The Solver Interface

The general CDCL($\mathcal{T}$) procedure has to be connected to a "Solver" for $\mathcal{T}$, a theory module that performs *at least* $\mathcal{T}$-satisfiability checks.

The solver is initialized with a list of all literals occurring in the input of the CDCL($\mathcal{T}$) procedure.

Internally, it keeps a stack I of theory literals that is initially empty. The solver performs the following operations on I:

SetTrue(L: $\mathcal{T}$-Literal):

Check whether $I \cup \{L\}$ is $\mathcal{T}$-satisfiable.

If no: return an explanation for $\overline{L}$, that is, a subset J of I such that $J \models_{\mathcal{T}} \overline{L}$.

If yes: push L on I.

Optionally: Return a list of literals that are $\mathcal{T}$-consequences of $I \cup \{L\}$ (and have not yet been detected before).

Note: Depending on $\mathcal{T}$, detecting (all) $\mathcal{T}$-consequences may be very cheap or very expensive.

Backtrack(n: $\mathbb{N}$):

Pop n literals from I.

Explanation(L: $\mathcal{T}$-Literal):

Return an explanation for L, that is, a subset J of I such that $J \models_{\mathcal{T}} L$.

We assume that L has been returned previously as a result of some SetTrue(L') operation. No literal of J may occur in I after L'.

Computing Backjump Clauses

Backjump clauses for a conflict can then be computed as in the propositional case:

Start with the conflicting clause.

Resolve with the clauses used for Unit Propagate or the explanations produced by the solver until a backjump clause (or $\bot$) is found.

2.2 Heuristic Instantiation

CDCL(T) is limited to ground (or existentially quantified) formulas. Even if we have decidability for more than the ground fragment of a theory $\mathcal{T}$, we cannot use this in CDCL(T).

Most current SMT implementations offer a limited support for universally quantified formulas by heuristic instantiation.

Goal:

> Create potentially useful ground instances of universally quantified clauses and add them to the given ground clauses.

Idea (Detlefs, Nelson, Saxe: Simplify):

> Select subset of the terms (or atoms) in $\forall \vec{x}\, C$ as "trigger" (automatically, but can be overridden manually).
>
> If there is a ground instance $C\theta$ of $\forall \vec{x}\, C$ such that $t\theta$ occurs (modulo congruence) in the current set of ground clauses for every $t \in \mathit{trigger}(C)$, add $C\theta$ to the set of ground clauses (incrementally).

Conditions for trigger terms (or atoms):

(1) Every quantified variable of the clause occurs in some trigger term (therefore more than one trigger term may be necessary).

(2) A trigger term is not a variable itself.

(3) A trigger is not explicitly forbidden by the user.

(4) There is no larger instance of the term in the formula:
(If $f(x)$ were selected as a trigger in $\forall x\, P(f(x), f(g(x)))$, a ground term $f(a)$ would produce an instance $P(f(a), f(g(a)))$, which would produce an instance $P(f(g(a)), f(g(g(a))))$, and so on.)

(5) No proper subterm satisfies (1)–(4).

Also possible (but expensive, therefore only in restricted form): Theory matching

> The ground atom $P(a)$ is not an instance of the trigger atom $P(x+1)$; it is however equivalent (in linear algebra) to $P((a-1)+1)$, which *is* an instance and may therefore produce a new ground clause.

Heuristic instantiation is obviously incomplete

> e. g., it does not find the contradiction for $f(x, a) \approx x$, $f(b, y) \approx y$, $a \not\approx b$

but it is quite useful in practice:

> modern implementations: CVC, Yices, Z3.

2.3 Local Theory Extensions

Under certain circumstances, instantiating universally quantified variables with "known" ground terms is sufficient for completeness.

Scenario:

$\Sigma_0 = (\Omega_0, \Pi_0)$: base signature;
$\mathcal{T}_0$: Σ_0-theory.

$\Sigma_1 = (\Omega_0 \cup \Omega_1, \Pi_0)$: signature extension;
K: universally quantified Σ_1-clauses;
G: ground clauses.

