Chapter 5

First-Order Logic With Equality

In this Chapter I combine the ideas of Superposition for first-order logic without equality, Section 3.13, and Knuth-Bendix Completion, Section 4.4, to get a calculus for equational clauses. In Section 3.1 I already argued that any literal can be represented by an equation by "moving predicates to functions" and introducing a new sort Bool with specific constant true that is minimal in any considered ordering.

$$\begin{array}{lcl} P(t_1, \ldots, t_n) & \Rightarrow & f_P(t_1, \ldots, t_n) \approx \text{true} \\ \neg P(t_1, \ldots, t_n) & \Rightarrow & f_P(t_1, \ldots, t_n) \not\approx \text{true} \end{array}$$

The concentration on equational literals eases notation as I will show below. The constant true is minimal in the ordering, so the left hand side of a transformed literal is always strictly maximal. The freshly introduced functions f_P only occur at top level of a term, so a critical pair overlap between two such functions corresponds exactly to a Superposition Left (resolution) or Factoring inference of the superposition calculus for first-order logic without equality. Note that a literal true $\not\approx$ true can be simplified to $\bot$ and a literal true $\approx$ true to $\top$, respectively. So from now on I only consider equational clauses, i.e., there are no predicate symbols, $\Pi = \emptyset$.

Inference rules are to be read modulo symmetry of the equality symbol. First, I explain the ideas and motivations behind the superposition calculus with equality and its completeness proof for the ground case. At start I do not consider selection, it will be eventually added in the obvious way when considering clauses with variables.

The running example for this chapter is the theory of arrays $\mathcal{T}_{\text{Array}}$, see also Section 7.3, which consists of the following three axioms:

$$\begin{array}{l} \forall x_A, y_I, z_V . \, \text{read}(\text{store}(x, y, z), y) \approx z \\ \forall x_A, y_I, {y'}_I, z_V . (y \not\approx y' \rightarrow \text{read}(\text{store}(x, y, z), y') \approx \text{read}(x, y')) \\ \forall x_A, {x'}_A . \exists y_I . (\text{read}(x, y) \not\approx \text{read}(x', y) \vee x \approx x'). \end{array}$$

The goal is to decide for an additional set of ground clauses N over the above signature plus further constants of the three different sorts, whether $\mathcal{T}_{\text{Array}} \cup N$ is satisfiable. I will show that superposition can be turned into a decision procedure for this problem, following [**?**]. The superposition calculus including some array specific refinements, will always terminate on a clause set $\mathcal{T}_{\text{Array}} \cup N$. This results in an alternative decision procedure compared to the instantiation-based procedures used in the SMT (Satisfiability Modulo Theories) context, see Section 7.3.

5.1 Ground Superposition

The idea of the superposition calculus without equality was to restrict inferences to maximal literals, Section 3.13. Knuth-Bendix completion considers critical pairs between maximal sides of equations, Section 4.4. Superposition on equational clauses combines the two restrictions: inferences are between maximal left hand sides of maximal literals in the respective clauses. Since all considered orderings are total on ground terms, they maximality conditions can be stated positively.

The ground inference rules corresponding to Knuth-Bendix critical pair computation generalized to clauses. Superposition Left on first-order logic without equality is generalized to equational clauses an inferences below top atom positions. Then the ordering construction of Definition 3.12.1 is lifted to equational clauses. The multiset $\{s, t\}$ is assigned to a positive literal $s \approx t$, the multiset $\{s, s, t, t\}$ is assigned to a negative literal $s \not\approx t$. The *literal ordering* $\succ_L$ compares these multisets using the multiset extension of $\succ$. The *clause ordering* $\succ_C$ compares clauses by comparing their multisets of literals using the multiset extension of $\succ_L$. Eventually $\succ$ is used for all three orderings depending on the context.

Superposition Left $(N \uplus \{D \vee t \approx t', C \vee s[t] \not\approx s'\}) \Rightarrow_{\text{SUPE}} (N \cup \{D \vee t \approx t', C \vee s[t] \not\approx s'\} \cup \{D \vee C \vee s[t'] \not\approx s'\})$

where $t \approx t'$ is strictly maximal and $s \not\approx s'$ is maximal in their respective clauses, $t \succ t'$, $s \succ s'$

Superposition Right $(N \uplus \{D \vee t \approx t', C \vee s[t] \approx s'\}) \Rightarrow_{\text{SUPE}} (N \cup \{D \vee t \approx t', C \vee s[t] \approx s'\} \cup \{D \vee C \vee s[t'] \approx s'\})$

where $t \approx t'$ and $s \approx s'$ are strictly maximal in their respective clauses, $t \succ t'$, $s \succ s'$

The two rules are not yet sufficient to obtain completeness. There is no rule corresponding to Factoring and there is no way to apply reflexivity of equality, i.e., refute negative equations. The latter is solved by the below rule Equality Resolution.

