3.7 Superposition

Goal:

Combine the ideas of ordered resolution (overlap maximal literals in a clause) and Knuth-Bendix completion (overlap maximal sides of equations) to get a calculus for equational clauses.

Recapitulation: Equational Clauses

Atom: either $p(s_1, \dots, s_m)$ with $p/m \in \Pi$ or $s \approx t$.

Literal: Atom or negated atom.

Clause: (possibly empty) disjunction of literals
(all variables implicitly universally quantified).

For refutational theorem proving, it is sufficient to consider sets of clauses:
every first-order formula F can be translated into a set of clauses N such that F is unsatisfiable if and only if N is unsatisfiable.

In the non-equational case, unsatisfiability can for instance be checked using the (ordered) resolution calculus.

Recapitulation: Ordered Resolution

(Ordered) resolution: inference rules:

	Ground case:	Non-ground case:
Resolution:	$\dfrac{D' \vee A \qquad C' \vee \neg A}{D' \vee C'}$	$\dfrac{D' \vee A \qquad C' \vee \neg A'}{(D' \vee C')\sigma}$ where $\sigma = \mathrm{mgu}(A, A')$.
Factoring:	$\dfrac{C' \vee A \vee A}{C' \vee A}$	$\dfrac{C' \vee A \vee A'}{(C' \vee A)\sigma}$ where $\sigma = \mathrm{mgu}(A, A')$.

Recapitulation: Ordered Resolution

Ordering restrictions:

Let $\succ$ be a well-founded and total ordering on ground atoms.

Literal ordering $\succ_L$:
compares literals by comparing lexicographically first the respective atoms using $\succ$ and then their polarities (negative $>$ positive).

Clause ordering $\succ_C$:
compares clauses by comparing their multisets of literals using the multiset extension of $\succ_L$.

Recapitulation: Ordered Resolution

Ordering restrictions (ground case):

Inference are necessary only if the following conditions are satisfied:

- The left premise of a Resolution inference is not larger than or equal to the right premise.
- The literals that are involved in the inferences ($[\neg]\, A$) are maximal in the respective clauses
  (strictly maximal for the left premise of Resolution).

Recapitulation: Ordered Resolution

Ordering restrictions (non-ground case):

Lift the ground ordering to non-ground literals:
A literal L is called [strictly] maximal in a clause C if and only if there exists a ground substitution σ such that for all other literals L' in C: $L\sigma \not\prec L'\sigma$ $[L\sigma \not\preceq L'\sigma]$.

Recapitulation: Refutational Completeness

Resolution is (even with ordering restrictions) refutationally complete:

Dynamic view of refutational completeness:

If N is unsatisfiable ($N \models \bot$) then *fair* derivations from N produce $\bot$.

Static view of refutational completeness:

If N is *saturated*, then N is unsatisfiable if and only if $\bot \in N$.

Recapitulation: Refutational Completeness

Proving refutational completeness for the ground case:

We have to show:

> If N is saturated (i. e., if sufficiently many inferences have been computed), and $\perp \notin N$, then N is satisfiable (i. e., has a model).

Recapitulation: Refutational Completeness

Model construction:

Suppose that N be saturated and $\bot \notin N$.
We inspect all clauses in N in ascending order and construct a sequence of Herbrand interpretations
(starting with the empty interpretation: all atoms are false).

If a clause C is false in the current interpretation, and has a positive and strictly maximal literal A, then extend the current interpretation such that C becomes true: add A to the current interpretation. (Then C is called *productive*.)

Otherwise, leave the current interpretation unchanged.

Recapitulation: Refutational Completeness

The sequence of interpretations has the following properties:

(1) If an atom is true in some interpretation, then it remains true in all future interpretations.

(2) If a clause is true at the time where it is inspected, then it remains true in all future interpretations.

(3) If a clause $C = C' \vee A$ is productive, then C remains true and C' remains false in all future interpretations.

Show by induction: if N is saturated and $\bot \notin N$, then every clause in N is either true at the time where it is inspected or productive.

Recapitulation: Refutational Completeness

Note:
For the induction proof, it is not necessary that the conclusion of an inference is contained in N.
It is sufficient that it is redundant w. r. t. N.