Assumption: clauses in G are Σ_1-flat and Σ_1-linear:

only constants as arguments of Ω_1-symbols,

if a constant occurs in two terms below an Ω_1-symbol, then the two terms are identical,

no term contains the same constant twice below an Ω_1-symbol.

Example: Monotonic functions over $\mathbb{Z}$.

$\mathcal{T}_0$: Linear integer arithmetic.

$\Omega_1 = \{f/1\}$.
$K = \{\ \forall x, y\ (\neg x \leq y \vee f(x) \leq f(y))\ \}$.

$G = \{\ f(3) \geq 6,\ f(5) \leq 9\ \}$.

Observation: If we choose interpretations for $f(3)$ and $f(5)$ that satisfy the G and monotonicity axiom, then it is always possible to define f for all remaining integers such that the monotonicity axiom is satisfied.

Example: Strictly monotonic functions over $\mathbb{Z}$.

$\mathcal{T}_0$: Linear integer arithmetic.

$\Omega_1 = \{f/1\}$.
$K = \{\ \forall x, y\ (\neg x < y \vee f(x) < f(y))\ \}$.

$G = \{\ f(3) > 6,\ f(5) < 9\ \}$.

Observation: Even though we can choose interpretations for $f(3)$ and $f(5)$ that satisfy G and the strict monotonicity axiom (map $f(3)$ to 7 and $f(5)$ to 8), we cannot define $f(4)$ such that the strict monotonicity axiom is satisfied.

To formalize the idea, we need partial algebras:

like (usual) total algebras, but $f_{\mathcal{A}}$ may be a partial function.

There are several ways to define equality in partial algebras (strong equality, Evans equality, weak equality, etc.). Here we use weak equality:

an equation $s \approx t$ holds w. r. t. $\mathcal{A}$ and β if both $\mathcal{A}(\beta)(s)$ and $\mathcal{A}(\beta)(t)$ are defined and equal or if at least one of them is undefined;

a negated equation $s \not\approx t$ holds w. r. t. $\mathcal{A}$ and β if both $\mathcal{A}(\beta)(s)$ and $\mathcal{A}(\beta)(t)$ are defined and different or if at least one of them is undefined.

If a partial algebra $\mathcal{A}$ satisfies a set of formulas N w. r. t. weak equality, it is called a weak partial model of N.

A partial algebra $\mathcal{A}$ embeds weakly into a partial algebra $\mathcal{B}$ if there is an injective total mapping $h : U_{\mathcal{A}} \to U_{\mathcal{B}}$ such that if $f_{\mathcal{A}}(a_1, \ldots, a_n)$ is defined in $\mathcal{A}$ then $f_{\mathcal{B}}(h(a_1), \ldots, h(a_n))$ is defined in $\mathcal{B}$ and equal to $h(f_{\mathcal{A}}(a_1, \ldots, a_n))$.

A theory extension $\mathcal{T}_0 \subseteq \mathcal{T}_0 \cup K$ is called *local*, if for every set G, $\mathcal{T}_0 \cup K \cup G$ is satisfiable if and only if $\mathcal{T}_0 \cup K[G] \cup G$ has a (partial) model, where $K[G]$ is the set of instances of clauses in K in which all terms starting with an Ω_1-symbol are ground terms occurring in K or G.

If every weak partial model of $\mathcal{T}_0 \cup K$ can be embedded into a a total model, then the theory extension $\mathcal{T}_0 \subseteq \mathcal{T}_0 \cup K$ is local (Sofronie-Stokkermans 2005).

Note: There are many variants of partial models and embeddings corresponding to different kinds of locality.

Examples of local theory extensions:

free functions, constructors/selectors, monotonic functions, Lipschitz functions.

2.4 Goal-driven Instantiation

Instantiation is used to refute the current model discovered by the ground solver.

Rather than a fast but loosely guided instantiation technique, we can search for the most suitable instance if it exists.

Scenario:

M: a model of the ground formula returned by the ground SMT solver.

$\mathcal{Q}$: the set of universally quantified clauses contained in the original input.

Problem:

Find a clause $\forall x\, C \in \mathcal{Q}$ and a grounding substitution σ such that $M \cup C\sigma$ is unsatisfiable, if it exists.

