Modular Proof Systems for Partial Functions with Weak Equality

Harald Ganzinger, Viorica Sofronie-Stokkermans, and Uwe Waldmann

Max-Planck-Institut für Informatik, Stuhlsatzenhausweg 85,
66123 Saarbrücken, Germany, {hg,sofronie,uwe}@mpi-sb.mpg.de

Abstract. The paper presents a modular superposition calculus for the combination of first-order theories involving both total and partial functions. Modularity means that inferences are pure, only involving clauses over the alphabet of either one, but not both, of the theories. The calculus is shown to be complete provided that functions that are not in the intersection of the component signatures are declared as partial. This result also means that if the unsatisfiability of a goal modulo the combined theory does not depend on the totality of the functions in the extensions, the inconsistency will be effectively found. Moreover, we consider a constraint superposition calculus for the case of hierarchical theories and show that it has a related modularity property. Finally we identify cases where the partial models can always be made total so that modular superposition is also complete with respect to the standard (total function) semantics of the theories.

1 Introduction

This paper aims at providing new modularity results for refutational theorem proving in first-order logic with equality. In Nelson/Oppen-style combinations of two first-order theories $\mathcal{T}_1$ and $\mathcal{T}_2$ over signatures Σ_1 and Σ_2, inferences are *pure* in that all premises of an inference are clauses over only one of the signatures Σ_i where i depends on the inference. Therefore, no mixed formulas are ever generated when refuting goals represented by sets of pure formulas. What needs to be passed between the two theory modules are only universal formulas[1] over the intersection $\Sigma_1 \cap \Sigma_2$ of the two signatures. For stably infinite theories where, in addition, $\Sigma_1 \cap \Sigma_2$ consists of constants only, pure inference systems exist. This is one of the main consequences of Nelson's and Oppen's results [12] (also see, e.g., Tinelli and Harandi [14] for additional clarification). Ghilardi [8] has recently extended these completeness results for modular inference systems to a more general case of "compatibility" between the component theories $\mathcal{T}_i$. Future work might aim at liberating these compatibility requirements even further.

In this paper we take a different point of departure. We will consider arbitrary theory modules and investigate what one loses in terms of completeness when

[1] For Nelson/Oppen-style combination of theories, one even restricts the information exchange between theories to ground clauses over the intersection signature.

superposition inferences are restricted to be pure. Superposition is refutationally complete for equational first-order logic, and by choosing term orderings appropriately (terms over $\Sigma_1 \cap \Sigma_2$ should be minimal in the term ordering), many, but not all, cases of impure inferences can be avoided. Impure inferences arise when one of the extensions $\Sigma_1 \setminus \Sigma_2$ or $\Sigma_2 \setminus \Sigma_1$ has additional non-constant function symbols. It is known that in such cases interpolants of implications of the form $\phi_1 \supset \phi_2$, with ϕ_i a Σ_i-formula, normally contain existential quantification. That means, that refutationally complete clausal theorem provers where existential quantifiers are skolemized, need to pass clauses from $\mathcal{T}_1$ to $\mathcal{T}_2$ [from $\mathcal{T}_2$ to $\mathcal{T}_1$] containing function symbols not in Σ_2 [Σ_1]. In other words, inference systems are either incomplete or impure necessarily. One of the main results of the paper is that if the extensions only introduce additional relations and partial functions,[2] a particular calculus of superposition for partial functions to be developed in this paper becomes a complete and modular proof system where inferences are pure. This result can be applied to problems where partial functions arise naturally. Alternatively we may think of this result as indicating what we lose if superposition is restricted to pure inferences. If a proof cannot be found in the pure system, a partial algebra model exists for the goal to be refuted. Conversely, if the inconsistency of a goal does not depend on the totality of the functions in the extensions, we will be able to find the inconsistency with the modular partial superposition calculus. There are interesting cases of problem classes where partial models can always be totalized and where the modular system is therefore in fact complete (cf. Section 5).

In Section 2 we will describe the logic of partial functions we are working with. The logic is that of weak equality in the sense of Evans [6]. This logic allows one to specify undefinedness, but not definedness, of a function. (However we may specify a kind of relative definedness as explained below.) Then, in Section 3, we state and prove sound and complete a superposition calculus for clauses over signatures where functions can be declared as either total or partial. The calculus might be of independent interest for problem domains where partial functions arise in a natural manner. (That aspect, however, will not be explored any further in this paper as we are mainly interested in modularity.) We show that the calculus only admits pure inferences in cases of theory combinations where all functions that are not in the intersection of the signatures are declared as partial. In Section 4 we consider a variant of the calculus, called constraint superposition, suitable for hierarchical extensions $\mathcal{T}_1$ of a base theory $\mathcal{T}_0$. It differs from the previous calculus in that unification is replaced by generating equality constraints over the base theory. This system is modular in that no inferences involving base clauses (over Σ_0) need to be made. Rather, we may integrate any refutationally complete prover for $\mathcal{T}_0$ accepting the base clauses generated from non-base inferences and returning falsum whenever the accumulated set of base

[2] A non-equational literal $p(t_1, \ldots, t_n)$ or $\neg\, p(t_1, \ldots, t_n)$, where p is a relation symbol, can be encoded as an equational literal $f_p(t_1, \ldots, t_n) \approx \mathit{true}$ or $\neg\, f_p(t_1, \ldots, t_n) \approx \mathit{true}$, where f_p is a partial function and true a total constant. Thus we will in the sequel not mention relations anymore.

clauses is inconsistent with $\mathcal{T}_0$. In Section 5 we consider both shallow and local extensions of base theories, showing that for those classes of extensions constraint superposition is complete also with respect to the total algebra semantics of theories and goals. Finally Section 6 discusses related work.

2 Partial Functions with Weak Equality

A *signature* Σ is the disjoint union of a set Σ_T of total function symbols and a set Σ_P of partial function symbols. Each function symbol comes with a fixed arity.[3] Terms are built over Σ and a set V of variables.

A *Σ-algebra* A consists of a non-empty set U_A, a total function $f_A : U_A^n \to U_A$ for every $f/n \in \Sigma_\mathrm{T}$ and a partial function $g_A : U_A^n \to U_A$ for every $g/n \in \Sigma_\mathrm{P}$.

