Interpolation in local theory extensions

Viorica Sofronie-Stokkermans

Max-Planck-Institut für Informatik, Stuhlsatzenhausweg 85, Saarbrücken, Germany
e-mail: sofronie@mpi-inf.mpg.de

Abstract. In this paper we study interpolation in local extensions of a base theory. We identify situations in which it is possible to obtain interpolants in a hierarchical manner, by using a prover and a procedure for generating interpolants in the base theory as black-boxes. We present several examples of theory extensions in which interpolants can be computed this way, and discuss applications in verification, knowledge representation, and modular reasoning in combinations of local theories.

1 Introduction

Many problems in mathematics and computer science can be reduced to proving satisfiability of conjunctions of (ground) literals modulo a background theory. This theory can be a standard theory, the extension of a base theory with additional functions, or a combination of theories. It is therefore very important to find efficient methods for reasoning in standard as well as complex theories. However, it is often equally important to find local causes for inconsistency. In distributed databases, for instance, finding local causes of inconsistency can help in locating errors. Similarly, in abstraction-based verification, finding the cause of inconsistency in a counterexample helps to rule out spurious counterexamples.

The problem can be formally described as follows: Let A and B be sets of ground clauses in a theory $\mathcal{T}$. Assume that $A \wedge B$ is inconsistent with respect to $\mathcal{T}$. Can we find a ground formula I, containing only constants and function symbols common to A and B, such that I is a consequence of A w.r.t. $\mathcal{T}$, and $B \wedge I$ is inconsistent modulo $\mathcal{T}$? If so, I is an *interpolant* of A and B, and can be regarded as a "local" explanation for the inconsistency of $A \wedge B$.

In this paper we study possibilities of obtaining ground interpolants in theory extensions. We identify situations in which it is possible to do this in a hierarchical manner, by using a prover and a procedure for generating interpolants in the base theory as "black-boxes".

The main contributions of the paper are summarized below:

- First, we identify new examples of local theory extensions. In these, hierarchical reasoning is possible.
- Second, we present a method for generating interpolants in extensions of a base theory by means of sets of clauses. The method is general, in the sense that it can be applied to an extension $\mathcal{T}_1$ of a theory $\mathcal{T}_0$ provided that:

(a) (i) $\mathcal{T}_0$ is convex; (ii) $\mathcal{T}_0$ is P-interpolating for a specified set P of predicates (cf. the definition in Section 4.2); (iii) in $\mathcal{T}_0$ every inconsistent conjunction of ground clauses $A \wedge B$ allows a ground interpolant.
(b) the extension clauses have a special form (i.e. type (3) in Section 4.2).

The method is *hierarchical*: the problem of finding interpolants in $\mathcal{T}_1$ is reduced to that of finding interpolants in the base theory $\mathcal{T}_0$. We can use the properties of $\mathcal{T}_0$ to control the form of interpolants in the extension $\mathcal{T}_1$.

- Third, we identify examples of theory extensions with properties (a) and (b).
- Fourth, we discuss application domains such as: modular reasoning in combinations of local theories (characterization of the type of information which needs to be exchanged), reasoning in distributed databases, and verification.

The existence of ground interpolants has been studied in several recent papers, mainly motivated by abstraction-refinement based verification [3,4,5,10]. In [4] McMillan presents a method for generating ground interpolants from proofs in an extension of linear rational arithmetic with uninterpreted function symbols. The use of free function symbols is sometimes too coarse (cf. the example in Section 1.1). Here, we show that similar results also hold for other types of extensions of a base theory, provided that the base theory has some of the properties of linear rational arithmetic. Another method for generating interpolants for combinations of theories over disjoint signatures from Nelson-Oppen-style unsatisfiability proofs was proposed by Yorsh and Musuvathi in [10]. Although we impose similar conditions on $\mathcal{T}_0$, our method is orthogonal to theirs, as it also allows to consider combinations of theories over non-disjoint signatures.

Structure of the paper: In Section 1.1 we provide motivation for the study. In Section 2 the basic notions needed in the paper are introduced. Section 3 contains results on local theory extensions. In Section 4 local extensions allowing hierarchical interpolation are identified and some applications (modular reasoning in combinations of theories, reasoning in complex databases, and verification) are presented. We end with conclusions and plans for future work.

1.1 Motivation

We present two fields of applications: knowledge representation and verification.

Knowledge representation. Consider a simple (and faulty) terminological database for chemistry, consisting of two extensions of a common kernel Chem (basic chemistry): AChem (anorganic chemistry) and $\mathsf{BioChem}$ (biochemistry). Assume that Chem contains a set $\mathsf{C}_0 = \{\mathsf{process}, \mathsf{reaction}, \mathsf{substance}, \mathsf{organic}, \mathsf{anorganic}\}$ of concepts and a set Γ_0 of constraints:

$$\Gamma_0 = \{\mathsf{organic} \wedge \mathsf{anorganic} = \emptyset, \quad \mathsf{organic} \subseteq \mathsf{substance}, \quad \mathsf{anorganic} \subseteq \mathsf{substance}\}$$

Let AChem be an extension of Chem with concepts $\mathsf{C}_1 = \{\mathsf{cat\text{-}oxydation}, \mathsf{oxydation}\}$, a rôle $\mathsf{R}_1 = \{\mathsf{catalyzes}\}$, terminology T_1 and constraints Γ_1:

$$\mathsf{T}_1 = \{\mathsf{cat\text{-}oxydation} = \mathsf{substance} \wedge \exists \mathsf{catalyzes}(\mathsf{oxydation})\}$$
$$\Gamma_1 = \{\mathsf{reaction} \subseteq \mathsf{oxydation}, \quad \mathsf{cat\text{-}oxydation} \subseteq \mathsf{anorganic}, \quad \mathsf{cat\text{-}oxydation} \neq \emptyset\}.$$

Let $\mathsf{BioChem}$ be an extension of Chem with a concept $\mathsf{C}_2 = \{\mathsf{enzyme}\}$, rôles $\mathsf{R}_2 = \{\mathsf{produces}, \mathsf{catalyzes}\}$, terminology T_2 and constraints Γ_2:

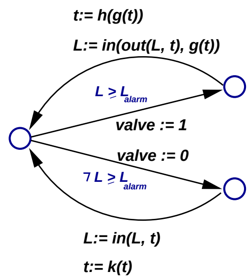
$\mathsf{T}_2 = \{\mathsf{reaction}{=}\mathsf{process} \wedge \exists\mathsf{produces}(\mathsf{substance}), \mathsf{enzyme} = \mathsf{organic} \wedge \exists\mathsf{catalyzes}(\mathsf{reaction})\}$
$\Gamma_2 = \{\mathsf{enzyme} \neq \emptyset\}$

The combination of Chem, AChem and $\mathsf{BioChem}$ is inconsistent (we wrongly added to Γ_1 the constraint $\mathsf{reaction} \subseteq \mathsf{oxydation}$ instead of $\mathsf{oxydation} \subseteq \mathsf{reaction}$). This can be proved as follows: By results in ([7], p.156 and p.166) the combination of Chem, AChem and $\mathsf{BioChem}$ is inconsistent iff

$$\Gamma_0 \wedge (\mathsf{T}_1 \wedge \Gamma_1) \wedge (\mathsf{T}_2 \wedge \Gamma_2) \models_{\mathcal{T}} \perp \tag{1}$$

where $\mathcal{T}$ is the extension $\mathsf{SLat} \wedge \bigcup_{f \in \mathsf{R}_1 \cup \mathsf{R}_2} \mathsf{Mon}_f$ of the theory of semilattices with first element 0 and monotone function symbols corresponding to $\exists r$ for each rôle $r \in \mathsf{R}_1 \cup \mathsf{R}_2$. Using, for instance, the hierarchical calculus presented in [8] (see also Section 3), the contradiction can be found in polynomial time. In order to find the mistake we look for an explanation for the inconsistency in the common language of AChem and $\mathsf{BioChem}$. (Common to AChem and $\mathsf{BioChem}$ are the concepts $\mathsf{substance}, \mathsf{organic}, \mathsf{anorganic}, \mathsf{reaction}$ and of rôle $\mathsf{catalyzes}$.) This can be found by computing an interpolant for the conjunction in (1) in the theory of semilattices with monotone operators. In this paper we show how such interpolants can be found in an efficient way.