E-ground (Dis)unification Problem

Given

E: a set of ground equality literals,
N: a set of equality literals,

find σ such that $E \models N\sigma$.

The E-ground (dis)unification problem can be used to encode the goal-driven instantiation problem:

For M and each $\forall x\, C \in \mathcal{Q}$, try to solve the E-ground (dis)unification problem $M \models (\neg C)\sigma$.

Congruence Closure with Free Variables

CCFV (Barbosa et al, 2017) decomposes N into sets of smaller constraints by replacing terms with equivalent smaller ones until either

1. a variable assignment is possible, and the decomposition restarts afterwards,
2. a contradiction occurs, and the corresponding search branch is closed,
3. a substitution satisfying the problem is found.

CCFV is sound, complete and terminating for the E-ground (dis)unification problem.

Modern implementations: CVC4, VeriT.

Literature

Haniel Barbosa, Pascal Fontaine, Andrew Reynolds: Congruence Closure with Free Variables. Tools and Algorithms for the Construction and Analysis of Systems, TACAS 2017, LNCS 10206, pp. 214-230, Springer, 2017.

David Detlefs, Greg Nelson, James B. Saxe: Simplify: A Theorem Prover for Program Checking. Journal of the ACM, 52(3):365–473, 2005.

Yeting Ge, Leonardo de Moura: Complete instantiation for quantified formulas in Satisfiabiliby Modulo Theories. International Conference on Computer Aided Verification, CAV 2009 LNCS 5643, pp. 306–320, Springer, 2009.

Leonardo de Moura, Nikolaj Bjørner: Efficient E-Matching for SMT solvers. Automated Deduction, CADE-21, LNAI 4603, pp. 183–198, Springer, 2007.

Robert Nieuwenhuis, Albert Oliveras, Cesare Tinelli: Solving SAT and SAT Modulo Theories: From an Abstract Davis-Putnam-Logemann-Loveland Procedure to DPLL(T). Journal of the ACM, 53(6):937–977, 2006.

Viorica Sofronie-Stokkermans: Hierarchic reasoning in local theory extensions. Automated Deduction, CADE-20, LNAI 3632, pp. 219–234, Springer, 2005.

3 Superposition

First-order calculi considered so far:

Resolution: for first-order clauses without equality.

(Unfailing) Knuth-Bendix Completion: for unit equations.

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.

3.1 Recapitulation

First-order logic:

Atom: either $P(s_1, \ldots, s_m)$ with $P \in \Pi$ or $s \approx t$.

Literal: Atom or negated atom.

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

Refutational theorem proving:

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.

(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 $\sigma = \mathrm{mgu}(A, A')$.

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 $\succ_L$.

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).

Ordering restrictions (non-ground case):

Define the atom ordering $\succ$ also for non-ground atoms.

Need stability under substitutions: $A \succ B$ implies $A\sigma \succ B\sigma$.

Note: $\succ$ cannot be total on non-ground atoms.

For literals involved in inferences we have the same maximality requirements as in the ground case.

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 $\bot \in N$.

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 $\bot \notin N$, then N is satisfiable (i. e., has a model).

Constructing a candidate interpretation:

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.

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.

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 $N \setminus Red(N)$ is contained in $N \cup Red(N)$.

Proving refutational completeness for the non-ground case:

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

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

and

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

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

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.

It is possible to encode an arbitrary predicate P using a function f_P and a new constant *true*:

$$\begin{array}{rcl} P(t_1,\ldots,t_n) & \rightsquigarrow & f_P(t_1,\ldots,t_n) \approx \mathit{true} \\ \neg\, P(t_1,\ldots,t_n) & \rightsquigarrow & \neg\, f_P(t_1,\ldots,t_n) \approx \mathit{true} \end{array}$$

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 $\neg\, s \approx t$.

3.2 The Superposition Calculus – Informally

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.

Ground inference rules:

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

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

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

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

Ordering wishlist:

Like in resolution, we want to perform only inferences between (strictly) maximal literals.

Like in completion, we want to perform only inferences between (strictly) maximal sides of literals.

Like in resolution, in inferences with two premises, the left premise should not be larger than the right one.