An *assignment* β into A is a mapping from the set of variables V to U_A. Given an algebra A and an assignment β into A, the value $(A,\beta)(t)$ of a Σ-term t is either a member of the universe U_A or one of the two special values $\bot_\mathrm{u}$ ("undefined") or $\bot_\mathrm{i}$ ("irrelevant"). It is defined as follows:

$(A,\beta)(x) = \beta(x)$
  if x is a variable.

$(A,\beta)(f(t_1,\dots,t_n)) = f_A(a_1,\dots,a_n)$
  if $(A,\beta)(t_i) = a_i \in U_A$ for all $i \in \{1,\dots,n\}$ and $f_A(a_1,\dots,a_n)$ is defined.

$(A,\beta)(f(t_1,\dots,t_n)) = \bot_\mathrm{u}$
  if $(A,\beta)(t_i) = a_i \in U_A$ for all $i \in \{1,\dots,n\}$ and $f_A(a_1,\dots,a_n)$ is undefined.

$(A,\beta)(f(t_1,\dots,t_n)) = \bot_\mathrm{i}$
  if $(A,\beta)(t_i) \in \{\bot_\mathrm{u},\bot_\mathrm{i}\}$ for some $i \in \{1,\dots,n\}$.

In other words, a term is irrelevant if one of its proper subterms is undefined.

To evaluate the truth of an equation, we use a three-valued logic with the values 1 (true), $\frac{1}{2}$ (undefined), and 0 (false). The truth values 1 and $\frac{1}{2}$ are called positive.

Given an algebra A and an assignment β into A, the truth value of a formula F w.r.t. A and β is denoted by $(A,\beta)(F)$. If F is an equation $s \approx t$, then $(A,\beta)(F) = 1$ if $(A,\beta)(s) = (A,\beta)(t) \in U_A$; $(A,\beta)(F) = \frac{1}{2}$ if $(A,\beta)(s) = (A,\beta)(t) = \bot_\mathrm{u}$ or $(A,\beta)(s) = \bot_\mathrm{i}$ or $(A,\beta)(t) = \bot_\mathrm{i}$; and otherwise $(A,\beta)(F) = 0$.

For complex formulas, we have

$$
\begin{aligned}
&(A,\beta)(F \wedge G) = \min\{(A,\beta)(F),(A,\beta)(G)\},\\
&(A,\beta)(F \vee G) = \max\{(A,\beta)(F),(A,\beta)(G)\},\\
&(A,\beta)(\neg F) = 1 - (A,\beta)(F),\\
&(A,\beta)(\forall x.F) = \min\{\, (A,\beta[x \mapsto a])(F) \mid a \in U_A \,\},\\
&(A,\beta)(\exists x.F) = \max\{\, (A,\beta[x \mapsto a])(F) \mid a \in U_A \,\}.
\end{aligned}
$$

[3] For simplicity, we restrict to the one-sorted case and to equality as the only predicate symbol. The extensions to a many-sorted framework where every function symbol has a unique declaration $f : s_1 \dots s_n \to s_0$ and to additional predicate symbols are obvious.

We use $s \not\approx t$ as a shorthand for $\neg\, s \approx t$; in inference rules, the symbol $\dot{\approx}$ denotes either $\approx$ or $\not\approx$.

An algebra A is a *model* of a formula F, if $(A, \beta)(F) \geq \frac{1}{2}$ for every β, or in other words, if F is positive (i.e., true or undefined) w.r.t. A and β; it is a model of a set N of formulas, if it is a model of every formula in N. If A is a model of F, we say that F holds in A. A formula F follows from a set N of formulas (denoted by $N \models F$), if every model of N is a model of F. Note that an algebra A is a model of a ground equation $s \approx t$ if both s and t are defined and equal in A, or if both are undefined, or if at least one of them is irrelevant; A is a model of $s \not\approx t$ unless both s and t are defined and different in A. It is easy to check that every ground clause C holds in an algebra A as soon as one term occurring in C is irrelevant in A. Intuitively, the ground instances of a clause that contain irrelevant terms are those instances which we choose to ignore.

Example 1. Let $\Sigma_{\mathrm{T}} = \{\mathsf{nil}/0, \mathsf{cons}/2\}$, $\Sigma_{\mathrm{P}} = \{\mathsf{car}/1, \mathsf{cdr}/1\}$, and let A be the algebra of finite lists with the usual interpretation of these symbols.

Then A is a model of $\forall x.\mathsf{cons}(\mathsf{car}(x), \mathsf{cdr}(x)) \approx x$: Suppose that x is mapped to some $a \in U_A$. Then either both $\mathsf{car}_A(a)$ and $\mathsf{cdr}_A(a)$ are undefined, hence the value of $\mathsf{cons}(\mathsf{car}(x), \mathsf{cdr}(x))$ is irrelevant, and the equation has the truth value $\frac{1}{2}$. Or $\mathsf{car}_A(a)$ and $\mathsf{cdr}_A(a)$ are defined; in this case $\mathsf{cons}_A(\mathsf{car}_A(a), \mathsf{cdr}_A(a)) = a$, so the equation has the truth value 1. The truth value of the universally quantified formula is $\min\{\frac{1}{2}, 1\} = \frac{1}{2}$, therefore A is a model of the formula.

Since $\mathsf{car}_A(\mathsf{nil}_A)$ and $\mathsf{cdr}_A(\mathsf{nil}_A)$ are undefined, A is a model of both the formula $\mathsf{car}(\mathsf{nil}) \approx \mathsf{cdr}(\mathsf{nil})$ and its negation $\mathsf{car}(\mathsf{nil}) \not\approx \mathsf{cdr}(\mathsf{nil})$. It is not a model of $\mathsf{car}(\mathsf{nil}) \approx \mathsf{nil}$ (the left-hand side is undefined, the right-hand side is defined), it is, however, a model of $\mathsf{car}(\mathsf{car}(\mathsf{nil})) \approx \mathsf{nil}$ (the left-hand side is irrelevant).

Note that explicit [un-]definedness predicates are not present in this logic. To express that a term t is not defined, one can simply state that $t \not\approx t$. Expressing that t (not containing partial function symbols below the top) is defined is only possible, if Σ_T contains appropriate total function symbols or can be extended by new symbols. For example, for an algebra B to be a model of $\forall x, y.\mathsf{car}(\mathsf{cons}(x, y)) \approx x$, car_B has to be defined for every b in the codomain of cons_B. Equations of this form implicitly express definedness requirements for partial functions.