Verification. Consider a water level controller modeled as follows: Changes in the water level by inflow/outflow are represented as functions $\mathsf{in}, \mathsf{out}$, depending on time t and water level L. Alarm and overflow levels $L_{\mathsf{alarm}} < L_{\mathsf{overflow}}$ are known.



- If $L \geq L_{\mathsf{alarm}}$ then a valve is opened until time $g(t)$, the water level changes by $L' := \mathsf{in}(\mathsf{out}(L, t), g(t))$ and time by $t' := h(g(t))$.
- If $L < L_{\mathsf{alarm}}$ then the valve is closed; the water level changes by $L' := \mathsf{in}(L, t)$ and time by $t' := k(t)$.

We want to show that if initially $L < L_{\mathsf{alarm}}$ then the water level always remains below L_{overflow}.

In [4], McMillan proposed a method in which interpolation (e.g. for linear arithmetic + free functions) is used for abstraction refinement. If $\mathsf{in}, \mathsf{out}$ are free functions then $L{<}L_{\mathsf{alarm}} \wedge L'{\approx}\mathsf{in}(L, t) \wedge t'{\approx}k(L) \wedge \neg L'{\leq}L_{\mathsf{overflow}}$ is satisfiable, so there exists a path from an initial state to an error state. To prove safety, we need to impose restrictions on in and out, e.g.:

$$\forall L, t\ (L{<}L_{\mathsf{alarm}} {\rightarrow} \mathsf{in}(L, t){<}L_{\mathsf{overflow}}), \quad \forall L, t\ (L{<}L_{\mathsf{overflow}} {\rightarrow} \mathsf{out}(L, t){<}L_{\mathsf{alarm}}) \tag{2}$$

The method we present here allows us to efficiently generate ground interpolants for extensions with functions satisfying condition (2), and also for a whole class of more general axioms. An immediate application is to verification by abstraction-refinement; there are other potential applications (e.g. goal-directed overapproximation for achieving faster termination, or automatic invariant generation).

2 Preliminaries

Theories and models. Theories can be regarded as sets of formulae or as sets of models. In this paper, whenever we speak about a theory $\mathcal{T}$ – if not otherwise specified – we implicitly refer to the set $\mathsf{Mod}(\mathcal{T})$ of all models of $\mathcal{T}$. Let $\mathcal{T}$ be a theory in a given signature $\Pi = (\Sigma, \mathsf{Pred})$, where Σ is a set of function symbols and Pred a set of predicate symbols. Let ϕ and ψ be formulae over the signature Π with variables in a set X. The notion of truth of formulae and of entailment is the usual one in logic. We say that ϕ is true w.r.t. $\mathcal{T}$ (denoted $\models_{\mathcal{T}} \phi$) if ϕ is true in each model $\mathcal{M}$ of $\mathcal{T}$. ϕ is satisfiable w.r.t. $\mathcal{T}$ if there exists at least one model $\mathcal{M}$ of $\mathcal{T}$ and an assignment $\beta : X \rightarrow \mathcal{M}$ such that $(\mathcal{M}, \beta) \models \phi$. Otherwise we say that ϕ is unsatisfiable. We say that ϕ entails ψ w.r.t. $\mathcal{T}$ (denoted $\phi \models_{\mathcal{T}} \psi$) if for every model $\mathcal{M}$ of $\mathcal{T}$ and every valuation β, if $(\mathcal{M}, \beta) \models \phi$ then $(\mathcal{M}, \beta) \models \psi$. Note that ϕ is unsatisfiable w.r.t. $\mathcal{T}$ iff $\phi \models_{\mathcal{T}} \perp$.

Interpolation. A theory $\mathcal{T}$ has interpolation if, for all formulae ϕ and ψ in the signature of $\mathcal{T}$, if $\phi \models_{\mathcal{T}} \psi$ then there exists a formula I containing only symbols which occur in both ϕ and ψ such that $\phi \models_{\mathcal{T}} I$ and $I \models_{\mathcal{T}} \psi$. First order logic has interpolation but even if ϕ and ψ are e.g. conjunctions of ground literals I may still be an arbitrary formula. It is often important to identify situations in which ground clauses have ground interpolants.

We say that a theory $\mathcal{T}$ has the *ground interpolation property* (or, shorter, that $\mathcal{T}$ has *ground interpolation*) if for all ground clauses $A(\overline{c}, \overline{d})$ and $B(\overline{c}, \overline{e})$, if $A(\overline{c}, \overline{d}) \wedge B(\overline{c}, \overline{e}) \models_{\mathcal{T}} \perp$ then there exists a ground formula $I(\overline{c})$, containing only the constants $\overline{c}$ occurring both in A and B, such that $A(\overline{c}, \overline{d}) \models_{\mathcal{T}} I(\overline{c})$ and $B(\overline{c}, \overline{e}) \wedge I(\overline{c}) \models_{\mathcal{T}} \perp$.

There exist results which relate ground interpolation to amalgamation or the injection transfer property [2,1,9] and thus allow us to recognize many theories with ground interpolation. Thus it can be proved that the following equational classes have ground interpolation: (abelian) groups, partially-ordered sets, lattices, semilattices, distributive lattices and Boolean algebras. However, in many applications one needs to consider extensions or combinations of theories, and proving amalgamation properties can be complicated. On the other hand, just knowing that ground interpolants exist is not sufficient: we would like to construct the interpolants fast, and to use the advantages of modular or hierarchical reasoning for constructing them. This is why in this paper we aim at giving methods for *constructing* interpolants in a hierarchical way.

3 Local theory extensions

Let $\mathcal{T}_0$ be a theory with signature $\Pi_0 = (\Sigma_0, \mathsf{Pred})$. We consider extensions $\mathcal{T}_1$ of $\mathcal{T}_0$ with signature $\Pi = (\Sigma, \mathsf{Pred})$, where $\Sigma = \Sigma_0 \cup \Sigma_1$ (i.e. the signature is extended by new *function symbols*) and $\mathcal{T}_1$ is obtained from $\mathcal{T}_0$ by adding a set $\mathcal{K}$ of (universally quantified) clauses. Thus, $\mathsf{Mod}(\mathcal{T}_1)$ consists of all Π-structures which are models of $\mathcal{K}$ and whose reduct to Π_0 is a model of $\mathcal{T}_0$.

A *partial Π-structure* is a structure $\mathcal{M} = (M, \{f_M\}_{f\in\Sigma}, \{P_M\}_{P\in\mathsf{Pred}})$, where $M \neq \emptyset$ and for every $f \in \Sigma$ with arity n, f_M is a partial function from M^n to M. The notion of evaluating a term t with respect to a variable assignment $\beta : X \to M$ for its variables in a partial structure $\mathcal{M}$ is the same as for total algebras, except that this evaluation is undefined if $t = f(t_1, \dots, t_n)$ and at least one of $\beta(t_i)$ is undefined, or else $(\beta(t_1), \dots, \beta(t_n))$ is not in the domain of f_M. Let $\mathcal{M}$ be a partial Π-structure, C a clause and $\beta : X \to M$. Then $(\mathcal{M}, \beta) \models_w C$ iff either (i) for some term t in C, $\beta(t)$ is undefined, or else (ii) $\beta(t)$ is defined for all terms t of C, and there exists a literal L in C s.t. $\beta(L)$ is true in $\mathcal{M}$. $\mathcal{M}$ *weakly satisfies* C (notation: $\mathcal{M} \models_w C$) if $(\mathcal{M}, \beta) \models_w C$ for all assignments β. $\mathcal{M}$ *weakly satisfies a set of clauses* $\mathcal{K}$ ($\mathcal{M} \models_w \mathcal{K}$) if $\mathcal{M} \models_w C$ for all $C \in \mathcal{K}$.