Like in resolution and completion, the conclusion should then by smaller than the larger premise.

The ordering should be total on ground literals.

Consequences:

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 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 $\succ_L$.

Constructing a candidate interpretation:

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.

Solution: Productive clauses contribute ground rewrite rules to a TRS R.

The interpretation has the universe $\mathrm{T}_\Sigma(\emptyset)/R = \mathrm{T}_\Sigma(\emptyset)/{\approx_R}$; a ground atom $s \approx t$ holds in the interpretation, if and only if $s \approx_R t$ if and only if $s \leftrightarrow_R^* t$.

We will construct R in such a way that it is terminating and confluent. In this case, $s \approx_R t$ if and only if $s \downarrow_R t$.

One problem:

The completeness proof for the resolution calculus depends on the following property:

> 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 property does not hold for superposition:

> 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 $b \approx d \vee b \approx c$.

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:
$$\frac{C' \vee s \approx t' \vee s \approx t}{C' \vee t \not\approx t' \vee s \approx t'}$$

Note: This inference rule subsumes the usual factoring rule.

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, use $\not\preceq$ instead of $\succ$.

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.

3.3 The Superposition Calculus – Formally

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.

Inference rules:

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.

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

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

Theorem 3.1 *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. □

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 $\succ_L$.

Inferences have to be computed only if the following ordering restrictions are satisfied (after applying the unifier to the premises):

- In superposition inferences, 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 literal (strictly maximal for positive literals in superposition inferences, i. e., there exists no greater or equal literal).
- In these literals, the lhs is neither smaller nor equal than the rhs (except in equality resolution inferences).

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 $Red(N)$.

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

3.4 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 $\mathrm{T}_\Sigma(\emptyset)/R$.

Construction of candidate interpretations (Bachmair & Ganzinger 1990):

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 $R_C = \bigcup_{D \prec_C C} E_D$.

The set E_C contains the rewrite rule $s \to t$, 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 $R_C \cup \{s \to t\}$.

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

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

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

Lemma 3.2 *If $E_C = \{s \to t\}$ and $E_D = \{u \to v\}$, then $s \succ u$ if and only if $C \succ_C D$.*

Proof. ($\Rightarrow$): By condition (b), $s \approx t$ is strictly maximal in C and $u \approx v$ is strictly maximal in D, and since the literal ordering is total on ground literals, this implies that all other literals in C or in D are actually smaller than $s \approx t$ or $u \approx v$, respectively.

Moreover, $s \succ t$ and $u \succ v$ by condition (c). Therefore $s \succ u$ implies $\{s, t\} \succ_{\text{mul}} \{u, v\}$. Hence $s \approx t \succ_L u \approx v \succeq_L L$ for every literal L of D, and thus $C \succ_C D$.

($\Leftarrow$): Let $C \succ_C D$, then $E_D \subseteq R_C$. By condition (f), s must be irreducible w. r. t. R_C, so $s \neq u$.

Assume that $s \not\succ u$. By totality, this implies $s \preceq u$, and since $s \neq u$, we obtain $s \prec u$. But then $C \prec_C D$ can be shown in the same way as in the ($\Rightarrow$)-part, contradicting the assumption. □

Corollary 3.3 *The rewrite systems R_C and R_∞ are convergent (i. e., terminating and confluent).*

Proof. By condition (c), $s \succ t$ for all rules $s \to t$ in R_C and R_∞, so R_C and R_∞ are terminating.

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).

Now the absence of critical pairs implies local confluence, and termination and local confluence imply confluence. □

Lemma 3.4 *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 $s \succeq u$ for every term u occurring in a positive literal in D.*

Proof. If $s \preceq u$ for some term u occurring in a negative literal $u \not\approx v$ in D, then $\{u, u, v, v\} \succ_{\text{mul}} \{s, t\}$. So $u \not\approx v \succ_L s \approx t \succeq_L L$ for every literal L of C, and therefore $D \succ_C C$.