N is called *saturated up to redundancy* if the conclusion of every inference from clauses in $N \setminus \mathit{Red}(N)$ is contained in $N \cup \mathit{Red}(N)$.

Recapitulation: Refutational Completeness

Proving refutational completeness for the non-ground case:

If $C_i\theta$ is a ground instance of the clause C_i for $i \in \{0, \dots, n\}$ and

$$\frac{C_n, \dots, C_1}{C_0}$$

and

$$\frac{C_n\theta, \dots, C_1\theta}{C_0\theta}$$

are inferences, then the latter inference is called a ground instance of the former.

Recapitulation: Refutational Completeness

For a set N of clauses, let $G_\Sigma(N)$ be the set of all ground instances of clauses in N.

Construct the interpretation from the set $G_\Sigma(N)$ of all ground instances of clauses in N:

N is saturated and does not contain $\bot$

$\Rightarrow$ $G_\Sigma(N)$ is saturated and does not contain $\bot$

$\Rightarrow$ $G_\Sigma(N)$ has a Herbrand model I

$\Rightarrow$ I is a model of N.

Observation

It is possible to encode an arbitrary predicate p using a function f_p and a new constant tt:

$$\begin{array}{rcr} p(t_1, \dots, t_n) & \rightsquigarrow & f_p(t_1, \dots, t_n) \approx tt \\ \neg\, p(t_1, \dots, t_n) & \rightsquigarrow & \neg\, f_p(t_1, \dots, t_n) \approx tt \end{array}$$

In equational logic it is therefore sufficient to consider the case that $\Pi = \emptyset$, i. e., equality is the only predicate symbol.

Abbreviation: $s \not\approx t$ instead of $\neg\, s \approx t$.

The Superposition Calculus

Conventions:

From now on: $\Pi = \emptyset$ (equality is the only predicate).

Inference rules are to be read modulo symmetry of the equality symbol.

We will first explain the ideas and motivations behind the superposition calculus and its completeness proof. Precise definitions will be given later.

The Superposition Calculus

Ground inference rules:

Pos. Superposition:
$$\frac{D' \lor t \approx t' \qquad C' \lor s[t] \approx s'}{D' \lor C' \lor s[t'] \approx s'}$$

Neg. Superposition:
$$\frac{D' \lor t \approx t' \qquad C' \lor s[t] \not\approx s'}{D' \lor C' \lor s[t'] \not\approx s'}$$

Equality Resolution:
$$\frac{C' \lor s \not\approx s}{C'}$$

(Note: We will need one further inference rule.)

The Superposition Calculus

Ordering restrictions:

Some considerations:

The literal ordering must depend primarily on the larger term of an equation.

As in the resolution case, negative literals must be a bit larger than the corresponding positive literals.

Additionally, we need the following property:
If $s \succ t \succ u$, then $s \not\approx u$ must be larger than $s \approx t$.
In other words, we must compare first the larger term, then the polarity, and finally the smaller term.

The Superposition Calculus

The following construction has the required properties:

Let $\succ$ be a *reduction ordering that is total on ground terms.*

To a positive literal $s \approx t$, we assign the multiset $\{s, t\}$,
to a negative literal $s \not\approx t$ the multiset $\{s, s, t, t\}$.
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 $\succ_L$.

The Superposition Calculus

Ordering restrictions:

Ground inferences are necessary only if the following conditions are satisfied:

- In superposition inferences, the left premise is smaller than the right premise.
- The literals that are involved in the inferences are maximal in the respective clauses
  (strictly maximal for positive literals in superposition inferences).
- In these literals, the lhs is greater than or equal to the rhs (in superposition inferences: greater than the rhs).

The Superposition Calculus

Model construction:

We want to use roughly the same ideas as in the completeness proof for resolution.

But: a Herbrand interpretation does not work for equality:
The equality symbol $\approx$ must be interpreted by equality in the interpretation.

The Superposition Calculus

Solution: Define a set E of ground equations and take $\mathrm{T}_\Sigma(\emptyset)/E = \mathrm{T}_\Sigma(\emptyset)/{\approx_E}$ 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 $s \downarrow_R t$.

