Superposition and Decision Procedures Back and Forth

Thomas Hillenbrand

Dissertation zur Erlangung des Grades
des Doktors der Naturwissenschaften
der Naturwissenschaftlich-Technischen Fakultäten
der Universität des Saarlandes

Saarbrücken
2008

Tag des Kolloquiums	24. April 2009
Dekan	Prof. Dr. Joachim Weickert
Vorsitzender des Prüfungsausschusses	Prof. Dr. Gert Smolka
Berichterstatter	Prof. Dr. Christoph Weidenbach
	Prof. Dr. Bernd Finkbeiner
	Prof. Dr. Robert Nieuwenhuis
Akademischer Mitarbeiter	Dr. Uwe Waldmann

Abstract

Two apparently different approaches to automating deduction are mentioned in the title; they are the subject of a debate on "big engines vs. little engines of proof". The contributions in this thesis advocate that these two strands of research can interplay in subtle and sometimes unexpected ways, such that mutual pervasion can lead to intriguing results: Firstly, superposition can be run on top of decision procedures. This we demonstrate for the class of Shostak theories, incorporating a little engine into a big one. As another instance of decision procedures within superposition, we show that ground confluent rewrite systems, which decide entailment problems in equational logic, can be harnessed for detecting redundancies in superposition derivations. Secondly, superposition can be employed as proof-theoretic means underneath combined decision procedures: We re-establish the correctness of the Nelson-Oppen procedure as an instance of the completeness of superposition. Thirdly, superposition can be used as a decision procedure for many interesting theories, turning a big engine into a little one. For the theory of bits and of fixed-size bitvectors, we suggest a rephrased axiomatization combined with a transformation of conjectures, based on which superposition decides the universal fragment. Furthermore, with a modification of lifting, we adapt superposition to the theory of bounded domains and give a decision procedure, which captures the Bernays-Schönfinkel class as well.

Zusammenfassung

Zwei augenscheinlich verschiedene Ansätze zum Automatisieren der Deduktion werden im Titel angeführt. Die Beiträge in dieser Dissertation zeigen auf, daß diese zwei Ansätze in subtiler und manchmal unerwarteter Weise im Wechselspiel stehen, so daß wechselseitige Durchdringung zu verblüffenden Ergebnissen führen kann: Erstens kann Superposition modulo Entscheidungsverfahren betrieben werden; wir zeigen dies für die Klasse der Shostak-Theorien. Als ein weiteres Beispiel von Entscheidungsverfahren innerhalb von Superposition zeigen wir, daß in Superpositionsableitungen mit grundkonfluenten Reduktionssysteme, die bekanntlich das gleichungslogische Wortproblem entscheiden, Redundanzen erkannt werden können. Zweitens kann man mit Superposition als beweistheoretischem Werkzeug die Kombination von Entscheidungsverfahren fundieren: Wir rekonstruieren die Korrektheit des Nelson-Oppen-Verfahrens aus der Vollständigkeit der Superposition. Drittens kann Superposition verwendet werden als Entscheidungsverfahren für viele wichtige Theorien. Für die Theorie der Bits und der Bitvektoren fester Länge stellen wir eine umformulierte Axiomatisierung zusammen mit einer Transformation von Konjekturen vor, so daß sich mit Superposition das universelle Fragment entscheiden läßt. Weiterhin passen wir Superposition mittels eines geänderten Liftings an die Theorie beschränkter Domänen an und geben ein Entscheidungsverfahren an; dieses entscheidet auch die Bernays-Schönfinkel-Klasse.