Similarly, if $s \prec u$ for some term u occurring in a positive literal $u \approx v$ in D, then $\{u, v\} \succ_{\text{mul}} \{s, t\}$. So $u \approx v \succ_L s \approx t \succeq_L L$ for every literal L of C, and therefore $D \succ_C C$. □

Corollary 3.5 *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 $s \not\downarrow_{R_\infty} t$. □

Corollary 3.6 *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 $R_D \cup \{u \to v\}$. 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 $u' \not\downarrow_{R_\infty} v'$. □

Lemma 3.7 ("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. □

Lemma 3.8 ("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. □

Theorem 3.9 ("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. We use induction on the clause ordering $\succ_{\mathrm{C}}$ and assume that (i)–(iii) are already satisfied for all clauses in $G_\Sigma(N)$ that are smaller than $C\theta$. Note that the "if" part of (i) is obvious from the construction and that condition (iii) follows immediately from (i) and Corollaries 3.5 and 3.6. So it remains to show (ii) and the "only if" part of (i).

Case 1: $C\theta$ is redundant w. r. t. $G_\Sigma(N)$.

If $C\theta$ is redundant w. r. t. $G_\Sigma(N)$, then it follows from clauses in $G_\Sigma(N)$ that are smaller than $C\theta$. By part (iii) of the induction hypothesis, these clauses are true in $R_{C\theta}$. Hence $C\theta$ is true in $R_{C\theta}$.

Case 2: $x\theta$ is reducible by $R_{C\theta}$.

Suppose there is a variable x occurring in C such that $x\theta$ is reducible by $R_{C\theta}$, say $x\theta \to_{R_{C\theta}} w$. Let the substitution θ' be defined by $x\theta' = w$ and $y\theta' = y\theta$ for every variable $y \neq x$. The clause $C\theta'$ is smaller than $C\theta$. By part (iii) of the induction hypothesis, it is true in $R_{C\theta}$. By congruence, every literal of $C\theta$ is true in $R_{C\theta}$ if and only if the corresponding literal of $C\theta'$ is true in $R_{C\theta}$; hence $C\theta$ is true in $R_{C\theta}$.

Case 3: $C\theta$ contains a maximal negative literal.

Suppose that $C\theta$ does not fall into Case 1 or 2 and that $C\theta = C'\theta \vee s\theta \not\approx s'\theta$, where $s\theta \not\approx s'\theta$ is maximal in $C\theta$. If $s\theta \approx s'\theta$ is false in $R_{C\theta}$, then $C\theta$ is clearly true in $R_{C\theta}$ and we are done. So assume that $s\theta \approx s'\theta$ is true in $R_{C\theta}$, that is, $s\theta \downarrow_{R_{C\theta}} s'\theta$. Without loss of generality, $s\theta \succeq s'\theta$.

Case 3.1: $s\theta = s'\theta$.

If $s\theta = s'\theta$, then there is an *equality resolution* inference

$$\frac{C'\theta \vee s\theta \not\approx s'\theta}{C'\theta} .$$

As shown in the Lifting Lemma, this is an instance of an *equality resolution* inference

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

where $C = C' \vee s \not\approx s'$ is contained in N and $\theta = \rho \circ \sigma$. (Without loss of generality, σ is idempotent, therefore $C'\theta = C'\sigma\rho = C'\sigma\sigma\rho = C'\sigma\theta$, so $C'\theta$ is a ground instance of $C'\sigma$.) Since $C\theta$ is not redundant w. r. t. $G_\Sigma(N)$, C is not redundant w. r. t. N. As N is saturated up to redundancy, the conclusion $C'\sigma$ of the inference from C is contained in $N \cup Red(N)$. Therefore, $C'\theta$ is either contained in $G_\Sigma(N)$ and smaller than $C\theta$, or it follows from clauses in $G_\Sigma(N)$ that are smaller than itself (and therefore smaller than $C\theta$). By the induction hypothesis, clauses in $G_\Sigma(N)$ that are smaller than $C\theta$ are true in $R_{C\theta}$, thus $C'\theta$ and $C\theta$ are true in $R_{C\theta}$.

Case 3.2: $s\theta \succ s'\theta$.