From now on, we will consider only the clausal fragment of this logic. As usual, all variables in a clause are implicitly universally quantified. The theorem proving calculus described below will check whether a set N of clauses is inconsistent, that is, whether $N \models \bot$, where $\bot$ is the empty clause. The entailment problem "does a clause F follow from N" can be reduced to this refutation problem, but the reduction is a bit more complicated than in usual two-valued logic. The following example demonstrates the principal ideas of the reduction:

Example 2. Suppose that $\Sigma_{\mathrm{T}} \supseteq \{a/0, b/0\}$ and $\Sigma_{\mathrm{P}} \supseteq \{f/1, g/1\}$. We want to check whether $N \models f(a) \approx g(b)$ for some set N of clauses. One might think that this is equivalent to $N \cup \{f(a) \not\approx g(b)\} \models \bot$, but this is not true: If $N = \{f(a) \approx g(b)\}$, then $N \models f(a) \approx g(b)$, but still the set $N \cup \{f(a) \not\approx g(b)\}$ has a model, namely one in which $f(a)$ and $g(b)$ are undefined. The statement $N \models f(a) \approx g(b)$ holds if in each model of N either $f(a)$ and $g(b)$ are defined and equal, or both are undefined. Conversely, it does not hold, if there is a model of N in which $f(a)$ is defined and $g(b)$ is undefined or defined and different from $f(a)$, *or vice versa*. To translate the entailment problem into a *set of* refutation problems, we need therefore a new total function symbol $c/0$: $N \models f(a) \approx g(b)$ holds if and only if both $N \cup \{f(a) \approx c,\, g(b) \not\approx c\} \models \bot$ and $N \cup \{f(a) \not\approx c,\, g(b) \approx c\} \models \bot$.

3 Superposition for Partial Functions

The inference rules of the traditional superposition calculus (Bachmair and Ganzinger [1]) must be modified in several ways in order to be sound for our logic of partial functions. For instance, a literal $s \not\approx s$ may hold in an algebra – namely if s is undefined or irrelevant – so the equality resolution rule may be applied only if s is guaranteed to be defined. Similarly, replacement of equals by equals may be unsound: Assume that g is a partial function, $f(g(a))$ is irrelevant in some algebra A, and b is defined, then $f(g(a)) \approx b$ and $f(g(a)) \not\approx b$ hold in A, but $b \not\approx b$ does not. Consequently, a term that is replaced using some inference rule may contain a partial function symbol at the top, but not below the top (so that it is either defined or undefined, but not irrelevant).

For the same reason, substitutions that introduce partial function symbols must be ruled out: We say that a substitution is *total*, if no variable is mapped to a term containing a partial function symbol. If Q is a term/formula/inference and σ is a total substitution, then $Q\sigma$ is a total instance of Q.

Inference System 3. Let us start the presentation of the inference rules of the *partial superposition calculus* with a few general conventions.

The inference system is parameterized by a reduction ordering $\succ$ on terms that is total on ground terms and that has the property that every ground term over Σ_{T} is smaller than every ground term containing a symbol from Σ_{P} (for instance, a lexicographic path ordering where all symbols from Σ_{P} have higher precedence than symbols from Σ_{T}).[4]

To a positive literal $s \approx t$, we assign the multiset $\{s, t\}$, to a negative literal $\neg\, s \approx t$ the multiset $\{s, s, t, t\}$. The literal ordering $\succ_{\mathrm{L}}$ compares these multisets using the multiset extension of $\succ$. The clause ordering $\succ_{\mathrm{C}}$ compares clauses by comparing their multisets of literals using the multiset extension of $\succ_{\mathrm{L}}$.

A literal that is involved in an inference must be maximal in the respective clause (except for the literal $s_0 \approx s'_0$ in *merging paramodulation* and the literals $t_i \approx t'_i$ $(i > 1)$ in *partial top-superposition*). A positive literal that is involved in a *superposition*, *partial top-superposition*, or *merging paramodulation* inference must be strictly maximal in the respective clause (with the exceptions above). In inferences with two premises, the left premise is not greater than or equal to the right premise.

[4] Since we are interested in total ground instances only, this implies that a variable may be considered as smaller than every term containing a symbol from Σ_{P}.

Equality Resolution

$$\frac{C' \lor s \not\approx s'}{C'\sigma}$$

if s does not contain partial function symbols and σ is a total most general unifier of s and s'.

Superposition

$$\frac{D' \lor t \approx t' \qquad C' \lor s[u] \mathrel{\dot{\approx}} s'}{(D' \lor C' \lor s[t'] \mathrel{\dot{\approx}} s')\sigma}$$

if u is not a variable, t does not contain partial function symbols below the top, σ is a total most general unifier of t and u, $t\sigma \not\preceq t'\sigma$, $s\sigma \not\prec s'\sigma$, and, if $s \mathrel{\dot{\approx}} s'$ occurs positively or s is a Σ_{T}-term, then $s\sigma \not\preceq s'\sigma$.

Partial Top-Superposition

$$\frac{D' \lor t_1 \approx t_1' \lor \ldots \lor t_n \approx t_n' \qquad C' \lor s \approx s'}{(D' \lor C' \lor s' \approx t_1' \lor \ldots \lor s' \approx t_n')\sigma}$$

if $n \geq 2$, s contains a partial function symbol at the top and no partial function symbols below the top, each t_i' contains a partial function symbol, σ is a total most general unifier of s and all t_i, $t_i\sigma \not\preceq t_i'\sigma$, and $s\sigma \not\preceq s'\sigma$.[5]

Merging Paramodulation

$$\frac{D' \lor t \approx t' \qquad C' \lor s_0 \approx s_0' \lor s \approx s'[u]}{(D' \lor C' \lor s_0 \approx s_0' \lor s \approx s'[t'])\sigma}$$

if u is not a variable, t does not contain partial function symbols below the top, σ is a total most general simultaneous unifier of t and u and of s_0 and s, $t\sigma \not\preceq t'\sigma$, $s\sigma \not\preceq s'\sigma$, $s\sigma \not\preceq s_0'\sigma$, and $s'\sigma \not\preceq s_0'\sigma$,

Factoring

$$\frac{C' \lor s \approx s' \lor t \approx t'}{(C' \lor s \approx s')\sigma}$$

if σ is a total most general simultaneous unifier of s and t and of s' and t'.