The Superposition Calculus

One problem:

In the completeness proof for the resolution calculus, the following property holds:

> If $C = C' \vee A$ with a strictly maximal and positive literal A is false in the current interpretation, then adding A to the current interpretation cannot make any literal of C' true.

This does not hold for superposition:

> Let $a \succ b \succ c$.
> Assume that the current rewrite system (representing the current interpretation) contains the rule $b \to c$.
> Now consider the clause $a \approx b \vee a \approx c$.

The Superposition Calculus

We need a further inference rule to deal with clauses of this kind, either the “Merging Paramodulation” rule of Bachmair and Ganzinger or the following “Equality Factoring” rule due to Nieuwenhuis:

Equality Factoring: $$\frac{C' \lor s \approx t' \lor s \approx t}{C' \lor t \not\approx t' \lor s \approx t'}$$

Note: This inference rule subsumes the usual factoring rule.

The Superposition Calculus

How do the non-ground versions of the inference rules for superposition look like?

Main idea as in the resolution calculus:

Replace identity by unifiability.

Apply the mgu to the resulting clause.

In the ordering restrictions, replace $\succ$ by $\not\preceq$.

The Superposition Calculus

However:

As in Knuth-Bendix completion, we do not want to consider overlaps at or below a variable position.

Consequence: there are inferences between ground instances $D\theta$ and $C\theta$ 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.

The Superposition Calculus

Until now, we have seen most of the ideas behind the superposition calculus and its completeness proof.

We will now start again from the beginning giving precise definitions and proofs.

The Superposition Calculus

Inference rules (part 1):

Pos. Superposition:

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

where $\sigma = \mathrm{mgu}(t, u)$ and u is not a variable.

Neg. Superposition:

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

where $\sigma = \mathrm{mgu}(t, u)$ and u is not a variable.

The Superposition Calculus

Inference rules (part 2):

Equality Resolution:
$$\frac{C' \lor s \not\approx s'}{C'\sigma}$$
where $\sigma = \mathrm{mgu}(s, s')$.

Equality Factoring:
$$\frac{C' \lor s' \approx t' \lor s \approx t}{(C' \lor t \not\approx t' \lor s \approx t')\sigma}$$
where $\sigma = \mathrm{mgu}(s, s')$.

The Superposition Calculus

Theorem 3.46:

All inference rules of the superposition calculus are correct, i. e., for every rule

$$\frac{C_n, \ldots, C_1}{C_0}$$

we have $\{C_1, \ldots, C_n\} \models C_0$.

Proof:

Exercise.

The Superposition Calculus

Orderings:

Let $\succ$ be a *reduction ordering that is total on ground terms.*

To a positive literal $s \approx t$, we assign the multiset $\{s, t\}$,
to a negative literal $s \not\approx t$ the multiset $\{s, s, t, t\}$.
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 $\succ_L$.

The Superposition Calculus

Inferences have to be computed only if the following ordering restrictions are satisfied:

- In superposition inferences, after applying the unifier to both premises, the left premise is not greater than or equal to the right one.
- The last literal in each premise is maximal in the respective premise (i. e., there exists no greater one)
  (strictly maximal for positive literals in superposition inferences, i. e., there exists no greater or equal one),
- In these literals, the lhs is not smaller than the rhs
  (in superposition inferences: neither smaller nor equal).

The Superposition Calculus

A ground clause C is called redundant w. r. t. a set of ground clauses N, if it follows from clauses in N that are smaller than C.

A clause is redundant w. r. t. a set of clauses N, if all its ground instances are redundant w. r. t. $G_\Sigma(N)$.

The set of all clauses that are redundant w. r. t. N is denoted by $\mathit{Red}(N)$.

N is called saturated up to redundancy, if the conclusion of every inference from clauses in $N \setminus \mathit{Red}(N)$ is contained in $N \cup \mathit{Red}(N)$.

Superposition: Refutational Completeness

For a set E of ground equations, $\mathrm{T}_\Sigma(\emptyset)/E$ is an E-interpretation (or E-algebra) with universe $\{\, [t] \mid t \in \mathrm{T}_\Sigma(\emptyset) \,\}$.