Contents

1 Introduction **1**

2 Formal Preliminaries **5**
2.1 Syntax of First-Order Logic 5
2.2 Semantics of First-Order Logic 8
2.3 Theory Reasoning Bits 11
2.4 Rewriting and Orderings 15
2.5 Calculi and Derivations 19

3 Superposition modulo a Shostak Theory **21**
3.1 Introduction 21
3.2 Preliminaries 22
3.3 Basic Components 23
3.3.1 The Canonizer and its Extension 23
3.3.2 The Black-Box Path Ordering 25
3.3.3 Canonizing and Rewriting 30
3.3.4 The Solver, and its Extension and Application 33
3.4 Superposition 36
3.4.1 Ground Case 36
3.4.2 Towards Non-ground Clauses 41
3.5 Summary 42

4 A Superposition View on Nelson-Oppen **43**
4.1 Introduction 43
4.2 Reconstructing Nelson-Oppen 43
4.2.1 Obtaining a Clausal Theory Presentation 44
4.2.2 Combining Convex Theories 46

5 Dealing with Bits and Vectors thereof **49**
5.1 Datapath Verification with SPASS 49
5.2 Axioms on Bits 57
5.3 A Transformational Approach 61

5.3.1 The Formula Transformation . . . 62
5.3.2 Soundness . . . 67
5.3.3 Completeness on Clauses . . . 69
5.3.4 Tools for the Level of Formulae . . . 69
5.3.5 Completeness on Formulae . . . 71
5.4 A Clausal Approximation . . . 75
5.4.1 Introducing the Approximation . . . 75
5.4.2 Saturating the Approximation . . . 77
5.4.3 Deciding the Universal Fragment of $\mathbb{B}$ . . . 79
5.5 Beyond Bits . . . 84
5.6 Axioms on Bitvectors . . . 85
5.7 Extending the Transformational Approach . . . 90

6 Superposition for Bounded Domains 93
6.1 Introduction . . . 93
6.2 Ground Horn Superposition . . . 96
6.3 A Calculus for $\mathcal{T}$-Unsatisfiability . . . 97
6.3.1 Calculus Rules . . . 97
6.3.2 Soundness and Refutational Completeness . . . 100
6.3.3 Redundancy in Detail . . . 104
6.3.4 Model Extraction . . . 112
6.3.5 Termination . . . 114
6.3.6 Extensions . . . 118
6.4 Combinations with First-Order Theories . . . 119
6.5 Summary . . . 122

7 On the Proliferation of an AC Deletion Rule 123
7.1 Background . . . 124
7.2 Ordered Completion . . . 125
7.3 Proofs in Ground Confluent Systems . . . 132
7.4 A Ground Confluent System for AC . . . 136
7.5 A Deletion Rule for AC Theories . . . 138
7.6 AC Deletion in a Clausal Setting . . . 143

8 Future Directions 149

Bibliography . . . 153

1 Introduction

> *Formal logic provides a mathematical foundation for many areas of computer science. Significant progress has been made in the challenge of making computers perform non-trivial logical reasoning, be it fully automatic, or in interaction with humans.*
>
> *In the last years it has become more and more evident that theory-specific reasoners, and in particular decision procedures, are extremely important in many applications of such deduction tools. General-purpose reasoning methods such as resolution or paramodulation alone are not efficient enough to handle the needs of real-world applications.*
>
> Proceedings of the 2007 Dagstuhl Seminar
> "Deduction and Decision Procedures" [BCGN07]

A tension between two apparently different approaches to automating deduction is reflected in the introductory quotation. These approaches were coined "big engines vs. little engines of proof" by Natarajan Shankar [Sha02] in an invited talk at the 2002 Federated Logic Conference on the question which sort of automated deduction fits more closely the needs of verification: either full first-order reasoning say with resolution and its refinements like superposition, or Nelson-Oppen style combinations of decision procedures for specific domains. Each style comes with its pros and cons: The latter is a push-button technology, but covers only particular theories, and usually is limited to the universal fragment. In contrast, the former offers a richer expressiveness, but in general is a semi-decision procedure only.

The contributions in this thesis advocate that these two strands of research can interplay in subtle and sometimes unexpected ways, such that mutual pervasion can lead to intriguing results:

(i) Superposition can be run *on top of* decision procedures, which is demonstrated for the class of Shostak theories [Sho84], thereby incorporating a little engine into a big one (Chap. 3).

Shostak introduced a congruence closure procedure that can be combined with decision procedures for other theories, provided that these theories have canonizers and solvers. The latter is a unitary unification algorithm that transforms an equation either into an equivalent set of equations with variables on the left-hand side, or into the empty clause if the equation is unsatisfiable. The former is a procedure that transforms every term into some normal form with respect to the given theory.