Theorem 4. *The inference rules of the partial superposition calculus are sound, that is, whenever the premises of an inference hold in some algebra A, then the conclusion holds in A.*

Proof. Let us consider first the *equality resolution* rule. Suppose that A is a model of the clause $C = C' \lor s \not\approx s'$, where s is a Σ_{T}-term; let σ be a total unifier of s and s' and let β be an arbitrary assignment. Since σ is total, $x\sigma$ is a Σ_{T}-term and $(A, \beta)(x\sigma) \in U_A$ for every variable x. Define the assignment γ by $\gamma(x) = (A, \beta)(x\sigma)$. By assumption, $\frac{1}{2} \leq (A, \gamma)(C) = (A, \beta)(C\sigma) = (A, \beta)(C'\sigma \lor s\sigma \not\approx$

[5] *Partial top-superposition* corresponds to iterated *superposition* into the right premise, except that the intermediate conclusions may not be eliminated if they are redundant as defined below; in fact, it can be implemented that way.

$s'\sigma$). Now note that $s\sigma = s'\sigma$ is a Σ_{T}-term, hence $(A,\beta)(s\sigma) = (A,\beta)(s'\sigma) \in U_A$ and therefore $(A,\beta)(s\sigma \not\approx s'\sigma) = 0$. Consequently, $(A,\beta)(C'\sigma) \geq \frac{1}{2}$. Since β could be chosen arbitrarily, A is a model of $C'\sigma$.

For the *superposition* rule assume that A is a model of the clauses $D = D' \vee t \approx t'$ and $C = C' \vee s[u] \mathrel{\dot{\approx}} s'$, where t does not contain Σ_{P}-symbols below the top. W.l.o.g., C and D have no common variables. Let σ be a total unifier of t and u and let β be an arbitrary assignment. Since σ is total, $x\sigma$ is a Σ_{T}-term and $(A,\beta)(x\sigma) \in U_A$ for every variable x. Define the assignment γ by $\gamma(x) = (A,\beta)(x\sigma)$. By assumption, $\frac{1}{2} \leq (A,\gamma)(C) = (A,\beta)(C\sigma) = (A,\beta)(C'\sigma \vee s\sigma[u\sigma] \mathrel{\dot{\approx}} s'\sigma)$ and $\frac{1}{2} \leq (A,\gamma)(D) = (A,\beta)(D\sigma) = (A,\beta)(D'\sigma \vee t\sigma \approx t'\sigma)$. If $(A,\beta)(C'\sigma) \geq \frac{1}{2}$ or $(A,\beta)(D'\sigma) \geq \frac{1}{2}$, it is obvious that the conclusion is positive w.r.t. A and β. Otherwise $(A,\beta)(s\sigma[u\sigma] \mathrel{\dot{\approx}} s'\sigma) \geq \frac{1}{2}$ and $(A,\beta)(t\sigma \approx t'\sigma) \geq \frac{1}{2}$. Since t does not contain Σ_{P}-symbols below the top, $(A,\beta)(t\sigma) \in U_A \cup \{\bot_{\mathrm{u}}\}$. This leaves two possible reasons why $(A,\beta)(t\sigma \approx t'\sigma)$ is positive: If $(A,\beta)(t\sigma) = (A,\beta)(t'\sigma) \in U_A \cup \{\bot_{\mathrm{u}}\}$, then clearly $(A,\beta)(s\sigma[t'\sigma] \mathrel{\dot{\approx}} s'\sigma) = (A,\beta)(s\sigma[t\sigma] \mathrel{\dot{\approx}} s'\sigma) = (A,\beta)(s\sigma[u\sigma] \mathrel{\dot{\approx}} s'\sigma) \geq \frac{1}{2}$. Otherwise $(A,\beta)(t'\sigma) = \bot_{\mathrm{i}}$, then $(A,\beta)(s\sigma[t'\sigma]) = \bot_{\mathrm{i}}$, hence $(A,\beta)(s\sigma[t'\sigma] \mathrel{\dot{\approx}} s'\sigma) = \frac{1}{2}$.

The soundness of the *partial top-superposition* and *merging paramodulation* rules is proved analogously.

Finally we consider the *factoring* rule. Let A be a model of the clause $C = C' \vee s \approx s' \vee t \approx t'$; let σ be a total simultaneous unifier of s and t and of s' and t', and let β be an arbitrary assignment. Define the assignment γ by $\gamma(x) = (A,\beta)(x\sigma)$. By assumption, $\frac{1}{2} \leq (A,\gamma)(C) = (A,\beta)(C\sigma) = (A,\beta)(C'\sigma \vee s\sigma \approx s'\sigma \vee t\sigma \approx t'\sigma)$. Clearly, $(A,\beta)(s\sigma \approx s'\sigma) = (A,\beta)(t\sigma \approx t'\sigma)$, hence $(A,\beta)(C'\sigma \vee s\sigma \approx s'\sigma) = (A,\beta)(C\sigma) \geq \frac{1}{2}$. Since β could be chosen arbitrarily, A is a model of the conclusion.

As usual, the inference rules of the partial superposition calculus are accompanied by a redundancy criterion. A ground clause C is called redundant w.r.t. a set N of ground clauses, if it follows from clauses in N that are smaller than C. A ground inference is called redundant w.r.t. a set N of ground clauses, if its conclusion follows from clauses in N that are smaller than the largest premise. For general clauses and inferences, redundancy is defined by lifting: A clause or inference is redundant w.r.t. a set N of clauses, if all its total ground instances are redundant w.r.t the set of total ground instances of clauses in N.

A set N of clauses is called saturated up to redundancy, if all inferences between clauses in N are redundant w.r.t. N. We will show that the partial superposition calculus is refutationally complete, that is, that a saturated set of clauses has a model if and only if it does not contain the empty clause. The "only if" part of this proposition is of course trivial. For the "if" part, we have to construct a model of a saturated set N. This model is represented by a convergent term rewrite system or, equivalently, by an equational theory. The universe of the model consists of all ground normal forms of the rewrite system that are Σ_{T}-terms (or, equivalently, of the congruence classes of all ground Σ_{T}-terms). Given such a model, a ground term is defined, if its normal form is a Σ_{T}-term; it is undefined, if all its immediate subterms have normal forms that are Σ_{T}-

terms, but the term itself does not; it is irrelevant, if some of its subterms does not have a normal forms that is a Σ_{T}-term. The rewrite system is constructed from the set $\overline{N}$ of total ground instances of clauses in N. Starting with an empty interpretation all such instances are inspected in ascending order w.r.t. the clause ordering. If a reductive clause is false and irreducible in the interpretation constructed so far, its maximal equation is turned into a rewrite rule and added to the interpretation. If the original clause set is saturated and doesn't contain the empty clause, then the final interpretation is a model of all ground instances, and thus of the original clause set (Bachmair and Ganzinger [1]).