3.1 Definition and examples

Let $\mathcal{K}$ be a set of (universally quantified) clauses in the signature $\Pi = (\Sigma, \mathsf{Pred})$, where $\Sigma = \Sigma_0 \cup \Sigma_1$. In what follows, when referring to sets G of ground clauses we assume they are in the signature $\Pi^c = (\Sigma \cup \Sigma_c, \mathsf{Pred})$ where Σ_c is a set of new constants. An extension $\mathcal{T}_0 \subseteq \mathcal{T}_0 \cup \mathcal{K}$ is *local* if, in order to prove unsatisfiability of a set G of clauses with respect to $\mathcal{T}_0 \cup \mathcal{K}$, it is sufficient to use only those instances $\mathcal{K}[G]$ of $\mathcal{K}$ in which the terms starting with extension functions are in the set $\mathsf{st}(G, \mathcal{K})$ of ground terms which already occur in G or $\mathcal{K}$. Formally, $\mathcal{T}_0 \subseteq \mathcal{T}_1 {=} \mathcal{T}_0 \cup \mathcal{K}$ is a local extension if it satisfies condition (Loc):

(Loc) For every set G of ground clauses, $G \models_{\mathcal{T}_1} \perp$ iff there is no partial Π^c-structure P such that $P_{|\Pi_0}$ is a total model of $\mathcal{T}_0$, all terms in $\mathsf{st}(\mathcal{K}, G)$ are defined in P, and P weakly satisfies $\mathcal{K}[G] \wedge G$.

In [8] we gave several examples of local theory extensions: any extension of a theory with free functions; extensions with selector functions for a constructor which is injective in the base theory; extensions of $\mathbb{R}$ with a Lipschitz function in a point x_0; extensions of partially ordered theories – in a class Ord consisting of the theories of posets, (dense) totally-ordered sets, semilattices, (distributive) lattices, Boolean algebras, or $\mathbb{R}^\infty$ – with a monotone function f, i.e. satisfying:

$$(\mathsf{Mon}_f) \qquad \bigwedge_{i=1}^{n} x_i \leq y_i \to f(x_1, \dots, x_n) \leq f(y_1, \dots, y_n),$$

Below, we give some additional examples with particular relevance in verification.

Theorem 1 *The following theory extensions are local:*

(1) Extensions of any theory $\mathcal{T}_0$ for which $\leq$ is reflexive with functions satisfying boundedness (Bound^t_f) or guarded boundedness (GBound^t_f) conditions

$$(\mathsf{Bound}^t_f) \qquad \forall x_1, \dots, x_n (f(x_1, \dots, x_n) \leq t(x_1, \dots, x_n))$$
$$(\mathsf{GBound}^t_f) \quad \forall x_1, \dots, x_n (\phi(x_1, \dots, x_n) \to f(x_1, \dots, x_n) \leq t(x_1, \dots, x_n)),$$

where $t(x_1, \dots, x_n)$ is a term in the base signature Π_0 and $\phi(x_1, \dots, x_n)$ a conjunction of literals in the signature Π_0, whose variables are in $\{x_1, \dots, x_n\}$.

(2) Extensions of any theory in Ord *with* $\mathsf{Mon}_f \wedge \mathsf{Bound}^t_f$*, if* $t(x_1, \dots, x_n)$ *is monotone in the variables* $x_1, \dots, x_n$*.*

(3) Extensions of any theory in Ord *with functions satisfying* $\mathsf{Leq}(f, g) \wedge \mathsf{Mon}_f$*.*

$(\mathsf{Leq}(f, g)) \quad \forall x_1, \dots, x_n (\bigwedge_{i=1}^n x_i \leq y_i \rightarrow f(x_1, \dots, x_n) \leq g(y_1, \dots, y_n))$

(4) Extensions of any totally-ordered theory in Ord *with functions satisfying* $\mathsf{SGc}(f, g_1, \dots, g_n) \wedge \mathsf{Mon}(f, g_1, \dots, g_n)$*.*

$(\mathsf{SGc}(f, g_1, \dots, g_n)) \quad \forall x_1, \dots, x_n, x (\bigwedge_{i=1}^n x_i \leq g_i(x) \rightarrow f(x_1, \dots, x_n) \leq x)$

(5) Extensions of theories in Ord *with functions satisfying* $\mathsf{SGc}(f, g_1) \wedge \mathsf{Mon}(f, g_1)$*.*

3.2 Hierarchic reasoning in local theory extensions

Let $\mathcal{T}_0 \subseteq \mathcal{T}_1 = \mathcal{T}_0 \cup \mathcal{K}$ be a local theory extension. To check the satisfiability of a set G of ground clauses w.r.t. $\mathcal{T}_1$ we can proceed as follows (for details cf. [8]):

Step 1: Use locality. By the locality condition, we know that G is unsatisfiable w.r.t. $\mathcal{T}_1$ iff $\mathcal{K}[G] \wedge G$ has no weak partial model in which all the subterms of $\mathcal{K}[G] \wedge G$ are defined, and whose restriction to Π_0 is a total model of $\mathcal{T}_0$.

Step 2: Flattening and purification. As in $\mathcal{K}[G]$ and G the functions in Σ_1 have as arguments only ground terms, $\mathcal{K}[G] \wedge G$ can be purified and flattened by introducing new constants for the arguments of the extension functions as well as for the (sub)terms $t = f(g_1, \dots, g_n)$ starting with extension functions $f \in \Sigma_1$, together with new corresponding definitions $c_t \approx t$. The set of clauses thus obtained has the form $\mathcal{K}_0 \wedge G_0 \wedge D$, where D is a set of ground unit clauses of the form $f(c_1, \dots, c_n) \approx c$, where $f \in \Sigma_1$ and $c_1, \dots, c_n, c$ are constants, and $\mathcal{K}_0, G_0$ are clauses without function symbols in Σ_1.

Step 3: Relational translation. We represent the function symbols in Σ_1 as partial, but functional relations. We thus obtain a relational translation D^* of D, in which each literal $f(c_1, \dots, c_n) \approx c$ is replaced by $r_f(c_1, \dots, c_n, c)$, and corresponding functionality axioms $\mathsf{Fun}(D^*)$ are added.

Step 4: Reduction to testing satisfiability in $\mathcal{T}_0$. We reduce the problem of testing satisfiability of G w.r.t. $\mathcal{T}_1$ to a satisfiability test in $\mathcal{T}_0$ as shown in Theorem 2.

Theorem 2 ([8]) *With the notation above, the following are equivalent:*
(1) $\mathcal{T}_0 \wedge \mathcal{K} \wedge G$ *has a model.*
(2) $\mathcal{T}_0 \wedge \mathcal{K}[G] \wedge G$ *has a weak partial model where all terms in* $\mathsf{st}(\mathcal{K}, \mathsf{G})$ *are defined.*
(3) $\mathcal{T}_0 \wedge \mathcal{K}_0 \wedge G_0 \wedge D$ *has a weak partial model with all terms in* $\mathsf{st}(\mathcal{K}, \mathsf{G})$ *defined.*
(4) $\mathcal{T}_0 \wedge \mathcal{K}_0 \wedge G_0 \wedge \mathsf{Fun}(D^*) \wedge D^*$ *has a relational model, where*

$$\mathsf{Fun}(D^*) = \{\bigwedge_{i=1}^n c_i \approx d_i \wedge r_f(c_1, \dots, c_n, c) \wedge r_f(d_1, \dots, d_n, d) \rightarrow c \approx d \mid f \in \Sigma_1, r_f(c_1, \dots, c_n, c), r_f(d_1, \dots, d_n, d) \in D^*\}.$$

(5) $\mathcal{T}_0 \wedge \mathcal{K}_0 \wedge G_0 \wedge N_0$ *has a (total)* Σ_0*-model, where*

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

Example 1. Let $\mathcal{T}_1 = \mathsf{SLat} \cup \mathsf{SGc}(f, g) \cup \mathsf{Mon}(f, g)$ be the extension of the theory of semilattices with two monotone functions f, g satisfying the semi-Galois condition $\mathsf{SGc}(f, g)$. Consider the ground formulae A,B in the signature of $\mathcal{T}_1$:

$$A: \quad d \leq g(a) \ \wedge \ a \leq c \qquad B: \quad b \leq d \ \wedge \ f(b) \not\leq c.$$

where c and d are shared constants. By Theorem 1(5), $\mathcal{T}_1$ is a local extension of the theory of semilattices. To prove that $A \wedge B \models_{\mathcal{T}_1} \perp$ we proceed as follows:

We purify and flatten the formula $A \wedge B$ by replacing the ground terms starting with f and g with new constants. The clauses are separated into a part containing definitions for terms starting with extension functions, $D_A \wedge D_B$, and a conjunction of formulae in the base signature, $A_0 \wedge B_0$. As the extension $\mathsf{SLat} \subseteq \mathcal{T}_1$ is local, $A \wedge B \models_{\mathcal{T}_1} \perp$ iff $A_0 \wedge B_0 \wedge N_0 \wedge \mathsf{Mon}_0 \wedge \mathsf{SGc}_0$ is unsatisfiable w.r.t. SLat, where N_0 consists of the flattened form of those instances of the congruence axioms containing only f- and g-terms which occur in D_A or D_B, and $\mathsf{Mon}_0 \wedge \mathsf{SGc}_0$ consists of those instances of axioms in $\mathsf{Mon}(f, g) \wedge \mathsf{SGc}(f, g)$ containing only f- and g-terms which occur in D_A or D_B.