Within the superposition framework, these main ingredients of Shostak's method become simplification devices that allow us to replace theory equality by syntactic equality and to deal efficiently with overlaps between theory axioms and non-theory axioms, because coherence pairs just become trivial. These results have previously been published in [GHW03].

In Chapter 7, we present another instance of decision procedures within superposition. We show that ground confluent rewrite systems, which decide entailment problems in equational logic, can be harnessed for detecting redundancies in superposition derivations. Originally, this criterion was formulated in the context of completion in [Hil00, Chap. 6.1] for the WALDMEISTER system. It has spread since then and now is applied within superposition-based theorem provers, but without correctness proof so far.

After a recapitulation of completion, we give a new proof of the abovementioned result, which exploits that, given ground confluence, one can always join ground instances of equivalent terms just by rewriting the skeleton parts. Thanks to this more abstract proceeding, our proof closes a small gap unnoticed so far. Moving over to superposition, we demonstrate that the criterion is not correct with respect to the standard notion of redundancy. We show that this can be fixed with a refined literal ordering. Interestingly, the latter can be extended such that superposition redundancy subsumes completion redundancy.

(ii) Vice versa, superposition can be employed as proof-theoretic means *underneath* decision procedures in order to justify their Nelson-Oppen style combination [NO79]: The correctness of the Nelson-Oppen procedure is reestablished as an instance of the completeness of the superposition calculus (Chap. 4).

In doing so, superposition is employed as a particular means of generating models. Notably, given two clause sets over disjoint signatures without isolated variable occurrences, if each of them is saturated and does not contain the empty clause, so is their union, which therefore is satisfiable as well: Inferences different from the superposition rule are unary and do not produce conclusions in the combined signature. The superposition rule behaves the same as long as the terms in the clauses that are to be unified both start

with a function symbol. Finally, with stable infiniteness one can get rid of unshielded variable occurrences. With a suitable fine-tuning of the calculus parameters, this result can be adapted to the Nelson-Oppen setting where the signatures share a finite number of free constants. The material in this chapter has appeared in [Hil04] before.

(iii) Finally, superposition can be used *as* a decision procedure for many interesting theories, thereby turning a big engine into a little one. This usually happens by the formulation of adequate inference and simplification strategies, such that the combination with reasoning outside the particular theory comes more or less for free. Here two theories are studied: fixed-size bitvectors, and bounded domains.

The origins of the approach to bitvector reasoning (Chap. 5) stem from a case study in processor verification, in the context of the Verisoft project [Ver08]. In contrast to using dedicated decision procedures, the focus in the case study was on the question what could be achieved with first-order methods. Since these are less specialized, weaker performance was to be expected, and a priori those methods are only semi-decision procedures. On the other hand, first-order methods offer higher expressiveness: Arbitrary first-order formulae can be used instead of only universal ones. Furthermore, non-theory operators can be introduced, and one can even give axioms for them. This allows to abstract over implementation details. Another advantage of first-order methods is that bitvectors can freely be combined with arbitrary other sorts that capture for example other data types.

Eventually, the datapath of a DLX-like processor could be proved correct with the superposition-based theorem prover SPASS, in a fully automatic fashion. This success was possible because of a new encoding for the theory of bitvectors. For this new theory presentation, superposition with standard simplifications decides the validity of universal formulae. Hence SPASS just off-the-shelf implements this decision procedure.

Reasoning about bounded domains in resolution calculi is often painful. Despite the principal decidability, superposition implementations typically will not terminate. The starting point for a refinement of superposition in Chapter 6 is the observation that lifting in the case of bounded domains can be made more economical: A variable needs to stand no longer for any ground term, but just for the finitely many digits that represent the domain. Conversely, inferences involving a most general unifier σ only have to be considered if the range of σ consists of variables and digits. In other words, no complex unifiers are needed; and inferences do not increase the number of variables. Secondly, for any non-ground inference one can easily determine those instantiations that satisfy its ordering constraints. Thirdly,

redundancy also needs to refer to digit instances only, such that stronger simplifications become possible in some situations, but compatibility with the corresponding notion of standard superposition is mostly preserved. The price for these achievements is negligible: The cardinality-bounding axiom needs to be exchanged for its functional instances in order to not lose completeness.