Let N be a set of clauses not containing $\bot$. Using induction on the clause ordering we define sets of rewrite rules E_C and R_C for all $C \in \overline{N}$ as follows:

Assume that E_D has already been defined for all $D \in \overline{N}$ with $D \prec_{\mathrm{C}} C$. Then $R_C = \bigcup_{D \prec_{\mathrm{C}} C} E_D$. The set E_C contains the rewrite rule $s \to s'$, if

(a) $C = C' \,\vee\, s \approx s'$.
(b) $s \approx s'$ is strictly maximal in C.
(c) $s \succ s'$.
(d) C is false in R_C.
(e) C' is false in $R_C \cup \{s \to s'\}$.
(f) s is irreducible w. r. t. R_C and contains no Σ_{P}-symbols below the top.
(g) the R_C-normal form of s' contains no Σ_{P}-symbols.
(h) no clause $D \in \overline{N}$ with $D \prec_{\mathrm{C}} C$ is false in $R_C \cup \{s \to s'\}$.

In this case, C is called productive. Otherwise $E_C = \emptyset$. Finally, $R_\infty = \bigcup_{D \in \overline{N}} E_D$.

Lemma 5. *Let N be a set of clauses that is saturated up to redundancy and does not contain the empty clause. Then we have for every total ground instance $C\theta \in \overline{N}$:*

(i) $E_{C\theta} = \emptyset$ if and only if $C\theta$ has positive truth value in $R_{C\theta}$.

(ii) $C\theta$ has positive truth value in R_∞ and in R_D for every $D \succ_{\mathrm{C}} C\theta$.

The proof can be found in the full version of the paper.

Theorem 6. *The partial superposition calculus is refutationally complete.*

Proof. We have to show that a saturated set N of clauses has a model if and only if does not contain the empty clause.

If N contains the empty clause, then obviously it does not have a model. Otherwise, the rewrite system R_∞ constructed above gives us a model A of N: The universe of A consists of all ground normal forms of R_∞ that are Σ_{T}-terms (or, equivalently, of the congruence classes of all ground Σ_{T}-terms). A function $f_A : U_A^n \to U_A$ maps the terms $t_1, \ldots, t_n$ to the R_∞-normal form of $f(t_1, \ldots, t_n)$ if this is a Σ_{T}-term, it is undefined otherwise.

There are alternative ways of dealing with partial functions in automated theorem proving, notably by encoding a partial function f/n as an $(n+1)$-ary relation r together with a clause $\neg\, r(x_1, \ldots, x_n, y) \,\vee\, \neg\, r(x_1, \ldots, x_n, y') \,\vee\, y \approx y'$.

One may ask whether partial superposition has any advantages over such an encoding. First, it is clear that the flattening of terms resulting from the relational encoding will generally make it more difficult to detect simplification opportunities. Second, the strengthened ordering restrictions of partial superposition reduce the number of possible inferences. The following trivial example illustrates this:

Example 7. Let $\Sigma_{\mathrm{T}} = \{a/0,\ b/0,\ c/0\}$, let $\Sigma_{\mathrm{P}} = \{f/1\}$, and suppose that N contains the clauses

$$\begin{aligned} f(a) &\approx b \\ f(a) &\approx c \\ b &\not\approx c \end{aligned}$$

where $a \succ b \succ c$. Partial superposition derives $b \approx c$ from the first two clauses, then $c \not\approx c$, and then the empty clause. This whole process is completely deterministic: no other inferences are possible. Besides, every superposition between a unit clause and a ground clause is necessarily a simplification of the second premise, so that the second premise $f(a) \approx b$ can be deleted from the set of clauses.

If we use relational encoding of partial functions, then N is turned into

$$\begin{gathered} r(a,b) \\ r(a,c) \\ \neg\, r(x,y) \ \vee\ \neg\, r(x,y') \ \vee\ y \approx y' \\ b \not\approx c \end{gathered}$$

In contrast to partial superposition, where we had exactly one way to derive $b \approx c$, there are now two different hyperresolution inferences that produce this clause, plus two further hyperresolution inferences that produce the tautologies $b \approx b$ and $c \approx c$. Moreover, we need now one further computation step to see that $b \approx c$ and $r(a,c)$ make $r(a,b)$ redundant.

We now show that the partial superposition calculus is modular for combinations of theories where all total functions are in the intersection of their signatures. Assume that we have two signatures Σ_1 and Σ_2. Call an inference pure if its premises are either all clauses over Σ_1 or they are all clauses over Σ_2. Note that a pure inference of the partial superposition calculus, in particular, derives a pure Σ_1-clause or a pure Σ_2-clause.

Theorem 8. *Suppose that Σ_1 and Σ_2 are two signatures that share the set of total function symbols and have disjoint sets of partial function symbols. Let N be a set of clauses, such that every clause in N is either a pure Σ_1-clause or a pure Σ_2-clause. Then all inferences of the partial superposition calculus with premises in N are pure.*

Proof. For the inference rules with only one premise, the result is trivial, since the clauses in N are pure. For the binary inference rules observe that either the term t or t_1 in the first premise contains a partial symbol; then this symbol must also occur in the second premise so that both premises are pure clauses over the same Σ_i. Or t is a Σ_{T}-term, then by the properties of the ordering the first premise contains only total symbols, hence is both a Σ_1- and a Σ_2-clause. Again, the inference is pure.

