cup G$ where Γ_0 and $(\mathcal{K}[G])_0$ belong to the array property fragment. For general guards and chains of extensions decidability depends on the form of the formulae obtained by hierarchical reduction(s).

Example 5. The example presented in Section 1.2 illustrates the extension of the fragment in [1] we consider. The task is to check the unsatisfiability of the formula $G = (1 \leq c \leq m \wedge 1 \leq d \leq m \wedge c \neq d \wedge p'(c) = p'(d))$ in the extension of the many sorted combination $\mathcal{T}_0$ of $\mathbb{Z}, \mathbb{R}_+, \{0,1\}$ with the axioms $\forall i,j(1 \leq i \leq m \wedge 1 \leq j \leq m \wedge i \neq j \rightarrow p(i) \neq p(j)) \wedge \mathsf{Update}(a,p,a',p')$.
The extension can be expressed as a chain: $\mathcal{T}_0 \subseteq \mathcal{T}_1 = \mathcal{T}_0 \cup \mathsf{Inj}(p) \subseteq \mathcal{T}_2 = \mathcal{T}_1 \cup \mathsf{Update}(a,p,a',p')$. By the locality of the second extension (with guarded boundedness axioms) we obtain the following reduction of the task of proving $\mathcal{T}_2 \wedge G \models \bot$ to a satisfiability problem w.r.t. $\mathcal{T}_1$. We take into account only the instances of $\mathsf{Update}(a,p,a',p')$ which contain ground terms occurring in G. This means that the axioms containing a' do not need to be considered. After purification and skolemization of the existentially quantified variables we obtain:

[7] An example are definitions of new arrays by writing x at a (constant) index c, axiomatized by $\{\forall i(i \neq c \rightarrow a'(i) = a(i)), \quad \forall i(i = c \rightarrow a'(i) = x)\}$.

Update_0	$1 \leq c \leq m \wedge (1 \leq k_c \leq m \wedge k_c \neq c \rightarrow p(c) > p(k_c)) \rightarrow c_1 = x$ $1 \leq d \leq m \wedge (1 \leq k_d \leq m \wedge k_d \neq d \rightarrow p(d) > p(k_d)) \rightarrow d_1 = x$ $\forall j (1 \leq j \leq m \wedge j \neq c \rightarrow p(c) > p(j)) \vee (1 \leq c \leq m \rightarrow c_1 = p(c) + y)$ $\forall j (1 \leq j \leq m \wedge j \neq d \rightarrow p(d) > p(j)) \vee (1 \leq d \leq m \rightarrow d_1 = p(d) + y)$
G_0	$1 \leq c \leq m \wedge 1 \leq d \leq m \wedge c \neq d \wedge c_1 = d_1$
N_0	$c = d \rightarrow c_1 = d_1$ (corresponds to $\mathsf{Def}: \; p'(c) = c_1 \wedge p'(d) = d_1$)

We reduced the problem to checking satisfiability of $G_1 = \mathsf{Update}_0 \wedge G_0 \wedge N_0$ (which contains universal quantifiers) w.r.t. $\mathcal{T}_1$. Let $G_1 = G_g \wedge G_\forall$, where G_g is the ground part of G and $G_\forall$ the part of G containing universally quantified variables. We now have to check whether $\mathcal{T}_0 \wedge \mathsf{Inj}(p) \wedge G_\forall \wedge G_g \models \bot$. Note that extensions of injectivity axioms and boundedness are local, and thus $\mathcal{T}_0 \subseteq \mathcal{T}_0 \wedge \mathsf{Inj} \wedge G_\forall$ is a local extension. This makes the following reduction possible:

Inj_0	$1 \leq i \neq j \leq m \rightarrow p(i) \neq p(j)$	where i, j are instantiated with c, d, k_c, k_d + purification
$G_{\forall 0}$	$(1 \leq j \leq m \wedge j \neq c \rightarrow c_2 > p(j)) \vee (1 \leq c \leq m \rightarrow c_1 = c_2 + y)$ $(1 \leq j \leq m \wedge j \neq d \rightarrow d_2 > p(j)) \vee (1 \leq d \leq m \rightarrow d_1 = d_2 + y)$	
G_g	$1 \leq c \leq m \wedge (1 \leq k_c \leq m \wedge k_c \neq c \rightarrow c_2 > c_3) \rightarrow c_1 = x$ $1 \leq d \leq m \wedge (1 \leq k_d \leq m \wedge k_d \neq d \rightarrow d_2 > d_3) \rightarrow d_1 = x$ $1 \leq c \leq m \wedge 1 \leq d \leq m \wedge c \neq d \wedge c_1 = d_1$	$c = d \rightarrow c_1 = d_1$
N_0'	$c = d \rightarrow c_2 = d_2, \; c = k_c \rightarrow c_2 = c_3, \; c = k_d \rightarrow c_2 = d_3, \; d = k_c \rightarrow d_2 = c_3, \; d = k_d \rightarrow d_2 = d_3,$ $k_c = k_d \rightarrow c_3 = d_3$ (corr. to $\mathsf{Def}_1 : p(c) = c_2 \wedge p(d) = d_2 \wedge p(k_c) = c_3 \wedge p(k_d) = d_3$)	

We can use a prover for a combination of integers and reals to determine whether the conjunction of formulae above is satisfiable or symbolic computation packages performing quantifier elimination over the combined theory to derive constraints between x and y which guarantee injectivity after update.

6 Experiments

We have implemented the approach for hierarchical reasoning in local theory extensions described in [13], cf. also Sect. 3.1. The tool we devised allows us to reduce satisfiability problems in an extended theory to a base theory for which we can then use existing solvers. It takes as input the axioms of the theory extension, the ground goal and the list of extension function symbols. Chains of extensions are handled by having a list of axiom sets, and correspondingly a list of lists of extension function symbols. We follow the steps in Sect. 3.1: the input is analyzed for ground subterms with extension symbols at the root. After instantiating the axioms w.r.t. these terms, the instances are purified (so the extension symbols are removed). The resulting formula is either given to a prover for a base theory, or taken as goal for another reduction (if we have a chain of extensions). Currently, we can produce base theory output for Yices, Mathsat, CVC and Redlog, but other solvers can be integrated easily. We ran tests on various examples, including different versions of a train controller example [10,4], an array version of the insertion algorithm, and reasoning in theories of lists. Test results and comparisons can be found in [17] (which contains preliminary versions of some of the results in this paper, in an extended form). Runtimes

range from 0.047s to 0.183s for various versions of the train controller example resp. to 0.4s for array examples (including an example from [1]). While Yices can also be used successfully directly for unsatisfiable formulae, this does not hold if we change the input problem to a formula which is satisfiable w.r.t. the extended theory. In this case, Yices returns "unknown" after a 300 second timeout. After the reduction with our tool, Yices (applied to the problem for the base theory) returns "satisfiable" in fractions of a second, and even a model for this problem that can easily be lifted to a model in the extended theory for the initial set of clauses[8]. Even more information can be obtained using the quantifier elimination facilities offered e.g. by Redlog for determining *constraints between the parameters* of the problems which guarantee safety.

We are working towards extending the tool support to stable locality, as well as for extensions with clauses containing proper first-order formulae.

7 Conclusions

We presented a general framework – based on a general notion of locality – which allows to identify complex theories important in verification for which efficient (hierarchical and modular) reasoning methods exist. We showed that locality considerations allow us to obtain parameterized decidability and complexity results for many (combinations of) theories important in verification (of parametric systems). We showed that many theories of data structures studied in the verification literature are local extensions of a base theory. The list of theories we considered is not exhaustive. (Due to space limitations we did not discuss the theory of arrays studied in [7], whose main ingredient is the existence of undefined values in arrays and properties (e.g. injectivity) are guarded by definedness conditions. The main result in [7] can be seen as a locality result as the arguments used are based on the possibility of completing partial to total models.) The general framework we use allows us to identify situations in which some of the syntactical restrictions imposed in previous papers can be relaxed.