This lifting modification applies to the family of superposition calculi. Soundness and refutational completeness are preserved, which is shown for a domain-specific calculus configuration in which non-Horn clauses are dealt with not by equality factoring, but by aggressive splitting. Combining ordered rewriting with some instantiation, a decision procedure for satisfiability modulo the cardinality bound is obtained, which decides the Bernays-Schönfinkel class as well. This solves a further classical decidability problem by superposition. In addition, the lifting modification is also applicable to bounded sorts in combination with arbitrary other, potentially infinite sorts.

For discussions of related work, see the respective chapters. We conclude in Chapter 8 with a brief summary and directions for future work.

2 Formal Preliminaries

In this chapter, we summarize the logical foundations of first-order reasoning and introduce some basic notions of theory reasoning and of rewriting. We use a many-sorted framework without subsorts or overloading, and equality is built into the logic.

2.1 Syntax of First-Order Logic

Definition 2.1 A many-sorted *signature* is a tuple $\Sigma = (\mathscr{S}, \mathscr{P}, \mathscr{F}, \mathscr{V}, \tau)$ where:

(i) $\mathscr{S}$ is a non-empty set of *sort symbols*;
(ii) $\mathscr{P}$ is a set of *predicate symbols*;
(iii) $\mathscr{F}$ is a set of *function symbols* or *operators*;
(iv) $\mathscr{V}$ is a set of *variable symbols* or *variables*;
(v) τ is a *sorting map* from $\mathscr{P} \cup \mathscr{F} \cup \mathscr{V}$ to $\mathscr{S}^*$ such that
  (a) $\tau(\mathscr{F}) \subseteq \mathscr{S}^+$,
  (b) $\tau(\mathscr{V}) \subseteq \mathscr{S}$,
  (c) $(\tau|_{\mathscr{V}})^{-1}(S)$ is infinite for every $S \in \mathscr{S}$;
(vi) the sets $\mathscr{S}, \mathscr{P}, \mathscr{F}, \mathscr{V}$ are disjoint.

By condition (v) (c) there are infinitely many variables for each sort. We frequently employ S, T, U and V as sort symbols, P and Q as predicate symbols, f and g as function symbols, x and y as variables. If $\tau(P) = S_1 \ldots S_n$ or $\tau(f) = S_1 \ldots S_n T$ where $n \geq 0$, then we call n the *arity* of P or f, the sequence $S_1 \ldots S_n$ the *domain sorts* of P or f, and T the *codomain sort* of f. We call f a *constant* in case that $n = 0$, and an *operator* otherwise.

Definition 2.2 Terms, atoms, formulae, and sentences over a signature Σ are defined as follows:

(i) The set $\mathscr{T}_S(\Sigma)$ of *terms of sort* S is the smallest set containing
  (a) x whenever $\tau(x) = S$,

(b) $f(t_1, \ldots, t_n)$ whenever $\tau(f) = T_1 \ldots T_n\, S$ and $t_i \in \mathscr{T}_{T_i}(\Sigma)$ for all i.

(ii) The set of *terms* is $\mathscr{T}(\Sigma) = \bigcup_{S \in \mathscr{S}} \mathscr{T}_S(\Sigma)$.

(iii) The set $\mathscr{A}(\Sigma)$ of *atoms* is the smallest set containing
(a) $s \simeq t$ whenever $s, t \in \mathscr{T}_S(\Sigma)$ for some $S \in \mathscr{S}$,
(b) $P(t_1, \ldots, t_n)$ whenever $\tau(P) = S_1 \ldots S_n$ and $t_i \in \mathscr{T}_{S_i}(\Sigma)$ for all i.
The former are called *equations*, the latter *predicative atoms*.

