Some Remarks on Transfinite E-Semantic Trees and Superposition

Georg Moser*
Technische Universität Wien

Abstract

We prove the refutational completeness of $\mathcal{P}_{\text{mep}}$ by proof techniques employed in establishing the completeness of weak superposition [9]. By giving a counter-example we show that the same approach is impossible wrt. $\mathcal{P}_{\text{eqf}}$. Hence, this result shows a semantic differences between $\mathcal{P}_{\text{mep}}$ and $\mathcal{P}_{\text{eqf}}$. We apply the result to Automated Model Building.

1 Introduction

We study the relationship between two possible instances of the superposition calculus [1], the equality factoring fragment, denoted by $\mathcal{P}_{\text{eqf}}$, and the merging paramodulation fragment, $\mathcal{P}_{\text{mep}}$. We establish refutational completeness for $\mathcal{P}_{\text{mep}}$ by methods extending the techniques used in the completeness proof for weak superposition [9].

From this result we conclude that the model I_H generated from the initial clause set $\mathcal{C}$ in the completeness proof for superposition [1] coincides exactly with the model I_P described by the (infinite) right-most maximal path P in a transfinite E-semantic tree corresponding to $\mathcal{C}$, where $\mathcal{C}$ is a saturated and consistent set of clauses. On the other hand, concerning $\mathcal{P}_{\text{eqf}}$ we give a counter-example showing that the models I_H and I_P are different. Moreover, we argue that the (counter-)model I_P is more intuitive and is therefore better suited for the methods used in Automated Model Building.

We are interested in enlarging the known decidable fragments of equational logic. The use of standard theorem proving calculi to decide fragments of first-order logic has been rather successful. Wrt. first-order logic with equality, e.g. the decidability of the monadic class with equality [2] and the Ackermann class with equality [4] could be established. (See [7] for an overview.) In an attempt to overcome the technical difficulties connected with equational logic it is an interesting questions how different paramodulation calculi are related. Our result provides us with an interesting distinction between the merging paramodulation and the equality factoring fragment.

However, assume we have accomplished a decision procedure $\mathcal{I}$ for a certain class $\mathcal{F}$ of equational logic. Suppose, we employ $\mathcal{I}$ on a set of input clauses $\mathcal{C}$ and $\mathcal{I}$ halts with a saturated set of clauses $\mathcal{C}$' *not* containing the empty clause. Provided $\mathcal{I}(\mathcal{C})$ is complete, $\mathcal{C}$' is consistent. Hence, the set $\mathcal{C}$ represents a (possibly infinite) class of counter-models,

*Institut für Algebra und Diskrete Mathematik, E118.2, Wiedner Hauptstrasse 8–10, A-1040 Wien, `moser@logic.at`. This research has been partly sponsored by FWF Austria, project number P11624-MAT

but it is not possible to recognize one of these models directly: The information gained is somehow incomplete. The aim of Automated Model Building is to provide algorithms to overcome this incompleteness. Recently, remarkable results in this field have been accomplished; c.f. R. Caferra, N. Peltier, and N. Zabel, C. Fermüller and A. Leitsch, or T. Tammet. See [5] for a comprehensive overview on Automated Model Building. A better understanding of the distinction between $\mathcal{P}_{\mathrm{eqf}}$ and $\mathcal{P}_{\mathrm{mep}}$, respectively, provides insights in the structure of the class of models represented by a clause set saturated wrt. one of these inference operators. On the basis of this information it is easier to come-up with techniques suited to represent a specific counter-model in human-readable form.

The second motivation stems from implementation issues. Let $\mathcal{C}$ be a fragment of clause logic, s.t. all positive clauses are decomposed[1]. Suppose the inference system $\mathcal{I}$ decides $\mathcal{C}$. Extending $\mathcal{I}$ by a splitting rule directly yields a model building procedure for $\mathcal{C}$, provided all negative literals are selected by $\mathcal{I}$. Although such a procedure would use back-tracking, computational experiments provide evidence that back-tracking algorithms can—in some cases—do better than back-tracking free procedures. Recently, there have been very efficient implementation of the superposition calculus (SATURATE by H. Ganzinger/L. Bachmair and R. Nieuwenhuis/P. Nivela; SPASS by C. Weidenbach and others). Although this approach seems rather elegant, it has the disadvantage that the counter-model constructed depends directly on the inference operator employed.