The deduction tasks we considered here are typical for invariant checking and bounded model checking. The next step would be to integrate these methods into verification tools based on abstraction/refinement. Our work on hierarchical interpolation in local extensions [14] can be extended to many of the theories of data structures described in this paper. This is the topic of a future paper.

Acknowledgments. We thank Aaron Bradley for helpful comments made on a preliminary version of this paper. 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). See `www.avacs.org` for more information.

[8] The lifting is straightforward, given the output of our tool, but is not automated at the moment.

References

1. Bradley, A.R., Manna, Z., Sipma, H.B.: What's decidable about arrays? In: Emerson, E.A., Namjoshi, K.S. (eds.) VMCAI 2006. LNCS, vol. 3855, pp. 427–442. Springer, Heidelberg (2006)
2. Burmeister, P.: A Model Theoretic Oriented Approach to Partial Algebras: Introduction to Theory and Application of Partial Algebras, Part I. In: Mathematical Research, vol. 31, Akademie-Verlag, Berlin (1986)
3. Burris, S.: Polynomial time uniform word problems. Mathematical Logic Quarterly 41, 173–182 (1995)
4. Faber, J., Jacobs, S., Sofronie-Stokkermans, V.: Verifying CSP-OZ-DC specifications with complex data types and timing parameters. In: Davies, J., Gibbons, J. (eds.) IFM 2007. LNCS, vol. 4591, pp. 233–252. Springer, Heidelberg (2007)
5. Ganzinger, H.: Relating semantic and proof-theoretic concepts for polynomial time decidability of uniform word problems. In: Proc. 16th IEEE Symposium on Logic in Computer Science (LICS 2001), pp. 81–92. IEEE Computer Society Press, Los Alamitos (2001)
6. Ganzinger, H., Sofronie-Stokkermans, V., Waldmann, U.: Modular proof systems for partial functions with Evans equality. Information and Computation 204(10), 1453–1492 (2006)
7. Ghilardi, S., Nicolini, E., Ranise, S., Zucchelli, D.: Deciding extensions of the theory of arrays by integrating decision procedures and instantiation strategies. In: Fisher, M., van der Hoek, W., Konev, B., Lisitsa, A. (eds.) JELIA 2006. LNCS (LNAI), vol. 4160, pp. 177–189. Springer, Heidelberg (2006)
8. 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, San Francisco (1992)
9. Givan, R., McAllester, D.A.: Polynomial-time computation via local inference relations. ACM Transactions on Computational Logic 3(4), 521–541 (2002)
10. Jacobs, S., Sofronie-Stokkermans, V.: Applications of hierarchical reasoning in the verification of complex systems. Electronic Notes in Theoretical Computer Science 174(8), 39–54 (2007)
11. McAllester, D.: Automatic recognition of tractability in inference relations. Journal of the Association for Computing Machinery 40(2), 284–303 (1993)
12. McPeak, S., Necula, G.C.: Data structure specifications via local equality axioms. In: Etessami, K., Rajamani, S.K. (eds.) CAV 2005. LNCS, vol. 3576, pp. 476–490. Springer, Heidelberg (2005)
13. 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)
14. Sofronie-Stokkermans, V.: Interpolation in local theory extensions. In: Furbach, U., Shankar, N. (eds.) IJCAR 2006. LNCS (LNAI), vol. 4130, pp. 235–250. Springer, Heidelberg (2006)
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 (LNAI), vol. 4720, pp. 47–71. Springer, Heidelberg (2007)

16. Sofronie-Stokkermans, V., Ihlemann, C.: Automated reasoning in some local extensions of ordered structures. In: Proc. of ISMVL-2007, IEEE Computer Society Press, Los Alamitos (2007), http://dx.doi.org/10.1109/ISMVL.2007.10
17. Sofronie-Stokkermans, V., Ihlemann, C., Jacobs, S.: Local theory extensions, hierarchical reasoning and applications to verification. In: Dagstuhl Seminar Proceedings 07401,, http://drops.dagstuhl.de/opus/volltexte/2007/1250