Extension	Base	
$D_A \wedge D_B$	$A_0 \wedge B_0 \wedge N_0 \wedge \mathsf{Mon}_0 \wedge \mathsf{SGc}_0$	
$a_1 \approx g(a)$	$A_0 = d \leq a_1 \wedge a \leq c$	$N_A \wedge \mathsf{Mon}_A = a \lhd a \rightarrow a_1 \lhd a_1, \lhd \in \{\approx, \leq\}$
$b_1 \approx f(b)$	$B_0 = b \leq d \wedge b_1 \not\leq c$	$N_B \wedge \mathsf{Mon}_B = b \lhd b \rightarrow b_1 \lhd b_1, \ \lhd \in \{\approx, \leq\}$
		$\mathsf{SGc}_0 = b \leq a_1 \rightarrow b_1 \leq a$

It is easy to see that $A_0 \wedge B_0 \wedge N_0 \wedge \mathsf{Mon}_0 \wedge \mathsf{SGc}_0$ is unsatisfiable w.r.t. $\mathcal{T}_0$: $A_0 \wedge B_0$ entails $b \leq a_1$, together with SGc_0 this yields $b_1 \leq a$, which together with $a \leq c$ and $b_1 \not\leq c$ leads to a contradiction.

4 A hierarchical interpolation procedure

Let $\mathcal{T}_0 \subseteq \mathcal{T}_1 = \mathcal{T}_0 \cup \mathcal{K}$ be a theory extension by means of a set of clauses $\mathcal{K}$. Assume that $A \wedge B \models_{\mathcal{T}_1} \perp$, where A and B are two sets of ground clauses. Our goal is to find a ground *interpolant*, that is a ground formula I containing only constants and extension functions which are common to A and B such that

$$A \models_{\mathcal{T}_1} I \quad \text{ and } \quad I \wedge B \models_{\mathcal{T}_1} \perp.$$

Flattening and purification do not influence the existence of ground interpolants:

Lemma 3 *Let A and B be two sets of ground clauses in the signature Π^c. Let $A_0 \wedge D_A$ and $B_0 \wedge D_B$ be obtained from A resp. B by purification and flattening. If I is an interpolant of $(A_0 \wedge D_A) \wedge (B_0 \wedge D_B)$ then the formula $\overline{I}$, obtained from I by replacing, recursively, all newly introduced constants with the terms in the original signature which they represent, is an interpolant for $A \wedge B$.*

Therefore we can restrict without loss of generality to finding interpolants for the *purified and flattened* conjunction of formulae $(A_0 \wedge D_A) \wedge (B_0 \wedge D_B)$.

We focus on interpolation in *local theory extensions.* Let $\mathcal{T}_0 \subseteq \mathcal{T}_1 = \mathcal{T}_0 \cup \mathcal{K}$ be a local theory extension. From Theorem 2 we know that in such extensions hierarchical reasoning is possible [8]: if A and B are sets of ground clauses in a signature Π^c, and $A_0 \wedge D_A$ (resp. $B_0 \wedge D_B$) are obtained from A (resp. B) by purification and flattening then:

$$(A_0 \wedge D_A) \wedge (B_0 \wedge D_B) \models_{\mathcal{T}_1} \perp \quad \text{iff} \quad \mathcal{K}_0 \wedge A_0 \wedge B_0 \wedge N_0 \models_{\mathcal{T}_0} \perp,$$

where $\mathcal{K}_0$ is obtained from $\mathcal{K}[D_A \wedge D_B]$ by replacing the Σ_1-terms with the corresponding constants contained in the definitions D_A and D_B and

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

In general we cannot use Theorem 2 for generating a ground interpolant because:

(i) $\mathcal{K}[D_A \wedge D_B]$ (hence also $\mathcal{K}_0$) may contain free variables.
(ii) The clauses in $\mathcal{K}[D_A \wedge D_B]$ and the instances of congruence axioms (and therefore the clauses in $\mathcal{K}_0 \wedge N_0$) may contain combinations of constants and extension functions from A and B.
(iii) If some clause in $\mathcal{K}$ contains two or more different extension functions, it is unlikely that these extension functions can be separated in the interpolants.

To solve (iii), we define a relation $\sim$ between extension functions, where $f \sim g$ if f and g occur in the same clause in $\mathcal{K}$. This defines an equivalence relation $\sim$ on Σ_1. We henceforth consider that a function $f \in \Sigma_1$ is common to A and B if there exist $g, h \in \Sigma_1$ such that $f \sim g$, $f \sim h$, g occurs in A and h occurs in B.

Example 2. Consider the reduction to the base theory in Example 1. *Ad* (ii): The clause $b \leq a_1 \rightarrow b_1 \leq a$ of SGc_0 is mixed, i.e. contains combinations of constants from A and B. *Ad* (iii): As $\mathsf{SGc}(f, g)$ contains occurrences of both f and g, it is not likely to find an interpolant with no occurrence of f and g, even if g only occurs in A and f only occurs in B. We assume that both f and g are shared.

4.1 Main Idea

The idea of our approach is to separate mixed instances of axioms in $\mathcal{K}_0$, or of congruence axioms in N_0, into an A-part and a B-part. This is, if $A \wedge B \models_{\mathcal{T}_1} \perp$ we find a set T of $\Sigma_0 \cup \Sigma_1$-terms containing only constants and extension functions common to A and B, such that $\mathcal{K}[A \wedge B]$ can be separated into a part $\mathcal{K}[A, T]$ consisting of instances with extension terms occurring in A and T, and a part $\mathcal{K}[B, T]$ containing only instances with extension terms in B and T, such that:

$$\mathcal{K}[A, T] \wedge A_0 \wedge \mathsf{Fun}((D_A \wedge D_T)^*) \wedge \mathcal{K}[B, T] \wedge B_0 \wedge \mathsf{Fun}((D_B \wedge D_T)^*)$$

has no weak partial model where all ground terms in $\mathcal{K}, D_A, D_B, T$ are defined.

Example 3. Consider the conjunction $A_0 \wedge D_A \wedge B_0 \wedge D_B \wedge N_0 \wedge \mathsf{Mon}_0 \wedge \mathsf{SGc}_0$ in Example 1. We obtain a separation for the clause $b \leq a_1 \rightarrow b_1 \leq a$ of SGc_0 as follows: Note that $A_0 \wedge B_0 \models b \leq a_1$. We can find an SLat-term t containing only shared constants of A_0 and B_0 such that $A_0 \wedge B_0 \models b \leq t \wedge t \leq a_1$. (Indeed, such a term is $t = d$.) We show that, instead of the axiom $a \leq g(b) \rightarrow f(a) \leq b$, whose flattened form is in SGc_0, we can use, without loss of unsatisfiability:

(1) an instance of the monotonicity axiom for f: $b \leq d \rightarrow f(b) \leq f(d)$, and
(2) another instance of SGc, namely: $d \leq g(a) \rightarrow f(d) \leq a$.

We introduce a new constant $c_{f(d)}$ for $f(d)$ (its definition, $c_{f(d)} \approx f(d)$, is stored in a set D_T), and the corresponding instances $\mathcal{H}_{\mathsf{sep}} = \mathcal{H}^A_{\mathsf{sep}} \wedge \mathcal{H}^B_{\mathsf{sep}}$ of the congruence, monotonicity and $\mathsf{SGc}(f, g)$-axioms, which are now separated into an A-part ($d \leq a_1 \rightarrow c_{f(d)} \leq a$) and a B-part ($b \leq d \rightarrow b_1 \leq c_{f(d)}$). We thus obtain a separated conjunction $\overline{A}_0 \wedge \overline{B}_0$ (where $\overline{A}_0 = \mathcal{H}^A_{\mathsf{sep}} \wedge A_0$ and $\overline{B}_0 = \mathcal{H}^B_{\mathsf{sep}} \wedge B_0$), which can be proved to be unsatisfiable in $\mathcal{T}_0 = \mathsf{SLat}$. To compute an interpolant in SLat for $\overline{A}_0 \wedge \overline{B}_0$ note that $\overline{A}_0$ is logically equivalent to the conjunction of unit literals $d \leq a_1 \wedge a \leq c \wedge c_{f(d)} \leq a$ and $\overline{B}_0$ is logically equivalent to $b \leq d \wedge b_1 \not\leq c \wedge b_1 \leq c_{f(d)}$. An interpolant is $I_0 = c_{f(d)} \leq c$. By replacing the new constants with the terms they denote we obtain the interpolant $I = f(d) \leq c$ for $A \wedge B$.