One can show (similar to the proof of Birkhoff's Theorem) that for every *ground* equation $s \approx t$ we have $\mathrm{T}_\Sigma(\emptyset)/E \models s \approx t$ if and only if $s \leftrightarrow_E^* t$.

In particular, if E is a convergent set of rewrite rules R and $s \approx t$ is a ground equation, then $\mathrm{T}_\Sigma(\emptyset)/R \models s \approx t$ if and only if $s \downarrow_R t$. By abuse of terminology, we say that an equation or clause is valid (or true) in R if and only if it is true in $\mathrm{T}_\Sigma(\emptyset)/R$.

Superposition: Refutational Completeness

Model construction:

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 $C \in G_\Sigma(N)$ as follows:

Assume that E_D has already been defined for all $D \in G_\Sigma(N)$ with $D \prec_C C$. Then $R_C = \bigcup_{D \prec_C C} E_D$.

Superposition: Refutational Completeness

The set E_C contains the rewrite rule $s \rightarrow t$, if

(a) $C = C' \vee s \approx t$.

(b) $s \approx t$ is strictly maximal in C.

(c) $s \succ t$.

(d) C is false in R_C.

(e) C' is false in $R_C \cup \{s \rightarrow t\}$.

(f) s is irreducible w. r. t. R_C.

In this case, C is called productive. Otherwise $E_C = \emptyset$.

Finally, $R_\infty = \bigcup_{D \in G_\Sigma(N)} E_D$.

Superposition: Refutational Completeness

Lemma 3.47:
If $E_C = \{s \rightarrow t\}$ and $E_D = \{u \rightarrow v\}$, then $s \succ u$ if and only if $C \succ_C D$.

Superposition: Refutational Completeness

Corollary 3.48:
The rewrite systems R_C and R_∞ are convergent.

Proof:
Obviously, $s \succ t$ for all rules $s \to t$ in R_C and R_∞.

Furthermore, it is easy to check that there are no critical pairs between any two rules: Assume that there are rules $u \to v$ in E_D and $s \to t$ in E_C such that u is a subterm of s. As $\succ$ is a reduction ordering that is total on ground terms, we get $u \prec s$ and therefore $D \prec_C C$ and $E_D \subseteq R_C$. But then s would be reducible by R_C, contradicting condition (f).

Superposition: Refutational Completeness

Lemma 3.49:
If $D \preceq_C C$ and $E_C = \{s \to t\}$, then $s \succ u$ for every term u occurring in a negative literal in D and $s \succeq v$ for every term v occurring in a positive literal in D.

Superposition: Refutational Completeness

Corollary 3.50:
If $D \in G_\Sigma(N)$ is true in R_D, then D is true in R_∞ and R_C for all $C \succ_C D$.

Proof:
If a positive literal of D is true in R_D, then this is obvious.

Otherwise, some negative literal $s \not\approx t$ of D must be true in R_D, hence $s \not\downarrow_{R_D} t$. As the rules in $R_\infty \setminus R_D$ have left-hand sides that are larger than s and t, they cannot be used in a rewrite proof of $s \downarrow t$, hence $s \not\downarrow_{R_C} t$ and $s \not\downarrow_{R_\infty} t$.

Superposition: Refutational Completeness

Corollary 3.51:
If $D = D' \vee u \approx v$ is productive, then D' is false and D is true in R_∞ and R_C for all $C \succ_C D$.

Proof:
Obviously, D is true in R_∞ and R_C for all $C \succ_C D$.

Since all negative literals of D' are false in R_D, it is clear that they are false in R_∞ and R_C. For the positive literals $u' \approx v'$ of D', condition (e) ensures that they are false in $R_D \cup \{u \rightarrow v\}$. Since $u' \preceq u$ and $v' \preceq u$ and all rules in $R_\infty \setminus R_D$ have left-hand sides that are larger than u, these rules cannot be used in a rewrite proof of $u' \downarrow v'$, hence $u' \not\downarrow_{R_C} v'$ and $u' \not\downarrow_{R_\infty} v'$.

Superposition: Refutational Completeness

Lemma 3.52 ("Lifting Lemma"):
Let C be a clause and let θ be a substitution such that $C\theta$ is ground. Then every equality resolution or equality factoring inference from $C\theta$ is a ground instance of an inference from C.