(iv) The set $\mathscr{F}(\Sigma)$ of *formulae* is the smallest set containing
(a) every atom of $\mathscr{A}(\Sigma)$,
(b) the logical constants $\top$ and $\bot$,
(c) applications of *connectives:*
$\neg\phi, \phi \wedge \psi, \phi \vee \psi, \phi \to \psi, \phi \leftrightarrow \psi$ whenever $\phi, \psi \in \mathscr{F}(\Sigma)$,
(d) *quantifier* applications:
$\forall x.\, \phi, \exists x.\, \phi$ whenever $\phi \in \mathscr{F}(\Sigma)$ and $x \in \mathscr{V}$.
The set $\mathscr{Q}(\Sigma)$ of *quantifier-free formulae* is the corresponding set satisfying (a) through (c).

(v) The set $\mathscr{F}_0(\Sigma)$ of *basic formulae* is the subset of formulae the connectives and quantifiers of which are a subset of $\{\neg, \wedge, \forall\}$. The set of *universal formulae* consists of all formulae $\forall x_1. \ldots \forall x_n.\, \phi$ for which ϕ is quantifier-free. A *positive formula* does not contain $\bot$ and uses as only connectives $\wedge$ and $\vee$.

(vi) We introduce *complementation* of connectives and quantifiers via $\overline{\wedge} = \vee$, $\overline{\vee} = \wedge$, $\overline{\forall} = \exists$ and $\overline{\exists} = \forall$.

(vii) The *length* $|\_|$ of a term is given via $|x| = 1$ and $|f(\vec{t})| = 1 + \Sigma_i |t_i|$. On atoms, let $|P(\vec{t})| = 1 + \Sigma_i |t_i|$ and furthermore $|s \simeq t| = |s| + |t|$.

(viii) An *expression* is a term or a formula. By the symbol $\equiv$ we denote syntactic equality of expressions. The i-th *component* $\_|_i$ of an expression is given by $P(t_1, ..., t_n)|_i \equiv f(t_1, ..., t_n)|_i \equiv t_i$, $(\neg\phi_1)|_1 \equiv (\phi_1 \otimes \phi_2)|_1 \equiv \phi_1$, $(Qx.\, \phi)|_1 \equiv x$ and $(\phi_1 \otimes \phi_2)|_2 \equiv (Qx.\, \phi_2)|_2 \equiv \phi_2$ for $\otimes \in \{\wedge, \vee, \to, \leftrightarrow\}$ and $Q \in \{\forall, \exists\}$.

(ix) A *position* is a finite sequence of positive integers, the empty one being denoted by λ. The *length* of a position is its length as a sequence.

(x) The *subexpression* $e|_p$ of an expression e at a position p is defined recursively via $e|_\lambda \equiv e$ and $e|_{i.q} \equiv e_i|_q$ where e_i is the i-th component of e. We say that $e|_p$ is a *strict* subexpression of e if $p \neq \lambda$. A *subformula* is a subexpression that is a formula, and a *subterm* is a subexpression that is a term.

(xi) We write $e[e']_p$ to indicate that e is a *context* of e', namely that $e|_p \equiv e'$, and (ambiguously) denote by $e[e'']_p$ the result of replacing the occurrence of e' at p in e by e''.

(xii) A binary relation ρ on terms or on formulae is *closed under contexts* or *stable under contexts* if $e\ \rho\ e'$ implies $e''[e]_p\ \rho\ e''[e']_p$ for all expressions e, e', e'' and positions p such that $e''[e]$ and $e''[e']$ are well-formed.

(xiii) The *free variables* of an expression are defined recursively via $\mathrm{free}(x) = \{x\}, \mathrm{free}(\forall x.\,\phi) = \mathrm{free}(\exists x.\,\phi) = \mathrm{free}(\phi) \setminus \{x\}$, $\mathrm{free}(e) = \bigcup_i \mathrm{free}(e_i)$ for the remaining expressions e provided they are compound, and $\mathrm{free}(e) = \emptyset$ otherwise.