4.2 Examples of theory extensions with hierarchic interpolation

We identify a class of theory extensions for which interpolants can be computed hierarchically (and efficiently) using a procedure for generating interpolants in the base theory $\mathcal{T}_0$. This allows us to exploit specific properties of $\mathcal{T}_0$ for obtaining simple interpolants in $\mathcal{T}_1$. We make the following assumptions about $\mathcal{T}_0$:

Assumption 1: $\mathcal{T}_0$ is *convex* with respect to the set Pred of all predicates (including equality $\approx$), i.e., for all conjunctions Γ of ground atoms, relations $R_1, \dots, R_m \in \mathsf{Pred}$ and ground tuples of corresponding arity $\overline{t}_1, \dots, \overline{t}_n$, if $\Gamma \models_{\mathcal{T}_0} \bigvee_{i=1}^m R_i(\overline{t}_i)$ then there exists $j \in \{1, \dots, m\}$ such that $\Gamma \models_{\mathcal{T}_0} R_j(\overline{t}_j)$.

Assumption 2: $\mathcal{T}_0$ is *P-interpolating*, i.e. for all conjunctions A and B of ground literals, all binary predicates $R \in P$ and all constants a and b such that a occurs in A and b occurs in B (or vice versa), if $A \wedge B \models_{\mathcal{T}_0} aRb$ then there exists a term t containing only constants common to A and B with $A \wedge B \models_{\mathcal{T}_0} aRt \wedge tRb$. (If we can always guarantee that $A \models_{\mathcal{T}_0} aRt$ and $B \models_{\mathcal{T}_0} tRb$ we say that $\mathcal{T}_0$ is *strongly P-interpolating*.)

Assumption 3: $\mathcal{T}_0$ has ground interpolation.

Some examples of theories satisfying these properties are given below.

Theorem 4 *The following theories have ground interpolation and are convex and P-interpolating w.r.t. the indicated set P of predicate symbols:*

(1) The theory of $\mathcal{EQ}$ of pure equality without function symbols (for $P = \{\approx\}$).
(2) The theory PoSet of posets (for $P = \{\approx, \leq\}$).
(3) Linear rational arithmetic $\mathsf{LI}(\mathbb{Q})$ and linear real arithmetic $\mathsf{LI}(\mathbb{R})$ (convex w.r.t. $P = \{\approx\}$, P-interpolating w.r.t. $P = \{\approx, \leq\}$, strongly P-interpolating w.r.t. $P = \{\leq\}$).

(4) The theories Bool *of Boolean algebras,* SLat *of semilattices and* DLat *of distributive lattices (for* $P = \{\approx, \leq\}$*).*

For the sake of simplicity we only consider sets A, B of unit clauses, i.e. conjunctions of ground literals. This is not a restriction, since if we can obtain interpolants for conjunctions of ground literals then we also can construct interpolants for conjunctions of arbitrary clauses by using standard methods[1] discussed e.g. in [4] or [10]. By Lemma 3, we can restrict w.l.o.g. to finding an interpolant for the purified and flattened conjunction of unit clauses $A_0 \wedge B_0 \wedge D_A \wedge D_B$.

By Theorem 2, $A_0 \wedge D_A \wedge B_0 \wedge D_B \models_{\mathcal{T}_1} \perp$ iff $\mathcal{K}_0 \wedge A_0 \wedge B_0 \wedge N_0 \models_{\mathcal{T}_0} \perp$, where $\mathcal{K}_0$ is obtained from $\mathcal{K}[D_A \wedge D_B]$ by replacing the Σ_1-terms with the corresponding constants contained in the definitions $D_A \wedge D_B$ and

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

In general, $N_0 = N_0^A \wedge N_0^B \wedge N_{\mathsf{mix}}$ and $\mathcal{K}_0 = \mathcal{K}_0^A \wedge \mathcal{K}_0^B \wedge \mathcal{K}_{\mathsf{mix}}$, where $N_0^A, \mathcal{K}_0^A$ only contain extension functions and constants which occur in A, $N_0^B, \mathcal{K}_0^B$ only contain extension functions and constants which occur in B, and N_{mix}, $\mathcal{K}_{\mathsf{mix}}$ contain mixed clauses with constants occurring in both A and B. Our goal is to separate N_{mix} and $\mathcal{K}_{\mathsf{mix}}$ into an A-local and a B-local part. We show that, under Assumptions 1 and 2, N_{mix} can always be separated, and $\mathcal{K}_{\mathsf{mix}}$ can be separated if $\mathcal{K}$ contains the following type of combinations of clauses:

$$\begin{cases} x_1 \, R_1 \, s_1 \wedge \cdots \wedge x_n \, R_n \, s_n \rightarrow f(x_1, \ldots, x_n) \, R \, g(y_1, \ldots, y_n) \\ x_1 \, R_1 \, y_1 \wedge \cdots \wedge x_n \, R_n \, y_n \rightarrow f(x_1, \ldots, x_n) \, R \, f(y_1, \ldots, y_n) \end{cases} \tag{3}$$

where $n \geq 1$, $x_1, \ldots, x_n$ are variables, $R_1, \ldots, R_n, R$ are binary relations with $R_1, \ldots, R_n \in P$ and R transitive, and each s_i is either a variable among the arguments of g, or a term of the form $f_i(z_1, \ldots, z_k)$, where $f_i \in \Sigma_1$ and all the arguments of f_i are variables occurring among the arguments of g. [2]

Example 4. The following local extensions are in the class above:

(a) Any extension with free functions ($\mathcal{K} = \emptyset$).
(b) Extensions of any theory in Ord (cf. Section 3.1) with monotone functions.
(c) Extensions of any totally-ordered theory in Ord with functions satisfying $\mathsf{SGc}(f, g_1, \ldots, g_n) \wedge \mathsf{Mon}(f, g_1, \ldots, g_n)$.
(d) Extensions of theories in Ord with functions satisfying $\mathsf{SGc}(f, g_1) \wedge \mathsf{Mon}(f, g_1)$.
(e) Extensions of theories in Ord with functions satisfying $\mathsf{Leq}(f, g) \wedge \mathsf{Mon}_f$.

Note: If the clauses in $\mathcal{K}$ are of type (3), then (i) the cardinality of $\mathcal{K}_0 \cup N_0$ is quadratic in the size of $A \wedge B$, (ii) all clauses in $\mathcal{K}_0$ are of the form $C = \bigwedge_{i=1}^{n} c_i \, R_i \, d_i \rightarrow c \, R \, d$, where $R_i \in P$, R is transitive, and c_i, d_i, c, d are constants.

[1] E.g. in a DPLL-style procedure partial interpolants are generated for the unsatisfiable branches and then recombined using ideas of Pudlák.

[2] More general types of clauses, in which instead of variables we can consider arbitrary base terms, can be handled if $\mathcal{T}_0$ has a P-interpolation property for terms instead of constants. Due to space limitations, such extensions are not discussed here.