2 Notions and Results

We assume familiarity with the concept of semantic trees, especially transfinite E-semantic trees, and maximal consistent trees $\mathrm{mct}(\mathcal{C})$, c.f. [8, 6, 9]. To simplify notation, we assume a fixed set of input clauses $\mathcal{C}$. We write $\mathcal{I}(\mathcal{C})$ to denote the closure of all inference $I \in \mathcal{I}$. We assume that tautologies are deleted wrt. $\mathcal{I}$ and that the redundancy criteria embodied in $\mathcal{I}$ is a semantic, contrary to a proof-theoretic one[2] . Either $\mathcal{P}_{\mathrm{eqf}}$ or $\mathcal{P}_{\mathrm{mep}}$ are employed as inference operators. Whenever $\mathcal{C}$ is closed under $\mathcal{I}$, $\mathcal{C}$ is called *saturated*. We write $\mathrm{mct}(\mathcal{C})$ to denote the maximal consistent E-semantic tree associated with $\mathcal{I}(\mathcal{C})$. Sometimes we write $\hat{T}$ for $\mathrm{mct}(\mathcal{C})$. Every node I in a transfinite E-semantic tree uniquely specifies a specific path P. By collecting all literals adjoined to the edges in P a partial E-interpretation is defined. To simplify notion we denote this partial interpretation by I; no confusion can arise from this simplification. In the rest of the abstract we assume $\succ$ to indicate a complete simplification ordering (CSO), for a precise definition please see [3].

Paramodulation calculi are best classified by regarding the restrictions posed on the paramodulation inference rule:

$$\frac{C_1 \vee s{\approx}t \qquad D_1 \vee u[s]{\approx}v}{C_1 \vee D_1 \vee u[t]{\approx}v}$$

If (S1) $s \succ t$ and (S2.a) $(s{\approx}t) \succ L$ for all literals L in C_1, then the clause $C = C_1 \vee s{\approx}t$ may be viewed as conditional rewrite rule with positive and negative conditions. Additionally one might require (S2.b) $s \succ w$ for all terms w in $(C)^-$. To optimize the effect a rule

[1] Literals L_1, L_2 are called decomposed if L_1, L_2 are variable-disjoint.

[2] By a semantic redundancy criteria we simply refer to a redundancy criteria embodying the classical simplification operators as subsumption deletion, demodulation, deletion of tautologies, etc.. L. Bachmair's spoke at the KGC'97 (held from August 25–29, in Vienna) about a more technical redundancy criteria that makes the inclusion of either merging paramodulation or equality factoring, superfluous.

should only be applied to a maximal atom: (S3) $(u[s]\approx v) \succ L$ for all literals L in D_1 or even to the maximal term thereof (S4) $u[s] \succ v$. Paramodulation calculi which fulfill all four conditions (S1)–(S4) are called *superposition* calculi. The *ordered literal strategy* [6] imposes only restrictions (S1), (S2.a), and (S3).

M. Rusinowitch established refutational completeness for a paramodulation rule satisfying (S1), (S3), and (S4) called *weak superposition*. The proof method used is an extension of the ordinary completeness proof by semantic trees to transfinite E-semantic trees. As $\succ$ can be of order type larger than ω it is necessary to use transfinite induction to establish completeness. The proof method is based on *reductio ad absurdum*. The existence of a non-void maximal consistent E-semantic tree is shown to be impossible for an unsatisfiable set of clauses $\mathcal{C}$. The contradiction is derived by a case analysis on the possible extensions of an assumed *maximal consistent path* Σ in $\hat{T}$.