Proof:
Exercise.

Superposition: Refutational Completeness

Lemma 3.53 ("Lifting Lemma"):
Let $D = D' \vee u \approx v$ and $C = C' \vee [\neg]\, s \approx t$ be two clauses (without common variables) and let θ be a substitution such that $D\theta$ and $C\theta$ are ground.

If there is a superposition inference between $D\theta$ and $C\theta$ where $u\theta$ and some subterm of $s\theta$ are overlapped, and $u\theta$ does not occur in $s\theta$ at or below a variable position of s, then the inference is a ground instance of a superposition inference from D and C.

Proof:
Exercise.

Superposition: Refutational Completeness

Theorem 3.54 ("Model construction"):
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 ground clause $C\theta \in G_\Sigma(N)$:

(i) $E_{C\theta} = \emptyset$ if and only if $C\theta$ is true in $R_{C\theta}$.

(ii) If $C\theta$ is redundant w. r. t. $G_\Sigma(N)$, then it is true in $R_{C\theta}$.

(iii) $C\theta$ is true in R_∞ and in R_D for every $D \in G_\Sigma(N)$ with $D \succ_C C\theta$.

Proof:
See separate sheet.

Superposition: Refutational Completeness

A Σ-interpretation $\mathcal{A}$ is called term-generated, if for every $b \in U_{\mathcal{A}}$ there is a ground term $t \in \mathrm{T}_{\Sigma}(\emptyset)$ such that $b = \mathcal{A}(\beta)(t)$.

Superposition: Refutational Completeness

Lemma 3.55:
Let N be a set of (universally quantified) Σ-clauses and let $\mathcal{A}$ be a term-generated Σ-interpretation. Then $\mathcal{A}$ is a model of $G_\Sigma(N)$ if and only if it is a model of N.

Proof:
($\Rightarrow$): Let $\mathcal{A} \models G_\Sigma(N)$; let $(\forall \vec{x}\, C) \in N$.
Then $\mathcal{A} \models \forall \vec{x}\, C$ iff $\mathcal{A}(\gamma[x_i \mapsto a_i])(C) = 1$ for all γ and a_i.
Choose ground terms t_i such that $\mathcal{A}(\gamma)(t_i) = a_i$; define θ such that $x_i\theta = t_i$, then $\mathcal{A}(\gamma[x_i \mapsto a_i])(C) = \mathcal{A}(\gamma \circ \theta)(C) = \mathcal{A}(\gamma)(C\theta) = 1$ since $C\theta \in G_\Sigma(N)$.

($\Leftarrow$): Let $\mathcal{A}$ be a model of N; let $C \in N$ and $C\theta \in G_\Sigma(N)$.
Then $\mathcal{A}(\gamma)(C\theta) = \mathcal{A}(\gamma \circ \theta)(C) = 1$ since $\mathcal{A} \models N$.

Superposition: Refutational Completeness

Theorem 3.56 (Refutational Completeness: Static View):
Let N be a set of clauses that is saturated up to redundancy.
Then N has a model if and only if N does not contain the empty clause.

Proof:
If $\bot \in N$, then obviously N does not have a model.
If $\bot \notin N$, then the interpretation R_∞ (that is, $\mathrm{T}_\Sigma(\emptyset)/R_\infty$) is a model of all ground instances in $G_\Sigma(N)$ according to part (iii) of the model construction theorem.
As $\mathrm{T}_\Sigma(\emptyset)/R_\infty$ is term generated, it is a model of N.

Superposition: Refutational Completeness

So far, we have considered only inference rules that add new clauses to the current set of clauses
(corresponding to the *Deduce* rule of Knuth-Bendix Completion).

In other words, we have derivations of the form
$N_0 \vdash N_1 \vdash N_2 \vdash \ldots$, where each N_{i+1} is obtained from N_i by adding the consequence of some inference from clauses in N_i.

Under which circumstances are we allowed to delete (or simplify) a clause during the derivation?