Proposition 5 *Assume that $\mathcal{T}_0$ satisfies Assumptions 1 and 2. Let $\mathcal{H}$ be a set of Horn clauses $\bigwedge_{i=1}^n c_i R_i d_i \rightarrow cRd$ in the signature Π_0^c (with R transitive and $R_i \in P$) which are instances of flattened and purified clauses of type (3) and of congruence axioms. Let A_0 and B_0 be conjunctions of ground literals in the signature Π_0^c such that $A_0 \wedge B_0 \wedge \mathcal{H} \models_{\mathcal{T}_0} \perp$. Then there exists a set T of $\Sigma_0 \cup \Sigma_c$-terms containing only constants common to A_0 and B_0 such that $A_0 \wedge B_0 \wedge (\mathcal{H} \backslash \mathcal{H}_{\mathsf{mix}}) \wedge \mathcal{H}_{\mathsf{sep}} \models_{\mathcal{T}_0} \perp$, where*

$$\begin{aligned}
\mathcal{H}_{\mathsf{mix}} = \{&\textstyle\bigwedge_{i=1}^n c_i R_i d_i \rightarrow cRd \in \mathcal{H} \mid c_i, c \text{ constants in } A, d_i, d \text{ constants in } B\} \cup \\
\{&\textstyle\bigwedge_{i=1}^n c_i R_i d_i \rightarrow cRd \in \mathcal{H} \mid c_i, c \text{ constants in } B, d_i, d \text{ constants in } A\} \\
\mathcal{H}_{\mathsf{sep}} = \{&(\textstyle\bigwedge_{i=1}^n c_i R_i t_i \rightarrow cRc_{f(t_1,\dots,t_n)}) \wedge (\bigwedge_{i=1}^n t_i R_i d_i \rightarrow c_{f(t_1,\dots,t_n)} Rd) \mid \\
&\textstyle\bigwedge_{i=1}^n c_i R_i d_i \rightarrow cRd \in \mathcal{H}_{\mathsf{mix}}, d_i \approx s_i(e_1,\dots,e_n), d \approx g(e_1,\dots,e_n) \in D_B, \\
&c \approx f(c_1,\dots,c_n) \in D_A \text{ or vice versa } \} = \mathcal{H}_{\mathsf{sep}}^A \wedge \mathcal{H}_{\mathsf{sep}}^B
\end{aligned}$$

and $c_{f(t_1,\dots,t_n)}$ are new constants in Σ_c (considered common to A_0, B_0) introduced for the terms $f(t_1,\dots,t_n)$.

Proof (Sketch): Proof by induction on the number of clauses in $\mathcal{H}$. Convexity and P-interpolation ensure that for each clause $C = \bigwedge c_i R_i d_i \rightarrow cRd \in \mathcal{H}_{\mathsf{mix}}$, e.g. obtained from the following instance of a clause of type (3):

$$c_1 R_1 s_1(e_1,\dots,e_m) \wedge \dots \wedge c_n R_n s_n(e_1,\dots,e_m) \rightarrow f(c_1,\dots,c_n) R g(e_1,\dots,e_m)$$

there exist terms $t_1,\dots,t_n$ containing only constants common to A_0 and B_0 such that $A_0 \wedge B_0 \wedge (\mathcal{H} \backslash \{C\}) \models_{\mathcal{T}_0} c_i R_i t_i \wedge t_i R_i d_i$. We thus can replace C with the conjunction of an instance of the monotonicity axiom, $C_A : \bigwedge_{i=1}^n c_i R_i t_i \rightarrow cRc_{f(t_1,\dots,t_n)}$ and an instance of the clause of type (3), $C_B : \bigwedge_{i=1}^n t_i R_i d_i \rightarrow c_{f(t_1,\dots,t_n)} Rd$. □

In what follows we assume that $\mathcal{K}$ only contains combinations of clauses of type (3). An immediate consequence of Proposition 5 is Theorem 6.

Theorem 6 *Assume $\mathcal{T}_0$ satisfies Assumptions 1, 2 and $\mathcal{K}_0 \wedge A_0 \wedge B_0 \wedge N_0 \models_{\mathcal{T}_0} \perp$. Then there exists a set T of $\Sigma_0 \cup \Sigma_c$-terms containing only constants common to A_0 and B_0 such that (if $N_0^D = N_0^{DA} \wedge N_0^{DB} = N_{0\mathsf{sep}}$ and $\mathcal{K}_0^D = \mathcal{K}_0^{DA} \wedge \mathcal{K}_0^{DB} = \mathcal{K}_{0\mathsf{sep}}$):*

$$\mathcal{K}_0^A \wedge \mathcal{K}_0^B \wedge \mathcal{K}_0^D \wedge A_0 \wedge B_0 \wedge N_0^A \wedge N_0^B \wedge N_0^D \models_{\mathcal{T}_0} \perp. \tag{4}$$

(As before, Σ_c contains the new constants $c_{f(t_1,\dots,t_n)}$, considered to be common to A_0 and B_0, introduced for terms $f(t_1,\dots,t_n)$, with $t_1,\dots,t_n \in T$.)

Corollary 7 *Assume $\mathcal{T}_0$ satisfies Assumptions 1–3 and $\mathcal{K}_0 \wedge A_0 \wedge B_0 \wedge N_0 \models_{\mathcal{T}_0} \perp$.*

(1) There exists a Π_0^c-formula I_0 containing only constants common to A_0, B_0 with $\mathcal{K}_0^A \wedge \mathcal{K}_0^{DA} \wedge A_0 \wedge N_0^A \wedge N_0^{DA} \models_{\mathcal{T}_0} I_0$ and $\mathcal{K}_0^B \wedge \mathcal{K}_0^{DB} \wedge B_0 \wedge N_0^B \wedge N_0^{DB} \wedge I_0 \models_{\mathcal{T}_0} \perp$.

(2) There exists a ground Π^c-formula I containing only constants and function symbols which occur both in A and B such that $A \models_{\mathcal{T}_1} I$ and $B \wedge I \models_{\mathcal{T}_1} \perp$.

Proof: If $\mathcal{T}_0$ has ground interpolation, (1) is a direct consequence of Theorem 6, since $\mathcal{K}_0^A, \mathcal{K}_0^{AD}, \mathcal{K}_0^B, \mathcal{K}_0^{BD}$ are ground. (2) Let I be obtained from I_0 by recursively replacing each constant c_t introduced in the separation process with the term t. Then I is an interpolant of $(A_0 \wedge D_A) \wedge (B_0 \wedge D_B)$. □

We obtain the following procedure for computing interpolants for $A \wedge B$:

Preprocess: Using locality, flattening and purification we obtain a set $\mathcal{H} \wedge A_0 \wedge B_0$ of formulae in the base theory, where $\mathcal{H}$ is as in Prop. 5. Let $\Delta := \mathsf{T}$.

Repeat as long as possible: Let $C \in \mathcal{H}$ whose premise is entailed by $A_0 \wedge B_0 \wedge \Delta$. If C is mixed, compute terms t_i which separate the premises in C, and separate the clause into an instance C_1 of monotonicity and an instance C_2 of a clause in $\mathcal{K}$ as in Prop. 5. Remove C from $\mathcal{H}$, and add C_1, C_2 to $\mathcal{H}_{\mathsf{sep}}$ and their conclusions to Δ. Otherwise move C to $\mathcal{H}_{\mathsf{sep}}$ and add its conclusion to Δ.

Compute interpolant in $\mathcal{T}_0$ for the separated formula obtained this way, and construct an interpolant for the extension as explained in Corollary 7(2).

Theorem 8 *Assume that the cycle of the procedure above stops after moving the processed clauses $\mathcal{H}_{\mathsf{proc}}$ into the set $\mathcal{H}_{\mathsf{sep}}$. The following are equivalent:*

(1) $A_0 \wedge D_A \wedge B_0 \wedge D_B \models_{\mathcal{T}_1} \perp$. *(2)* $A_0 \wedge B_0 \wedge (\mathcal{H} \backslash \mathcal{H}_{\mathsf{proc}}) \wedge \mathcal{H}_{\mathsf{sep}} \models_{\mathcal{T}_0} \perp$.

Proof: (1)$\Rightarrow$(2) is a consequence of Theorems 2 and 6. As the conjunction in (2) corresponds to a subset of instances of $\mathcal{K} \wedge A_0 \wedge D_A \wedge B_0 \wedge D_B$, (2) imples (1). $\Box$

Note: If $\mathcal{K}_0 \wedge A_0 \wedge B_0 \wedge N_0 \models_{\mathcal{T}_0} \perp$ then no matter which terms are chosen for separating mixed clauses in $N_0 \wedge \mathcal{K}_0$, we obtain a separated conjunction of clauses unsatisfiable w.r.t. $\mathcal{T}_0$. Theorem 8 shows that if the set of clauses obtained when the procedure stops is satisfiable then $A \wedge B$ was satisfiable, and conversely, so the procedure can be used to test satisfiability and to compute interpolants at the same time. (However, it is more efficient to first test $A \wedge B \models_{\mathcal{T}_1} \perp$.)