(xiv) A *sentence* is a formula without free variables. If $\mathrm{free}(\phi) = \{x_1, \ldots, x_n\}$, then $\forall X.\,\phi$ is an abbreviation for the sentence $\forall x_1. \ldots \forall x_n.\,\phi$, the *universal closure* of ϕ. The *existential closure* is defined correspondingly.

(xv) For every variable x, the function $|\_|_x$ gives the number of occurrences of x within an argument term or atom.

(xvi) A *literal* is an atom or a negated atom. A *clause* is a disjunction of finitely many literals, possibly written as implication. Within clauses we usually consider the order of literals irrelevant to the syntactic equality relation $\equiv$, as well as identifying $s \simeq t$ and $t \simeq s$.

(xvii) A *Horn clause* is a clause with at most one positive literal. A *negative clause* consists of negative literals only.

(xviii) A *clause normal form* or *CNF* is a formula that consists of a sequence of universal quantifications applied to a conjunction of clauses. We identify the *empty clause* with $\bot$, and the *empty conjunction* of clauses with $\top$.

(xix) A formula is in *negation normal form* if it contains neither implications nor equivalences, and every negation is applied to an atom.

(xx) On clauses and terms, we use $\mathrm{var}(\_)$ as a synonym for $\mathrm{free}(\_)$ to denote the set of variables that occur in the expression. Terms and clauses without variables are called *ground terms* and *ground clauses*, respectively.

(xxi) A *substitution* σ is a map from a finite set $\mathscr{V}_0 \subseteq \mathscr{V}$ to $\mathscr{T}(\Sigma)$ such that $x\sigma \in \mathscr{T}_{\tau(x)}(\Sigma)$ for every $x \in \mathscr{V}_0$. Applications are written in postfix notation. We extend σ as identity to $\mathscr{V}$, and homomorphically to terms, atoms and formulae without quantifiers. For $Q \in \{\forall, \exists\}$ let $(Qx.\,\phi)\sigma \equiv Qy.\,(\phi\sigma')$ where y is a fresh variable and σ' equals σ except that $x\sigma' \equiv y$.

(xxii) If σ is a substitution, then $\mathrm{dom}\,\sigma$ is the set of all variables for which $x\sigma \not\equiv x$ holds, $\mathrm{ran}\,\sigma$ is the image of $\mathrm{dom}\,\sigma$ under σ, and $\mathrm{cdom}\,\sigma$ is the set of variables occurring in $\mathrm{ran}\,\sigma$. Furthermore σ is *ground* if $\mathrm{ran}\,\sigma$ is a set of ground terms; and σ *grounds* a term or a clause e if $e\sigma$ is a ground term or clause. We say that σ is a *variable renaming* if it is a bijection on $\mathrm{dom}\,\sigma$.

(xxiii) Two terms s and t are *unifiable* if $s\sigma \equiv t\sigma$ for some substitution σ, which is called a *unifier* of s and t. A *most general unifier* of s and t, denoted by $\mathrm{mgu}(s,t)$, is any unifier σ such that any other unifier ρ has a presentation $\rho = \sigma\sigma'$; notably $\mathrm{mgu}(s,t)$ is unique up to renaming of

variables.

(xxiv) A clause C *subsumes* a clause D if $D \equiv C\sigma \vee D'$ for some substitution σ and clause D'.

Terms will be denoted by the meta variables s and t, variables by x and y. Sequences $t_1, \ldots, t_n$ of terms will freely be abbreviated as vectors $\vec{t}$. Operations on terms like substitutions are silently extended to term vectors by componentwise application, and vectors of functions are composed componentwise.

Atoms are denoted by A and B, predicative atoms by $P(\vec{t})$, literals by L, clauses by C and D, formulae by ϕ and ψ, and sets of formulae by Γ and Δ. Parentheses are used to indicate which way an expression is meant to have been generated. We stipulate that substitution and context bind tightest, followed by negation, then by conjunction and disjunction, then by implication and equivalence, and finally by the quantifiers. As shorthand notation for $\neg(s \simeq t)$ we write $s \not\simeq t$, which is called a *disequation*. Furthermore, $s \bowtie t$ arbitrarily denotes one of the literals $s \simeq t$ and $s \not\simeq t$, and ${}^{\pm}P(\vec{t})$ stands for any of the literals $P(\vec{t})$ and $\neg P(\vec{t})$. Finally we stipulate that for every sort there shall exist at least one ground term of that sort.