It is well-known that superposition without further inference rules is incomplete when tautologies are deleted, cf. [1]. Therefore two additional rules were introduced by L. Bachmair and H. Ganzinger:

Definition 2.1 (Equality Factoring).

$$\frac{C_1 \vee s\approx t \vee u\approx v}{C_1\sigma \vee t\sigma\not\approx v\sigma \vee u\sigma\approx v\sigma}$$

where σ is a m.g.u. of s and u, and (i) $t\sigma \not\succeq s\sigma$, and (ii) $s\sigma\approx t\sigma$ is maximal in $C_1\sigma \vee s\sigma\approx t\sigma \vee u\sigma\approx v\sigma$.

Definition 2.2 (Merging Paramodulation).

$$\frac{C_1 \vee s\approx t \qquad D_1 \vee w\approx z \vee u\approx v[l]}{C_1\sigma \vee D_1\sigma \vee u\sigma\approx z\sigma \vee u\sigma\approx v[t]\sigma}$$

where σ is the composition $\tau\rho$ of the m.g.u. τ of s and l and $\rho = \mathrm{mgu}(u\tau, w\tau)$. Moreover (i)$s\sigma\approx t\sigma$ is strictly maximal in $\mathcal{C}_1\sigma \vee s\sigma\approx t\sigma$, (ii)$u\sigma\approx v[l]\sigma$ is strict maximal in $D_1\sigma \vee w\sigma\approx z\sigma \vee u\sigma\approx v[l]\sigma$, and (iii) $u\sigma \succ z\sigma$, and $v[l]\sigma \not\preceq z\sigma$, (iv) l is not a variable.

To restore completeness it is necessary to add either merging paramodulation or equality factoring to the inference system. We denote the instance of superposition calculus including equality factoring by $\mathcal{P}_{\mathrm{eqf}}$. Equality factoring is mainly an extension of factoring to equations, though in a quite clever way. This rule has less expressive power than merging paramodulation. Merging Paramodulation is best seen as an inference step that merges a set of different equations $\{s\approx t_i : 1 \leq i \leq k\}$ to one equation $s\approx t$ that expresses the meaning of this set The fragment including merging paramodulation is denoted by $\mathcal{P}_{\mathrm{mep}}$.

To deal with the restriction (S4) M. Rusinowitch introduced the notion of *quasi-failure nodes*, i.e. an extension of failure nodes. Let R be an extension node of $\hat{T}$. Equations true in R can be used to rewrite a clause $C \in \mathcal{C}$ to some clause C'. R is a quasi-failure node if $R(C') = \mathbf{F}$. Contrary to this approach, L. Bachmair and H. Ganzinger use induction on the clause ordering $\succ_C$ to generate a Herbrand interpretation I_H that serves as a model of $\mathcal{C}$, whenever $\mathcal{C}$ is consistent: Suppose $E_{C'}$ is defined for all ground instances $C' \prec_C C$. Let R_C be the set $\bigcup_{C \succ_C C'} E'_C$, and $I_H = R^*_C$ where R^* is the smallest congruence containing R. If A is the maximal atomic formula in C and C is false in I_C, then $E_C = \{A\}$. Otherwise E_C is the empty set. We say that C *generates* A, if $E_C = \{A\}$.

The structure of a clause falsified by a quasi-failure node R has the following form:

$$(*) \qquad C_1 \vee s{\approx}u_1 \vee \cdots \vee s{\approx}u_m \vee s{\approx}u_{m+1} \cdots \vee s{\approx}u_k$$

where $s{\approx}u_i {\to}_I\, s{\approx}t$, for $1 \leq i \leq m$ and $s{\approx}u_i {\to}_I\, s{\approx}v_i$, $m < i \leq k$, and s does not occur in C_1. I denotes the partial E-interpretation defined by the right-most maximal path Σ in $\hat{T}$ up-to R. We say that (*) *produces* the equation $s{\approx}t$.

Suppose I is the only extension node of the maximal path Σ in $\hat{T}$ and D be the clause falsified by I. One step in the completeness proof for weak superposition consists in the simulation of a sequence of paramodulation steps starting with D and $(*)$. In construction this sequence it is necessary to use non-maximal equations in (*) for paramodulation steps, otherwise it wouldn't be possible that the conclusion E is smaller and has the same truth value as D. Hence, there is no hope to meet restriction (S3).