Complexity: Assume that in $\mathcal{T}_0$ for a formula of length n (a) interpolants can be computed in time $g(n)$, (b) P-interpolating terms can be computed in time $h(n)$, (c) entailment can be checked in time $k(n)$. The size n of the set of clauses obtained after the preprocessing phase is quadratic in the size of the input. The procedure above computes an interpolant in time of order $n \cdot (k(n) + h(n)) + g(n)$.

Remark 9 *If $\mathcal{T}_0$ satisfies Assumptions 1,3 and is strongly P-interpolating, the procedure above can be changed to separate* all *clauses in $\mathcal{H}$ and store the conclusions of the separated clauses in $\Delta = \Delta_A \cup \Delta_B$. If $\mathcal{K}_0 \wedge A_0 \wedge B_0 \wedge N_0 \models_{\mathcal{T}_0} \perp$ then there exists a set T of $\Sigma_0 \cup \Sigma_c$-terms containing only constants common to A_0 and B_0, and common new constants in a set Σ_c such that the terms in T can be used to separate $N_0 \cup \mathcal{K}_0$ into $H_{\mathsf{sep}} = (\mathcal{K}_0^{DA} \wedge N_0^{DA}) \wedge (\mathcal{K}_0^{DB} \wedge N_0^{DB})$, where:*

$$H_{\mathsf{sep}} = \{(\bigwedge_{i=1}^n c_i R_i t_i \rightarrow c R c_{f(t_1,\dots,t_n)}) \wedge (\bigwedge_{i=1}^n t_i R_i d_i \rightarrow c_{f(t_1,\dots,t_n)} R d) \mid \bigwedge_{i=1}^n c_i R_i d_i \rightarrow c R d \in N_0 \cup \mathcal{K}_0\} = (\mathcal{K}_0^{DA} \wedge N_0^{DA}) \wedge (\mathcal{K}_0^{DB} \wedge N_0^{DB})$$

such that for each premise $c_i R_i d_i$ of a rule in $N_0 \cup \mathcal{K}_0$, at some step in the procedure $A_0 \wedge B_0 \wedge \Delta_A \wedge \Delta_B \models c_i R_i d_i$ and there exists $t_i \in T$ such that $A_0 \wedge \Delta_A \models c_i R_i t_i$ and $B_0 \wedge \Delta_B \models t_i R_i d_i$. In this case $A_0 \wedge \mathcal{K}_0^{DA} \wedge N_0^{DA}$ is logically equivalent to $\overline{A}_0$, and $B_0 \wedge \mathcal{K}_0^{DB} \wedge N_0^{DB}$ is logically equivalent to $\overline{B}_0$, where $\overline{A}_0$, $\overline{B}_0$ are the following conjunctions of literals:

$$\overline{A}_0 = A_0 \wedge \bigwedge\{c R c_{f(\overline{t})} \mid \text{ conclusion of some clause } (\Gamma \rightarrow c R c_{f(\overline{t})}) \in \mathcal{K}_0^{DA} \cup N_0^{DA}\}$$
$$\overline{B}_0 = B_0 \wedge \bigwedge\{c_{f(\overline{t})} R d \mid \text{ conclusion of some clause } (\Gamma \rightarrow c_{f(\overline{t})} R d) \in \mathcal{K}_0^{DB} \cup N_0^{DB}\}.$$

Thus, if for instance in $\mathcal{T}_0$ interpolants for conjunctions of ground literals are again conjunctions of ground literals, the same is also true in the extension.

Example 5. The following theory extensions have ground interpolation:

(a) Extensions of any theory in Theorem 4(1)–(4) with free function symbols.
(b) Extensions of the theories in Theorem 4(2),(4) with monotone functions.
(c) Extensions of the theories in Theorem 4(2),(4) with $\mathsf{Leq}(f, g) \wedge \mathsf{Mon}_f$.
(d) Extensions of the theories in Theorem 4(2),(4) with $\mathsf{SGc}(\mathsf{f}, \mathsf{g_1}) \wedge \mathsf{Mon}(f, g_1)$.
(e) Extensions of any theory in Theorem 4(1)–(4) with Bound^t_f or GBound^t_f (where t is a term and ϕ a set of literals in the base theory).
(f) Extensions of the theories in Theorem 4(2),(4) with $\mathsf{Mon}_f \wedge \mathsf{Bound}^t_f$, if t is monotone in its variables.
(g) $\mathbb{R} \cup (\mathsf{L}^\lambda_f)$, the extension of the theory of reals with a unary function which is λ-Lipschitz in a point x_0, where (L^λ_f) is $\forall x \ |f(x) - f(x_0)| \leq \lambda \cdot |x - x_0|$.

Proof: (a)–(d) are direct consequences of Corollary 7, since all sets of extension clauses are of type (3). For extensions of linear arithmetic note that due to the totality of $\leq$ we can always assume that A and B are positive, so convexity w.r.t. $\approx$ is sufficient (cf. proof of Proposition 5). (e)–(g) follow from Corollary 7 and the fact that if each clause in $\mathcal{K}$ contains only one occurrence of an extension function, no mixed instances can be generated when computing $\mathcal{K}[A \wedge B]$. □

4.3 Applications

Modular reasoning in local combinations of theories. Let $\mathcal{T}_i{=}\mathcal{T}_0{\cup}\mathcal{K}_i, i{=}1, 2$ be local extensions of a theory $\mathcal{T}_0$ with signature $\Pi_0{=}(\Sigma_0, \mathsf{Pred})$, where $\Sigma_0 = \Sigma_1{\cap}\Sigma_2$. Assume that (a) all variables in $\mathcal{K}_i$ occur below some extension function, (b) the extension $\mathcal{T}_0 \subseteq \mathcal{T}_0{\cup}\mathcal{K}_1{\cup}\mathcal{K}_2$ is local[3], and (c) $\mathcal{T}_0$ has ground interpolation.

Let G be a ground clause in the signature $\Pi^c = (\Sigma_0 \cup \Sigma_1 \cup \Sigma_2 \cup \Sigma_c, \mathsf{Pred})$. G can be flattened and purified, so we assume w.l.o.g. that $G = G_1 \wedge G_2$, where G_1, G_2 are flat and linear sets of clauses in the signatures Π_1, Π_2 respectively, i.e. for $i = 1, 2$, $G_i = G^0_i \wedge G_0 \wedge D_i$, where G^0_i and G_0 are clauses in the base theory and D_i conjunctions of unit clauses of the form $f(c_1, \dots, c_n) = c, f \in \Sigma_i$.

Theorem 10 *With the notations above, assume that $G_1 \wedge G_2 \models_{\mathcal{T}_1 \cup \mathcal{T}_2} \perp$. Then there exists a ground formula I, containing only constants shared by G_1 and G_2, with $G_1 \models_{\mathcal{T}_1 \cup \mathcal{T}_2} I$ and $I \wedge G_2 \models_{\mathcal{T}_1 \cup \mathcal{T}_2} \perp$.*

Proof: By Theorem 2, the following are equivalent:

(1) $\mathcal{T}_0 \cup \mathcal{K}_1 \cup \mathcal{K}_2 \cup (G^0_1 \wedge G_0 \wedge D_1) \wedge (G^0_2 \wedge G_0 \wedge D_2) \models \perp$,
(2) $\mathcal{T}_0 \cup \mathcal{K}_1[G_1] \wedge \mathcal{K}_2[G_2] \wedge (G^0_1 \wedge G_0 \wedge D_1) \wedge (G^0_2 \wedge G_0 \wedge D_2) \models \perp$,
(3) $\mathcal{K}^0_1 \wedge \mathcal{K}^0_2 \wedge (G^0_1 \wedge G_0) \wedge (G^0_2 \wedge G_0) \wedge N_1 \wedge N_2 \models_{\mathcal{T}_0} \perp$, where, for $j = 1, 2$,

$$N_j = \bigwedge \{ \bigwedge_{i=1}^{n} c_i \approx d_i \rightarrow c = d \mid f(c_1, \dots, c_n) \approx c, f(d_1, \dots, d_n) \approx d \in D_j \},$$

and $\mathcal{K}^0_i$ is the formula obtained from $\mathcal{K}_i[G_i]$ after purification and flattening, taking into account the definitions from D_i. Let $A = \mathcal{K}^0_1 \wedge (G^0_1 \wedge G_0) \wedge N_1$ and

[3] If $\mathcal{T}_0$ is a $\forall\exists$ theory then (b) is implied by (a) and the locality of $\mathcal{T}_1, \mathcal{T}_2$ [6].