If $s\theta \downarrow_{R_{C\theta}} s'\theta$ and $s\theta \succ s'\theta$, then $s\theta$ must be reducible by some rule in some $E_{D\theta} \subseteq R_{C\theta}$. (Without loss of generality we assume that C and D are variable disjoint; so we can use the same substitution θ.) Let $D\theta = D'\theta \vee t\theta \approx t'\theta$ with $E_{D\theta} = \{t\theta \to t'\theta\}$. Since $D\theta$ is productive, $D'\theta$ is false in $R_{C\theta}$. Besides, by part (ii) of the induction hypothesis, $D\theta$ is not redundant w. r. t. $G_\Sigma(N)$, so D is not redundant w. r. t. N. Note that $t\theta$ cannot occur in $s\theta$ at or below a variable position of s, say $x\theta = w[t\theta]$, since otherwise $C\theta$ would be subject to Case 2 above. Consequently, the *negative superposition* inference

$$\frac{D'\theta \vee t\theta \approx t'\theta \qquad C'\theta \vee s\theta[t\theta] \not\approx s'\theta}{D'\theta \vee C'\theta \vee s\theta[t'\theta] \not\approx s'\theta}$$

is a ground instance of a *negative superposition* inference from D and C. By saturation up to redundancy, its conclusion is either contained in $G_\Sigma(N)$ and smaller than $C\theta$, or it follows from clauses in $G_\Sigma(N)$ that are smaller than itself (and therefore smaller than $C\theta$). By the induction hypothesis, these clauses are true in $R_{C\theta}$, thus $D'\theta \vee C'\theta \vee s\theta[t'\theta] \not\approx s'\theta$ is true in $R_{C\theta}$. Since $D'\theta$ and $s\theta[t'\theta] \not\approx s'\theta$ are false in $R_{C\theta}$, both $C'\theta$ and $C\theta$ must be true.

Case 4: $C\theta$ does not contain a maximal negative literal.

Suppose that $C\theta$ does not fall into Cases 1 to 3. Then $C\theta$ can be written as $C'\theta \vee s\theta \approx s'\theta$, where $s\theta \approx s'\theta$ is a maximal literal of $C\theta$. If $E_{C\theta} = \{s\theta \to s'\theta\}$ or $C'\theta$ is true in $R_{C\theta}$ or $s\theta = s'\theta$, then there is nothing to show, so assume that $E_{C\theta} = \emptyset$ and that $C'\theta$ is false in $R_{C\theta}$. Without loss of generality, $s\theta \succ s'\theta$.

Case 4.1: $s\theta \approx s'\theta$ is maximal in $C\theta$, but not strictly maximal.

If $s\theta \approx s'\theta$ is maximal in $C\theta$, but not strictly maximal, then $C\theta$ can be written as $C''\theta \vee t\theta \approx t'\theta \vee s\theta \approx s'\theta$, where $t\theta = s\theta$ and $t'\theta = s'\theta$. In this case, there is a *equality factoring* inference

$$\frac{C''\theta \vee t\theta \approx t'\theta \vee s\theta \approx s'\theta}{C''\theta \vee t'\theta \not\approx s'\theta \vee t\theta \approx t'\theta}$$

This inference is a ground instance of an inference from C. By saturation, its conclusion is true in $R_{C\theta}$. Trivially, $t'\theta = s'\theta$ implies $t'\theta \downarrow_{R_{C\theta}} s'\theta$, so $t'\theta \not\approx s'\theta$ must be false and $C\theta$ must be true in $R_{C\theta}$.

Case 4.2: $s\theta \approx s'\theta$ is strictly maximal in $C\theta$ and $s\theta$ is reducible.