In the presence of merging paramodulation we can deduce the following clause from (*):

$$(**) \qquad C_1 \vee s{\approx}t \vee s{\approx}u_{m+1} \cdots \vee s{\approx}u_k$$

Now the dropping of restriction (S3) is no longer necessary.

Theorem 2.3 $\mathcal{P}_{\text{mep}}$ *is provable complete via proof techniques that employ transfinite E-semantic trees.*

Corollary 2.4. *Suppose a saturated and consistent set of clauses $\mathcal{C}$, $\hat{T}$ is the maximal consistent transfinite E-semantic tree associated with $\mathcal{C}$. Then the Herbrand model I_H generated by the model generation process (concerning $\mathcal{C}$), is exactly representable by the I_P, the model obtained from an (infinite) right-most maximal path in $\hat{T}$.*

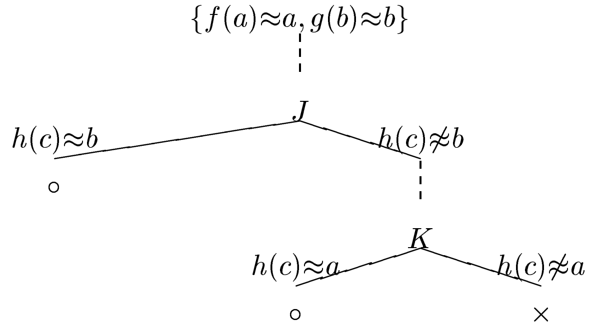
The corollary follows as the notions of *generated* and *produced* clauses coincide. In the sequel we show that the approach that was successful wrt. $\mathcal{P}_{\text{mep}}$ is doomed to fail wrt. $\mathcal{P}_{\text{eqf}}$. We give a counter-example that shows that I_P and I_H are different in this context. For the rest of this abstract we assume $\mathcal{C}$ to be a consistent and saturated clause set.

Example 2.5. We interpret $\succ$ as *LPO* with precedence $h \succ_f g \succ_f f \succ_f c \succ_f a \succ_f b$. The literals in $\mathcal{C}$ are ordered as $h(c) \succ g(a) \succ g(b) \succ f(a) \succ f(b) \succ c \succ a \succ b$. Let $\mathcal{C} = \{f(a){\approx}a, g(b){\approx}b, h(c){\approx}f(a) \vee h(c){\approx}g(b), g(b){\not\approx}f(a) \vee h(c){\approx}f(a), \ldots\}$ be a part of a saturated set of clauses[3]

The model-generation algorithm gives us the rules $R = \{f(a){\approx}a, g(b){\approx}b, h(c){\approx}g(b), \ldots\}$ and therefore the partial E-interpretation $I_H = \{f(a){\approx}a, g(b){\approx}b, h(c){\approx}b, h(c){\approx}g(b), \ldots\}$. Note that $C = f(a){\not\approx}g(b) \vee h(c){\approx}f(a)$ does not generate $h(c){\approx}f(a)$, as C is true in I_C.

The corresponding part of the right-most path P in $\text{mct}(\mathcal{C})$ can best be represented by a small figure.

[3] Although we can superpose $h(c){\approx}f(a) \vee h(c){\approx}g(b)$ on $g(b){\not\approx}f(a) \vee h(c){\approx}f(a)$, we can derive no other clause than the tautology $g(b){\not\approx}f(a) \vee g(b){\approx}f(a) \vee h(c){\approx}f(a)$



Remark, the right successor of J is *not* a quasi-failure node. Although $h(c) \approx g(b) \rightarrow_J h(c) \approx b$ and $h(c) \approx f(a) \rightarrow_J h(c) \approx a$, J does not know whether $h(c) \approx a$ is true or false.

Now suppose there exists an literal A (below $\circ$) reducible by $h(c) \approx a$. We cannot use $D = h(c) \approx g(b) \vee h(c) \approx f(a)$ to simulate this reducibility by a superposition step, as $h(c) \approx f(a)$ is not strict maximal wrt. C. Therefore the induction step of the completeness proof is blocked by this situation. In the presence of merging paramodulation it would be possible to rewrite the clause $h(c) \approx g(b) \vee h(c) \approx f(a)$ to $h(c) \approx a \vee h(c) \approx b$. Then the completeness proof could be carried out as usual.