4 Hierarchic Extensions

The inference system of the partial superposition calculus (and its completeness proof) can be turned – with slight modifications – into a calculus for hierarchic structures. We assume the following scenario: Let Σ_0 be a (total) signature, and let $\mathcal{T}_0$ be a first-order theory over Σ_0 such that some refutationally complete prover exists that is able to check the unsatisfiability of sets of Σ_0-clauses w.r.t. $\mathcal{T}_0$. Now we consider an extension $\Sigma_1 \supseteq \Sigma_0$ of the signature, such that all symbols in $\Sigma_1 \setminus \Sigma_0$ are partial, and a set N of Σ_1-clauses. The task is to check whether N is unsatisfiable relative to $\mathcal{T}_0$, that is, whether there is no model of N whose restriction to Σ_0 is a model of $\mathcal{T}_0$, using the prover for $\mathcal{T}_0$ as a black-box.

We will modify the rules of the partial superposition calculus as follows:

- Replace unification by lazy unification: Change the inference rules in such a way that total most general unifiers are not computed, but instead the unification constraint is turned into new antecedent literals.
- Modify the term ordering in such a way that (sub-)terms that do not contain partial function symbols are treated like variables (that is, they are smaller than any term containing a partial function symbol, but incomparable with other terms not containing partial function symbols).
- Do not perform inferences that do not involve partial function symbols. (This implies that the *equality resolution* rule is dropped completely.)
- Add a new inference rule that allows to derive a contradiction from any finite set of Σ_0-clauses that is inconsistent with $\mathcal{T}_0$.

Inference System 9. The *constraint superposition calculus* uses the inference rules defined below.

Term and literal orderings are defined as before. A literal that is involved in an inference must be maximal in the respective clause (except for the literal $s_0 \approx s_0'$ in *merging paramodulation* and the literals $t_i \approx t_i'$ $(i > 1)$ in *partial top-superposition*). A positive literal that is involved in a *superposition*, *partial top-superposition*, or *merging paramodulation* inference must be strictly maximal in the respective clause (with the exceptions above). In inferences with two premises, the left premise is not greater than or equal to the right premise.

Superposition

$$\frac{D' \vee f(\vec{u}) \approx t' \qquad C' \vee s[f(\vec{v})] \mathrel{\dot{\approx}} s'}{D' \vee C' \vee \vec{u} \not\approx \vec{v} \vee s[t'] \mathrel{\dot{\approx}} s'}$$

if $f \in \Sigma_1 \setminus \Sigma_0$, $\vec{u}$ and $\vec{v}$ do not contain partial function symbols, $f(\vec{u}) \not\preceq t'$, and $s \not\prec s'$.

Partial Top-Superposition

$$\frac{D' \vee \bigvee_{1\le i\le n} f(\vec{u}_i) \approx t'_i \qquad C' \vee f(\vec{v}) \approx s'}{D' \vee C' \vee \bigvee_{1\le i\le n} \vec{u}_i \not\approx \vec{v} \vee \bigvee_{1\le i\le n} s' \approx t'_i}$$

if $n \geq 2$, $f \in \Sigma_1 \setminus \Sigma_0$, $\vec{u}_i$ and $\vec{v}$ do not contain partial function symbols, each t'_i contains a partial function symbol, $f(\vec{u}_i) \not\preceq t'_i$, and $f(\vec{v}) \not\preceq s'$.

Merging Paramodulation

$$\frac{D' \vee f(\vec{u}) \approx t' \qquad C' \vee g(\vec{q}) \approx s'_0 \vee g(\vec{r}) \approx s'[f(\vec{v})]}{D' \vee C' \vee \vec{q} \not\approx \vec{r} \vee \vec{u} \not\approx \vec{v} \vee g(\vec{q}) \approx s'_0 \vee g(\vec{r}) \approx s'[t']}$$

if $f, g \in \Sigma_1 \setminus \Sigma_0$, $\vec{u}$ and $\vec{v}$ do not contain partial function symbols, $f(\vec{u}) \not\preceq t'$, $g(\vec{r}) \not\preceq s'$, $g(\vec{r}) \not\preceq s'_0$, and $s' \not\preceq s'_0$,

Factoring

$$\frac{C' \vee s[\vec{q}] \approx s'[\vec{r}] \vee s[\vec{u}] \approx s'[\vec{v}]}{C' \vee \vec{q} \not\approx \vec{u} \vee \vec{r} \not\approx \vec{v} \vee s[\vec{u}] \approx s'[\vec{v}]}$$

if s contains some partial function symbol and $\vec{q}$, $\vec{r}$, $\vec{u}$, $\vec{v}$ do not contain partial function symbols.

Constraint Refutation

$$\frac{M}{\bot}$$

if M is a finite set of Σ_0-clauses that is inconsistent with the base theory $\mathcal{T}_0$.

Theorem 10. *Let N be a saturated set of clauses. Then N has a model whose restriction to Σ_0 is a model of the base theory $\mathcal{T}_0$ if and only if N does not contain the empty clause.*

Proof. The proof proceeds in essentially the same way as for the partial superposition calculus. There are two main modifications: We do not start the model construction with the empty interpretation but with some convergent ground rewrite system R_0 that is contained in $\succ$ and represents an arbitrary term-generated $\mathcal{T}_0$-model of all Σ_0-clauses in N. Such a model exists; otherwise, by compactness of $\mathcal{T}_0$, the *constraint refutation* rule would have been applied. Furthermore, we define $\overline{N}$ as the set of R_0-normal forms of total ground instances of clauses in N.

The constraint superposition calculus is related to a calculus presented by Bachmair, Ganzinger, and Waldmann [3], where a base theory is extended by *total* functions, but where sufficient completeness of the extension is necessary for the refutational completeness of the calculus.

5 Shallow and Local Extensions of a Base Theory

As shown in Section 4, constraint superposition is complete whenever all functions in the extension are declared as partial. In fact this result can be extended to the many-sorted case where only functions in the extension having a codomain of base sort have to be partial, whereas functions with a result of extension sort can be declared as either total or partial.

From our point of view, an important application of this result is to approximate refutational theorem proving in extensions of base theories for which refutationally complete black box theorem provers exist. If constraint superposition finds a contradiction for a set of clauses in the extended signature, the set is unsatisfiable in particular also with respect to total algebras. In that sense constraint superposition is a sound and modular approximation of refutational theorem proving for hierarchical first-order theories.