Equality Resolution $(N \uplus \{C \vee s \not\approx s\}) \Rightarrow_{\text{SUPE}} (N \cup \{C \vee s \not\approx s\} \cup \{C\})$

where $s \not\approx s$ is maximal in the clause

Similar to Factoring on ground clauses, Equality Resolution is also a simplification on ground clauses, i.e., the parent clause becomes redundant with respect to the result of the derivation step. Once Equality Resolution is lifted to clauses with variables this is no longer the case, because the applied substitution may instantiate further literals in C.

It turns out that a direct adaption of the Factoring rule from superposition for first-order logic without equality is not sufficient for completeness. This becomes obvious in the context of the model construction. Basically, for the model construction the same ideas as in the completeness proof for superposition without equality apply, see Section 3.13. However, a Herbrand interpretation does not work for equality: the equality symbol $\approx$ must be interpreted by equality in the interpretation. The solution is to define a set E of ground equations and take $T(\Sigma, \emptyset)/E = T(\Sigma, \emptyset)/\approx_E$ as the universe. Then two ground terms s and t are equal in the interpretation if and only if $s \approx_E t$. If E is a terminating and confluent rewrite system R, then two ground terms s and t are equal in the interpretation, if and only if $s \downarrow_R t$.

Now the problem with the standard factoring rule is that in the completeness proof for the superposition calculus without equality, the following property holds: if $C = C' \vee A$ with a strictly maximal atom A is false in the current interpretation N_C with respect to some clause set, see Definition 3.12.5, then adding A to the current interpretation cannot make any literal in C' true. This does not hold anymore in the presence of equality. Let $b \succ c \succ d$. Assume that the current rewrite system (representing the current interpretation) contains the rule $c \to d$. Now consider the clause $b \approx c \vee b \approx d$ where $b \approx c$ is strictly maximal. A further needed inference rule to deal with clauses of this kind, is the below Equality Factoring rule, a generalization of the non-equational Factoring rule.

Equality Factoring $(N \uplus \{C \vee s \approx t' \vee s \approx t\}) \Rightarrow_{\text{SUPE}} (N \cup \{C \vee s \approx t' \vee s \approx t\} \cup \{C \vee t \not\approx t' \vee s \approx t'\})$

where $s \succ t'$, $s \succ t$ and $s \approx t$ is maximal in the clause

5.2 Superposition

The lifting from the ground case to the first-order case with variables is then identical to the case of superposition without equality: identity is replaced by unifiability, the mgu is applied to the resulting clause, and $\succ$ is replaced by $\not\preceq$. In addition, as in Knuth-Bendix completion, overlaps at or below a variable position are not considered. The consequence is that there are inferences between ground instances $D\sigma$ and $C\sigma$ of clauses D and C which are not ground instances of inferences between D and C. Such inferences have to be treated in a special way in the completeness proof and will be shown to be obsolete.

Until now I mostly described the ideas behind the superposition calculus and its completeness proof. Now, precise definitions and proofs will be given.

Inference rules are applied with respect to the commutativity of equality $\approx$. Selection of negative literals is considered as well.

Superposition Right $(N \uplus \{D \vee t \approx t', C \vee s[u] \approx s'\}) \Rightarrow_{\text{SUPE}} (N \cup \{D \vee t \approx t', C \vee s[u] \approx s'\} \cup \{(D \vee C \vee s[t'] \approx s')\sigma\})$

where σ is the mgu of t, u, the term u is not a variable, $t\sigma \not\preceq t'\sigma$, $s\sigma \not\preceq s'\sigma$, $(t \approx t')\sigma$ strictly maximal in $(D \vee t \approx t')\sigma$, nothing is selected in $D \vee t \approx t'$, and $(s \approx s')\sigma$ is strictly maximal in $(C \vee s \approx s')\sigma$ and nothing is selected in $C \vee s \approx s'$

Superposition Left $(N \uplus \{D \vee t \approx t', C \vee s[u] \not\approx s'\}) \Rightarrow_{\text{SUPE}} (N \cup \{D \vee t \approx t', C \vee s[u] \not\approx s'\} \cup \{(D \vee C \vee s[t'] \not\approx s')\sigma\})$

where σ is the mgu of t, u, the term u is not a variable, $t\sigma \not\preceq t'\sigma$, $s\sigma \not\preceq s'\sigma$, $(t \approx t')\sigma$ is strictly maximal in $(D \vee t \approx t')\sigma$, nothing is selected in $D \vee t \approx t'$, and $(s \not\approx s')\sigma$ is maximal in $(C \vee s \not\approx s')\sigma$ or selected

Equality Resolution $(N \uplus \{C \vee s \not\approx s'\}) \Rightarrow_{\text{SUPE}} (N \cup \{C \vee s \not\approx s'\} \cup \{C\sigma\})$

where σ is the mgu of s, s', $(s \not\approx s')\sigma$ maximal in $(C \vee s \not\approx s')\sigma$ or selected

Equality Factoring $(N \uplus \{C \vee s' \approx t' \vee s \approx t\}) \Rightarrow_{\text{SUPE}} (N \cup \{C \vee s' \approx t' \vee s \approx t\} \cup \{(C \vee t \not\approx t' \vee s \approx t')\sigma\})$

where σ is the mgu of s, s', $s'\sigma \not\preceq t'\sigma$, $s\sigma \not\preceq t\sigma$, $(s \approx t)\sigma$ maximal in $(C \vee s' \approx t' \vee s \approx t)\sigma$ and nothing selected

Proving soundness of the rules is not difficult, completeness, however, requires a non-trivial proof.