The example shows that the models I_H and I_P are significantely different. This difference stems from the fact that we use quasi-labelings to define $\mathrm{mct}(\mathcal{C})$. On the other hand it is impossible to drop the notion of quasi-failure nodes when one wishes to satisfy condition (S4). However, an other way to overcome the obstacle would be to one refine the notion of E-interpretations defined by a path P s.t. P assigns a truth value to $h(c) \approx a$. But an attempt to refine E-interpretations must lead to an unintuitive notion of semantic trees: A clause C investigated at a node higher in $\hat{T}$, can transform a failure-node to a non-failure node, hence back-tracking on the semantic tree can be necessary. It follows that it is impossible to prove the completeness of $\mathcal{P}_{\mathrm{eqf}}$ by transitive E-semantic trees.

In the introduction we have briefly described an ad-hoc method to extend any paramodulation inference rule system to a back-tracking model building procedure for a class $\mathcal{F}$ of clause logic. Suppose, $\mathcal{F}$ meets the restriction already given. Then, our results recommend that if one is tempted to use this ad-hoc approach $\mathcal{P}_{\mathrm{eqf}}$ is not a good choice. By the above example we have seen that the clause set $\mathcal{C}$ (saturated wrt. $\mathcal{P}_{\mathrm{eqf}}$) can represent a model I that we would conceive as unnatural: I is generated by literals L that are maximal on the term level, though not maximal on the semantic level, where term equivalences are taken into account

Proposition 2.6. *Whenever one wishes to use superposition as underlying inference system for a back-tracking model building procedure, choose $\mathcal{P}_{\mathrm{mep}}$ to saturate the initial clause set $\mathcal{C}$.*

Although we have given an example that shows the distinction between the Herbrand models generated wrt. $\mathcal{P}_{\mathrm{eqf}}$ and $\mathcal{P}_{\mathrm{mep}}$, respectively, it remains to establish an elusive characterization of the semantic distinction between those fragments. We are confident to come up with a result in this direction soon.

References

[1] L. Bachmair and H. Ganzinger. Rewrite-based equational theorem proving with selection and simplification. *Journal of Logic and Computation*, 4(3):217–247, 1994. Revised version of Technical Report MPI-I-91-208, 1991.

[2] L. Bachmair, H. Ganzinger, and U. Waldmann. *Superposition with Simplification as a Decision Procedure for the Monadic Class with Equality*, pages 86–93. Number 3 in Kurt Gödel Colloquium: Computational Logic and Proof Theory. Springer Verlag, LNCS 713, 1993.

[3] Nachum Dershowitz and Jean-Pierre Jouannaud. *Rewrite Systems*, pages 245–319. Elsevier Science, 1990.

[4] C.G. Fermüller and G. Salzer. Ordered paramodulation and resolution as decision procedure. In A. Voronkov, editor, *Logic Programming and Automated Reasoning*, pages 122–133. Springer Verlag, LNCS 698, 1993.

[5] J. J. Hirschberger. Resolution-based automated model building, 1995. *Masterthesis*, TU Wien, Inst. f. Computersprachen, Abt. für Anwendungen der Formalen Logik.

[6] J. Hsiang and M. Rusinowitch. Proving refutational completeness of theorem-proving strategies: The transfinite semantic tree method. *Journal of the ACM*, 383:559–587, 1991.

[7] Georg Moser. Paramodulation calculi, 1996. *Masterthesis*, TU Wien, Inst. f. Computersprachen, Abt. für Anwendungen der Formalen Logik.

[8] G. Peterson. A technique for establishing completeness results in theorem proving with equality. *SIAM J. Comput.*, pages 82–100, 1983.

[9] M. Rusinowitch. Theorem-proving with resolution and superposition. *Journal of Symbolic Computation*, pages 21–49, 1991.