In this section we discuss cases for when this approximation is, in fact, complete. A particularly simple case is that of a shallow extension. Suppose $\mathcal{T}_0 \subseteq \mathcal{T}_1$ is a theory extension in which all functions in the extension $\Sigma_1 \setminus \Sigma_0$ having a codomain in the set S_0 of (base) sorts in Σ_0 are declared partial. A Σ_1-clause C is called *shallow* if partial function symbols occur in C only positively and only at the root of terms. The theory extension $\mathcal{T}_0 \subseteq \mathcal{T}_1$ is shallow if $\mathcal{T}_1 \setminus \mathcal{T}_0$ consists only of shallow clauses.

For an example, suppose we have the natural numbers (of sort nat) as base theory. Consider as an extension the two clauses

$$\mathsf{read}(\mathsf{write}(a, i, x), i) \approx x$$
$$i \approx j \ \vee \ \mathsf{read}(\mathsf{write}(a, i, x), j) \approx \mathsf{read}(a, j)$$

where array is a new sort, $\mathsf{write} : \mathsf{array} \times \mathsf{nat} \times \mathsf{nat} \to \mathsf{array}$ is a total and $\mathsf{read} : \mathsf{array} \times \mathsf{nat} \to \mathsf{nat}$ a partial function symbol, and a, i, j, x are variables of suitable sort. Under these assumptions the two clauses are shallow. This definition of read is tail-recursive, and tail-recursive definitions in general of a partial function will be shallow. Other kinds of recursive definitions will normally not be shallow, as exemplified by the case of append over nat-lists

$$\mathsf{append}(\mathsf{cons}(x, l), l') \approx \mathsf{cons}(x, \mathsf{append}(l, l'))$$

where $\mathsf{append} : \mathsf{list} \times \mathsf{list} \to \mathsf{list}$ is partial and $\mathsf{cons} : \mathsf{nat} \times \mathsf{list} \to \mathsf{list}$ is total.

Shallow extensions enjoy the property that any partial algebra model can be extended to total algebra model.

Theorem 11. *Suppose that $\mathcal{T}_0 \subseteq \mathcal{T}_1$ is a theory extension in which all functions in $\Sigma_1 \setminus \Sigma_0$ of a codomain in S_0 are declared partial. If all clauses in $\mathcal{T}_1 \setminus \mathcal{T}_0$ are shallow, then $\mathcal{T}_1$ has a partial model if, and only if, $\mathcal{T}_1$ has a total model.*

Proof. Suppose A is a partial Σ_1-algebra that is a model of $\mathcal{T}_1$. Pick, for each sort s, an element a_s from the carrier A_s associated with the sort s in A. Let B be the extension of A into a total algebra obtained by making f_B return a_s, wherever f_A is undefined in A, for every partial f of codomain s. It is easy to see that B is also a model of $\mathcal{T}_1$: Since all function symbols in Σ_0 are total, $B|_{\Sigma_0}$ coincides with $A|_{\Sigma_0}$ so that B is a model of $\mathcal{T}_0$. Suppose $\mathcal{T}_1 \setminus \mathcal{T}_0$ contains, say, an equation $f(\vec{s}) \approx g(\vec{t})$ with f partial and of sort s. Since the equation is shallow, neither $\vec{s}$ nor $\vec{t}$ contain any partial function symbol. Thus, for each assignment of the variables, the values $\vec{a}$ and $\vec{b}$ for $\vec{s}$ and $\vec{t}$, respectively, are defined. Therefore, in order for the equation to be satisfied in A, f_A is defined on $\vec{a}$ if, and only if, g_A is defined on $\vec{b}$. If $f_A(\vec{a})$ is defined, so is $g_A(\vec{b})$, and $f_B(\vec{a}) = f_A(\vec{a}) = g_A(\vec{b}) = g_B(\vec{b})$. If $f_A(\vec{a})$ is undefined so is $g_A(\vec{b})$, thus $f_B(\vec{a}) = a_s = g_B(\vec{b})$. For the case of general clauses also note that partial functions do not occur negatively in shallow clauses.

Note that any set of ground clauses can be turned into a set of shallow ground clauses by introducing new (total) constants for subterms that start with a partial function. That *flattening transformation* preserves [un-]satisfiability with respect to total algebra semantics. Therefore, a simple application of the preceding theorem is to the case where a base theory is extended by free function symbols and where we want to prove unsatisfiability of sets of ground clauses over this extension: flattening the clauses followed by applying constraint superposition is a sound and complete (with respect to total algebra semantics) and modular method for this problem. If $\mathcal{T}_0$ is a first-order theory with a decidable universal (clause) theory, then constraint superposition can be used to prove that the universal theory of any extension of $\mathcal{T}_0$ by free function symbols is also decidable. This provides a simple proof of a result established also in (Tinelli and Zarba [15]).

A more general, but related, case is that of local theories. Call a theory extension $\mathcal{T}_0 \subseteq \mathcal{T}_1$ *local* if, for every set G of ground Σ_1-clauses, whenever $\mathcal{T}_1 \cup G$ is unsatisfiable then also $\mathcal{T}_0 \cup \mathcal{T}_1[G] \cup G$ is unsatisfiable where $\mathcal{T}_1[G]$ is the set of those instances of clauses in $\mathcal{T}_1 \setminus \mathcal{T}_0$ in which each subterm starting with a partial function is a ground subterm appearing in G or in $\mathcal{T}_1 \setminus \mathcal{T}_0$. The set $\mathcal{T}_1[G]$ is finite and can be effectively computed from G. Flattening the terms in $\mathcal{T}_1[G] \cup G$ by introducing auxiliary total constants and then applying constraint superposition is, therefore, a sound and complete (with respect to total algebra semantics) and modular method for proving unsatisfiability of ground goals G with respect to $\mathcal{T}_1$.

For a flat local theory extension (i.e. a local extension $\mathcal{T}_0 \subseteq \mathcal{T}_1$ with the property that $\mathcal{T}_1 \setminus \mathcal{T}_0$ consists of clauses in which all subterms starting with a partial function functions are flat), constraint superposition is also complete (w.r.t. total semantics) when applied directly to $\mathcal{T}_1 \cup G_F$, with G_F the flattened form of G. Thus, we can avoid computing the ground instances $\mathcal{T}_1[G]$, which might be too expensive in many cases.

6 Related Work

In this section related work is summarized and compared with the results presented in the paper.