$B = \mathcal{K}_2^0 \wedge (G_2^0 \wedge G_0) \wedge N_2$. By assumption (a), A and B are both ground. As A and B have no function symbols in common and only share the constants which G_1 and G_2 share, there exists an interpolant I_0 in the signature Π_0, containing only Σ_0-function symbols and only constants shared by G_1, G_2, such that $A \models_{\mathcal{T}_0} I_0$ and $B \wedge I_0 \models_{\mathcal{T}_0} \perp$. An interpolant for $G_1 \wedge G_2$ w.r.t. $\mathcal{T}_1$ can now be obtained by replacing the newly introduced constants by the terms they replaced. □

By Remark 9, if $\mathcal{T}_0$ is strongly P-interpolating and has equational interpolation then I is a conjunction of literals, so for modularly proving $G_1 \wedge G_2 \models_{\mathcal{T}_1} \perp$ only conjunctions of ground literals containing constants shared by G_1, G_2 need to be exchanged between specialized provers for $\mathcal{T}_1$ and $\mathcal{T}_2$.

Verification. Consider the example presented in the verification example in Section 1.1. We illustrate our method for generating interpolants for a formula corresponding to a path of length 2 from an initial state to an unsafe state:

$$\begin{aligned} G = {} & l < L_{\mathsf{alarm}} \ \wedge \ l' \approx \mathsf{in}(l, t) \ \wedge \ t' \approx k(l) \ \wedge \\ & l' \geq L_{\mathsf{alarm}} \ \wedge \ l'' \approx \mathsf{in}(\mathsf{out}(l', t'), g(t')) \ \wedge \ t'' \approx h(g(t')) \ \wedge \ \neg l'' \leq L_{\mathsf{overflow}} \end{aligned}$$

Hierarchic reasoning. The extension $\mathcal{T}_1$ of linear arithmetic with the clauses (2) in Section 1.1 is local, so to prove $G \models_{\mathcal{T}_1} \perp$ it is sufficient to consider ground instances $\mathcal{K}[G]$ of (2) in which all extension terms already occur in G: After flattening and purifying $\mathcal{K}[G] \wedge G$, we separate the problem into an extension part Ext and a base part Base. By Theorem 2 [8], the problem can be reduced to testing the satisfiability in the base theory of the conjunction $\mathsf{Base} \wedge N_0$. As this conjunction is unsatisfiable w.r.t. $\mathcal{T}_0$, G is unsatisfiable.

Ext		$\mathsf{Base} \wedge N_0$		
$c \approx \mathsf{in}(l, t)$	$d \approx k(t)$	$l < L_{\mathsf{alarm}}$	$l' \approx c$	$\mathcal{K}_0 : l < L_{\mathsf{alarm}} \rightarrow c < L_{\mathsf{overflow}}$
$l_o \approx \mathsf{out}(l', t')$	$t_o \approx g(t')$	$l' \geq L_{\mathsf{alarm}}$	$t' \approx d$	$l_o < L_{\mathsf{alarm}} \rightarrow c' < L_{\mathsf{overflow}}$
$c' \approx \mathsf{in}(l_o, t_o)$	$d' \approx h(t_o)$	$\neg l'' \leq L_{\mathsf{overflow}}$	$l'' \approx c'$	$l' < L_{\mathsf{overflow}} \rightarrow l_o < L_{\mathsf{alarm}}$
			$t'' \approx d'$	$N_0 : l \approx l_o \wedge t \approx t_o \rightarrow c \approx c'$

Interpolation. Let $A = l < L_{\mathsf{alarm}} \wedge l' \approx \mathsf{in}(l, t) \wedge t' \approx k(l)$ and $B = l' \geq L_{\mathsf{alarm}} \wedge l'' \approx \mathsf{in}(\mathsf{out}(l', t'), g(t')) \wedge t'' \approx h(g(t')) \wedge \neg l'' \leq L_{\mathsf{overflow}}$. To generate an interpolant for $A \wedge B$, we partition the clauses in Base as $A_0 \wedge B_0$, where $A_0 = l < L_{\mathsf{alarm}} \wedge l' \approx c \wedge t' \approx d \wedge \mathcal{K}_0^A$ and $B_0 = l' \geq L_{\mathsf{alarm}} \wedge l'' \approx c' \wedge t'' \approx d' \wedge \neg l'' \leq L_{\mathsf{overflow}} \wedge \mathcal{K}_0^B$. The clause in N_0 is mixed. Since already the conjunction of the formulae in Base is unsatisfiable, N_0 is not needed to prove unsatisfiability. The conjunction of the formulae in Base is equivalent to $A_0' \wedge B_0'$, where $A_0' = l < L_{\mathsf{alarm}} \wedge l' \approx c \wedge t' \approx d \wedge c < L_{\mathsf{overflow}}$ and $B_0' = l' > L_{\mathsf{alarm}} \wedge l'' \approx c' \wedge t'' \approx d' \wedge \neg l'' \leq L_{\mathsf{overflow}} \wedge c' < L_{\mathsf{overflow}} \wedge l_o < L_{\mathsf{alarm}}$. The interpolant for $A_0' \wedge B_0'$ is $L' < L_{\mathsf{overflow}}$, which is also an interpolant for $A \wedge B$.

For the database example in Section 1.1 the interpolant is computed similarily.

5 Conclusions

We presented a method for obtaining simple interpolants in theory extensions. We identified situations in which it is possible to do this in a hierarchical manner, by using a prover and a procedure for generating interpolants in the base theory as "black-boxes". This allows us to use the properties of $\mathcal{T}_0$ (e.g. the form of interpolants) to control the form of interpolants in the extension $\mathcal{T}_1$. We discussed applications of interpolation in verification and knowledge representation.

The method we presented is more general than the results of McMillan [4] on interpolation in extension of linear rational arithmetic with uninterpreted function symbols. Our method is orthogonal to the method for generating interpolants for combinations of theories over disjoint signatures from Nelson-Oppen-style unsatisfiability proofs proposed by Yorsh and Musuvathi in [10], as it allows us to consider combinations of theories over non-disjoint signatures.

The hierarchical interpolation method presented here (for the special case of the extension of linear arithmetic with free function symbols) was implemented by Andrey Rybalchenko in Prolog. First tests suggest that our method is considerably faster than other existing methods. Details about the implementation and benchmarks for the special case of linear arithmetic + free function symbols are the subject of a separate joint paper.

Acknowledgements. I thank Andrey Rybalchenko for interesting discussions. 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.

References

1. P.D. Bacsich. Amalgamation properties and interpolation theorem for equational theories. *Algebra Universalis*, 5:45–55, 1975.
2. B Jónsson. Extensions of relational structures. In J.W. Addison, L. Henkin, and A. Tarski, editors, *The Theory of Models, Proc. of the 1963 Symposium at Berkeley*, pages 146–157, Amsterdam, 1965. North-Holland.
3. K.L. McMillan. Interpolation and SAT-based model checking. In *CAV'2003: Computer Aided Verification, LNCS 2725*, pages 1–13. Springer, 2003.
4. K.L. McMillan. An interpolating theorem prover. In *TACAS'2004: Tools and Algorithms for the Construction and Analysis of Systems, LNCS 2988*, pages 16–30. Springer, 2004.
5. K.L. McMillan. Applications of Craig interpolants in model checking. In *TACAS'2005: Tools and Algorithms for the Construction and Analysis of Systems, LNCS 3440*, pages 1–12. Springer, 2005.
6. V. Sofronie-Stokkermans. On combinations of local theory extensions. Submitted.
7. V. Sofronie-Stokkermans. Automated theorem proving by resolution in non-classical logics. In *4th Int. Conf. Journees de l'Informatique Messine: Knowledge Discovery and Discrete Mathematics (JIM-03)*, pages 151–167, 2003.
8. V. Sofronie-Stokkermans. Hierarchic reasoning in local theory extensions. In R. Nieuwenhuis, editor, *20th International Conference on Automated Deduction (CADE-20), LNAI 3632*, pages 219–234. Springer, 2005.
9. A. Wroński. On a form of equational interpolation property. In *Foundations of logic and linguistics (Salzburg, 1983)*, pages 23–29, New York, 1985. Plenum.
10. G. Yorsh and M. Musuvathi. A combination method for generating interpolants. In R. Nieuwenhuis, editor, *20th International Conference on Automated Deduction (CADE-20), LNAI 3632*, pages 353–368. Springer, 2005.