Superposition: Refutational Completeness

A run of the superposition calculus is a sequence
$N_0 \vdash N_1 \vdash N_2 \vdash \dots$, such that
(i) $N_i \models N_{i+1}$, and
(ii) all clauses in $N_i \setminus N_{i+1}$ are redundant w. r. t. N_{i+1}.

In other words, during a run we may add a new clause if it follows from the old ones, and we may delete a clause, if it is redundant w. r. t. the remaining ones.

For a run, $N_\infty = \bigcup_{i\geq 0} N_i$ and $N_* = \bigcup_{i\geq 0} \bigcap_{j\geq i} N_j$.
The set N_* of all persistent clauses is called the limit of the run.

Superposition: Refutational Completeness

Lemma 3.57:

If $N \subseteq N'$, then $Red(N) \subseteq Red(N')$.

Proof:

Obvious.

Superposition: Refutational Completeness

Lemma 3.58:
If $N' \subseteq \mathit{Red}(N)$, then $N \setminus N' \models N$ and $\mathit{Red}(N) \subseteq \mathit{Red}(N \setminus N')$.

Proof:
Follows from the compactness of first-order logic and the well-foundedness of the multiset extension of the clause ordering.

Superposition: Refutational Completeness

Lemma 3.59:

Let $N_0 \vdash N_1 \vdash N_2 \vdash \ldots$ be a run.

Then $\mathit{Red}(N_i) \subseteq \mathit{Red}(N_\infty)$ and $\mathit{Red}(N_i) \subseteq \mathit{Red}(N_*)$ for every i.

Proof:

Exercise.

Superposition: Refutational Completeness

Corollary 3.60:
$N_i \subseteq N_* \cup \mathit{Red}(N_*)$ for every i.

Proof:
If $C \in N_i \setminus N_*$, then there is a $k \geq i$ such that $C \in N_k \setminus N_{k+1}$, so C must be redundant w. r. t. N_{k+1}. Consequently, C is redundant w. r. t. N_*.

Superposition: Refutational Completeness

A run is called fair, if the conclusion of every inference from clauses in $N_* \setminus \mathit{Red}(N_*)$ is contained in some $N_i \cup \mathit{Red}(N_i)$.

Lemma 3.61:
If a run is fair, then its limit is saturated up to redundancy.

Proof:
If the run is fair, then the conclusion of every inference from non-redundant clauses in N_* is contained in some $N_i \cup \mathit{Red}(N_i)$, and therefore contained in $N_* \cup \mathit{Red}(N_*)$. Hence N_* is saturated up to redundancy.

Superposition: Refutational Completeness

Theorem 3.62 (Refutational Completeness: Dynamic View):
Let $N_0 \vdash N_1 \vdash N_2 \vdash \ldots$ be a fair run, let N_* be its limit.
Then N_0 has a model if and only if $\bot \notin N_*$.

Proof:
($\Leftarrow$): By fairness, N_* is saturated up to redundancy.
If $\bot \notin N_*$, then it has a model.
Since every clause in N_0 is contained in N_* or redundant w. r. t. N_*, this model is also a model of N_0.

($\Rightarrow$): Obvious, since $N_0 \models N_*$.

Superposition: Extensions

Extensions and improvements:

simplification techniques,

selection functions (as for ordered resolution),

redundancy for inferences,

basic strategies,

constraint reasoning.

Theory Reasoning

Superposition vs. resolution + equality axioms:

specialized inference rules,
thus no inferences with theory axioms,

computation modulo symmetry,

stronger ordering restrictions,

no variable overlaps,

stronger redundancy criterion.

Theory Reasoning

Similar techniques can be used for other theories:

transitive relations,

dense total orderings without endpoints,

commutativity,

associativity and commutativity,

abelian monoids,

abelian groups,

divisible torsion-free abelian groups.

Theory Reasoning

Observations:

no inferences with theory axioms:
yes, usually possible.

computation modulo theory axioms:
often possible, but requires unification and orderings modulo theory.

stronger ordering restrictions, no variable overlaps:
sometimes possible, but in many cases, certain variable overlaps remain necessary.

stronger redundancy criterion:
depends on the model construction.

Theory Reasoning

Observations:

In many cases, integrating more theory axioms simplifies matters.

Inefficient unification procedures may be replaced by constraints.