Suppose that $s\theta \approx s'\theta$ is strictly maximal in $C\theta$ and $s\theta$ is reducible by some rule in $E_{D\theta} \subseteq R_{C\theta}$. Let $D\theta = D'\theta \vee t\theta \approx t'\theta$ and $E_{D\theta} = \{t\theta \to t'\theta\}$. Since $D\theta$ is productive, $D\theta$ is not redundant and $D'\theta$ is false in $R_{C\theta}$. We can now proceed in essentially the same way as in Case 3.2: If $t\theta$ occurred in $s\theta$ at or below a variable position of s, say $x\theta = w[t\theta]$, then $C\theta$ would be subject to Case 2 above. Otherwise, the *positive superposition* inference

$$\frac{D'\theta \vee t\theta \approx t'\theta \qquad C'\theta \vee s\theta[t\theta] \approx s'\theta}{D'\theta \vee C'\theta \vee s\theta[t'\theta] \approx s'\theta}$$

is a ground instance of a *positive superposition* inference from D and C. By saturation up to redundancy, its conclusion is true in $R_{C\theta}$. Since $D'\theta$ and $C'\theta$ are false in $R_{C\theta}$, $s\theta[t'\theta] \approx s'\theta$ must be true in $R_{C\theta}$. On the other hand, $t\theta \approx t'\theta$ is true in $R_{C\theta}$, so by congruence, $s\theta[t\theta] \approx s'\theta$ and $C\theta$ are true in $R_{C\theta}$.

Case 4.3: $s\theta \approx s'\theta$ is strictly maximal in $C\theta$ and $s\theta$ is irreducible.

Suppose that $s\theta \approx s'\theta$ is strictly maximal in $C\theta$ and $s\theta$ is irreducible by $R_{C\theta}$. Then there are three possibilities: $C\theta$ can be true in $R_{C\theta}$, or $C'\theta$ can be true in $R_{C\theta} \cup \{s\theta \to s'\theta\}$, or $E_{C\theta} = \{s\theta \to s'\theta\}$. In the first and the third case, there is nothing to show. Let us therefore assume that $C\theta$ is false in $R_{C\theta}$ and $C'\theta$ is true in $R_{C\theta} \cup \{s\theta \to s'\theta\}$. Then $C'\theta = C''\theta \vee t\theta \approx t'\theta$, where the literal $t\theta \approx t'\theta$ is true in $R_{C\theta} \cup \{s\theta \to s'\theta\}$ and false in $R_{C\theta}$. In other words, $t\theta \downarrow_{R_{C\theta} \cup \{s\theta \to s'\theta\}} t'\theta$, but not $t\theta \downarrow_{R_{C\theta}} t'\theta$. Consequently, there is a rewrite proof of $t\theta \to^* u \leftarrow^* t'\theta$ by $R_{C\theta} \cup \{s\theta \to s'\theta\}$ in which the rule $s\theta \to s'\theta$ is used at least once. Without loss of generality we assume that $t\theta \succeq t'\theta$. Since $s\theta \approx s'\theta \succ_{\mathrm{L}} t\theta \approx t'\theta$ and $s\theta \succ s'\theta$ we can conclude that $s\theta \succeq t\theta \succ t'\theta$. But then there is only one possibility how the rule $s\theta \to s'\theta$ can be used in the rewrite proof: We must have $s\theta = t\theta$ and the rewrite proof must have the form $t\theta \to s'\theta \to^* u \leftarrow^* t'\theta$, where the first step uses $s\theta \to s'\theta$ and all other steps use rules from $R_{C\theta}$. Consequently, $s'\theta \approx t'\theta$ is true in $R_{C\theta}$. Now observe that there is an *equality factoring* inference

$$\frac{C''\theta \vee t\theta \approx t'\theta \vee s\theta \approx s'\theta}{C''\theta \vee t'\theta \not\approx s'\theta \vee t\theta \approx t'\theta}$$

whose conclusion is true in $R_{C\theta}$ by saturation. Since the literal $t'\theta \not\approx s'\theta$ must be false in $R_{C\theta}$, the rest of the clause must be true in $R_{C\theta}$, and therefore $C\theta$ must be true in $R_{C\theta}$, contradicting our assumption. This concludes the proof of the theorem. □

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 $b = \mathcal{A}(\beta)(t)$.

Lemma 3.10 *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 $\forall \vec{x}\, C \in N$ and $C\theta \in G_\Sigma(N)$. Then $\mathcal{A} \models \forall \vec{x}\, C$ and therefore $\mathcal{A} \models C$. Consequently $\mathcal{A}(\gamma)(C\theta) = \mathcal{A}(\gamma \circ \theta)(C) = 1$. □

Theorem 3.11 (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. □