Validity of identities in partial algebras. There are many possibilities for defining validity of identities in partial algebras, from which we mention only a few (for further details we refer to [4]): "existential validity" ($(A, \beta) \models t \stackrel{e}{=} t'$ if and only if $(A, \beta)(t)$ and $(A, \beta)(t')$ are both defined and equal); "strong validity" ($(A, \beta) \models t \stackrel{s}{=} t'$ if and only if either both $(A, \beta)(t)$ and $(A, \beta)(t')$ are undefined or both are defined and equal); and validity in the sense of Evans [6, 5, 9], which is the form of validity we use in this paper. We have chosen Evans' definition, which we call weak equality, as it is too weak to allow one to define totality of a partial function (except by definitions involving previously defined total functions). On the other hand Evans validity is often related to properties of embeddability of partial algebras into total algebras [6, 5, 9]. This connection allows us to replace equational reasoning for total functions with reasoning about partial functions, or with relational reasoning. Evans validity was also used in (Ganzinger [7]) for establishing relationships between semantic and proof-theoretic approaches to polynomial time decidability for uniform word problems for quasi-varieties, in particular connections between embeddability and locality of equational theories. In the present paper we have extended Evans' validity of identities to a notion of validity for clauses, and have shown an embeddability result for shallow and for local extensions of base theories.

Resolution calculi for partial functions and partial congruences. An alternative way to dealing with undefinedness, which goes back to Kleene [11], is to use many-valued logic, with an additional truth value for "undefined". Kleene's logic has been used by various authors for giving logical systems for partial functions and for reasoning about partial functions in a many-valued framework. A resolution calculus for partial functions, where undefinedness is formalized using Kleene's strong three valued logic, was proposed by Kerber and Kohlhase in [10]. Refinements of resolutions such as paramodulation or superposition are not considered in [10].

Bachmair and Ganzinger [2] give a version of ordered chaining for partial equivalence and congruence axioms. This calculus is devised for *strong* or *existential validity*; consequently, reflexivity resolution is replaced with a rule which encodes partial reflexivity.

Modular theorem proving in combinations of theories. In Nelson/Oppen-style combinations of stably infinite theories $\mathcal{T}_1$ and $\mathcal{T}_2$ over signatures Σ_1 and Σ_2 which are disjoint or share only constants, inferences are always pure. Ghilardi [8] has recently extended the completeness results for modular inference systems for combinations of theories over non-disjoint signatures. Theorem 8, one of the main results of our paper, also provides a modular way of combining extensions $\mathcal{T}_1$ and $\mathcal{T}_2$ of an *arbitrary* base theory $\mathcal{T}_0$. The main difference between Ghilardi's approach and our work is that in (Ghilardi [8]) the component theories need to satisfy a rather strong compatibility condition with respect to the shared theory. On the other hand, our calculi are only complete with respect to the partial function semantics. We have shown, however, that for shallow or local extensions of base theories partial models can always be made total. Ghilardi's compatibility conditions [8] ensure, in addition, that the Craig interpolants consist of positive ground clauses whereas in the modular partial superposition calculus described in this paper clauses with variables need to be exchanged between the theory modules.

For Theorem 8 to be applicable, the theories $\mathcal{T}_1$ and $\mathcal{T}_2$ (regarded as theories with partial functions in Σ_1, Σ_2) have the same total function symbols. A similar situation was analyzed by Tinelli [13], who gives a method for cooperation of background reasoners for universal theories which have *the same function symbols*. However, we have shown that there are interesting problem classes where partial models can always be totalized, and therefore, in these cases the condition that the theories $\mathcal{T}_1$ and $\mathcal{T}_2$ have the same total function symbols can be relaxed. The superposition calculus for partial functions developed in this paper also allows to efficiently compute the (universal) Craig interpolant even in this more general case.

References

1. Leo Bachmair and Harald Ganzinger. Rewrite-based equational theorem proving with selection and simplification. *Journal of Logic and Computation*, 4(3):217–247, 1994.
2. Leo Bachmair and Harald Ganzinger. Ordered chaining calculi for first-order theories of transitive relations. *Journal of the ACM*, 45(6), 1998.
3. Leo Bachmair, Harald Ganzinger, and Uwe Waldmann. Refutational theorem proving for hierarchic first-order theories. *Applicable Algebra in Engineering, Communication and Computing*, 5(3/4):193–212, 1994.
4. Peter Burmeister. *A Model Theoretic Oriented Approach to Partial Algebras: Introduction to Theory and Application of Partial Algebras, Part I*, vol. 31 of *Mathematical Research*. Akademie-Verlag, Berlin, 1986.
5. Stanley Burris. Polynomial time uniform word problems. *Mathematical Logic Quarterly*, 41:173–182, 1995.
6. T. Evans. Embeddability and the word problem. *J. London Math. Soc.*, 28:76–80, 1953.
7. Harald Ganzinger. Relating semantic and proof-theoretic concepts for polynomial time decidability of uniform word problems. In *Sixteenth Annual IEEE Symposium on Logic in Computer Science*, 2001, pp. 81–92. IEEE Computer Society Press.
8. Silvio Ghilardi. Quantifier elimination and provers integration. *Electronic Notes in Theoretical Computer Science*, 86(1), 2003.
9. George Grätzer. *Universal algebra*. Springer Verlag, 2. edition, 1968.
10. Manfred Kerber and Michael Kohlhase. A mechanization of strong Kleene logic for partial functions. In Alan Bundy, ed., *Twelfth International Conference on Automated Deduction*, Nancy, France, 1994, LNAI 814, pp. 371–385. Springer-Verlag.
11. Stephen C. Kleene. *Introduction to Metamathematics*. D. Van Nostrand Company, Inc., Princeton, New Jersey, 1952.
12. Greg Nelson and Derek C. Oppen. Simplification by cooperating decision procedures. *ACM Transactions on Programming Languages and Systems*, 1(2):245–257, 1979.
13. Cesare Tinelli. Cooperation of background reasoners in theory reasoning by residue sharing. *Journal of Automated Reasoning*, 30(1):1–31, 2003.
14. Cesare Tinelli and Mehdi Harandi. A new correctness proof of the Nelson-Oppen combination procedure. In Franz Baader and Klaus U. Schulz, eds., *Frontiers of Combining Systems, First International Workshop*, Munich, Germany, 1996, Applied Logic Series, Vol. 3, pp. 103–119. Kluwer Academic Publishers.
15. Cesare Tinelli and Calogero Zarba. Combining non-stably infinite theories. *Electronic Notes in Theoretical Computer Science*, 86(1), 2003.