Theorem 5.2.1 (Superposition Soundness)**.** All inference rules of the superposition calculus are *sound*, i.e., for every rule $N \uplus \{C_1, \ldots, C_n\} \Rightarrow N \cup \{C_1, \ldots, C_n\} \cup \{D\}$ it holds that $\{C_1, \ldots, C_n\} \models D$.

The notion of redundancy does not change, i.e., a clause is redundant if it is implied by smaller clauses.

Definition 5.2.2 (Abstract Redundancy)**.** A clause C is *redundant* with respect to a clause set N if for all ground instances $C\sigma$ there are clauses $\{C_1, \ldots, C_n\} \subseteq N$ with ground instances $C_1\tau_1, \ldots, C_n\tau_n$ such that $C_i\tau_i \prec C\sigma$ for all i and $C_1\tau_1, \ldots, C_n\tau_n \models C\sigma$.

Given a set N of clauses $\text{red}(N)$ is the set of clauses redundant with respect to N.

The superposition calculus for first-order logic with equality is a generalization of the superposition calculus for first-order logic wihtout equality, Section 3.13. Hence the concrete redundancy notions from Section 3.13, namely Subsumption, Tautology Deletion, Condensation, and Subsumption Resolution

all apply to the superposition calculus for first-order logic with equality as well. In case of equations, the before mentioned criteria are tested with respect to the commutativity of equality. In addition, unit rewriting is also an instance of the abstract redundancy notion, Definition 5.2.2.

Variable Substitution $(N \uplus \{C \vee x \not\approx t\}) \Rightarrow_{\text{SUPE}} (N \cup \{C\{x \mapsto t\}\})$
provided $x \notin \text{vars}(t)$

Unit Rewriting $(N \uplus \{C \vee L, t \approx s\}) \Rightarrow_{\text{SUPE}} (N \cup \{C \vee L[s\sigma]_p, t \approx s\})$
provided $L|_p = t\sigma$ and $t\sigma \succ s\sigma$

Definition 5.2.3 (Saturation)**.** A clause set N is *saturated up to redundancy* if for every derivation $N \setminus \text{red}(N) \Rightarrow_{\text{SUPE}} N \cup \{C\}$ it holds $C \in (N \cup \text{red}(N))$.

For a set E of ground equations, $T(\Sigma, \emptyset)/E$ is an E-interpretation (or E-algebra) with universe $\{[t] \mid t \in T(\Sigma, \emptyset)\}$. Then for every *ground* equation $s \approx t$, $T(\Sigma, \emptyset)/E \models s \approx t$ holds if and only if $s \leftrightarrow_E^* t$, see Theorem 4.1.11. In particular, if E is a convergent set of rewrite rules R and $s \approx t$ is a ground equation, then $T(\Sigma, \emptyset)/R \models s \approx t$ if and only if $s \downarrow_R t$. An equation or clause is valid (or true) in R if and only if it is true in $T(\Sigma, \emptyset)/R$.

Definition 5.2.4 (Partial Model Construction)**.** Given a clause set N and an ordering $\succ$ a (partial) model $N_{\mathcal{I}}$ can be constructed inductively over all ground clause instances of N as follows:

$$
\begin{aligned}
N_C &:= \textstyle\bigcup_{D \prec C}^{D \in \text{grd}(\Sigma, N)} E_D \\
E_D &:= \begin{cases}
\{s \approx t\} & \text{if } D = D' \vee s \approx t, \\
 & (i)\ s \approx t \text{ is strictly maximal in } D \\
 & (ii)\ s \succ t \\
 & (iii)\ D \text{ is false in } N_D \\
 & (iv)\ D' \text{ is false in } N_D \cup \{s \to t\} \\
 & (v)\ s \text{ is irreducible by } N_D \\
 & (vi) \text{ no negative literal is selected in } D' \\
\emptyset & \text{otherwise}
\end{cases} \\
N_{\mathcal{I}} &:= \textstyle\bigcup_{C \in \text{grd}(\Sigma, N)} N_C
\end{aligned}
$$

where N_D, $N_{\mathcal{I}}$, E_D are also considered as rewrite systems with respect to $\succ$. If $E_D \neq \emptyset$ then D is called *productive*.

Lemma 5.2.5 (Maximal Terms in Productive Clauses)**.** If $E_C = \{s \to t\}$ and $E_D = \{l \to r\}$, then $s \succ l$ if and only if $C \succ D$.

Corollary 5.2.6 (Partial Models are Convergent Rewrite Systems)**.** The rewrite systems N_C and $N_{\mathcal{I}}$ are convergent.