2.2 Semantics of First-Order Logic

Definition 2.3 A Σ-*algebra* is a function $\mathcal{A}$ on $\mathscr{S} \cup \mathscr{P} \cup \mathscr{F}$ that maps

(i) every sort symbol S to a non-empty *carrier set*,

(ii) every predicate symbol P where $\tau(P) = S_1 \ldots S_n$ to a relation on $\mathcal{A}(S_1) \times \ldots \times \mathcal{A}(S_n)$,

(iii) every function symbol f where $\tau(f) = S_1 \ldots S_n\, T$ to a function of type $\mathcal{A}(S_1) \times \ldots \times \mathcal{A}(S_n) \to \mathcal{A}(T)$.

Regarding the case $n = 0$, we identify the empty Cartesian product of sets with $\{\emptyset\}$. Hence $\mathcal{A}(f)$ is a function from a one-element domain to $\mathcal{A}(T)$, and $\mathcal{A}(P)$ is either $\emptyset$ or $\{\emptyset\}$, which is reminiscent of a propositional variable.

We will occasionally write $S^{\mathcal{A}}, P^{\mathcal{A}}, f^{\mathcal{A}}$ instead of $\mathcal{A}(S), \mathcal{A}(P), \mathcal{A}(f)$, and denote Σ-algebras by $\mathcal{A}$ and $\mathcal{B}$. Greek letters α, β will be used for elements of carrier sets.

Definition 2.4 Given a Σ-algebra $\mathcal{A}$, we define how to interpret terms and formulae within $\mathcal{A}$.

(i) An $\mathcal{A}$-*assignment* is a map μ on $\mathscr{V}$ such that $\mu(x) \in \tau(x)^{\mathcal{A}}$.

(ii) For every $x \in \mathscr{V}$ and $\alpha \in \tau(x)^{\mathcal{A}}$, the assignment μ^x_α is identical to μ except that x is mapped onto α.

(iii) Every assignment μ induces an *interpretation* $\mathcal{A}_\mu$ as homomorphic extension of μ on $\mathscr{T}(\Sigma)$, i. e. $\mathcal{A}_\mu|_\mathscr{V} = \mu$ and $\mathcal{A}_\mu(f(\vec{t})) = f^\mathcal{A}(\mathcal{A}_\mu(\vec{t}))$.

(iv) $\mathcal{A}_\mu$ is extended to atoms via:
  (a) $\mathcal{A}_\mu(s \simeq t)$ iff $\mathcal{A}_\mu(s) = \mathcal{A}_\mu(t)$ holds in $S^\mathcal{A}$ where $s, t \in \mathscr{T}_S(\Sigma)$,
  (b) $\mathcal{A}_\mu(P(\vec{t}))$ iff $P^\mathcal{A}(\mathcal{A}_\mu(\vec{t}))$ holds.

(v) $\mathcal{A}_\mu$ is extended to formulae by stipulating that Boolean connectives and quantifiers are interpreted as such in the carrier sets. In particular, we have $\mathcal{A}_\mu(\forall x.\, \phi)$ iff $\mathcal{A}_{\mu^x_\alpha}(\phi)$ for every $\alpha \in \tau(x)^\mathcal{A}$. Furthermore, $\mathcal{A}_\mu(\top)$ is always true and $\mathcal{A}_\mu(\bot)$ never.

(vi) An interpretation $\mathcal{A}_\mu$ *satisfies* a formula ϕ if $\mathcal{A}_\mu(\phi)$ holds, which is denoted by $\mathcal{A}_\mu \models \phi$. Then ϕ is called *satisfiable*, and $\mathcal{A}_\mu$ is called a *model of* ϕ.

(vii) If ϕ is a sentence, then let furthermore $\mathcal{A} \models \phi$ if $\mathcal{A}_\mu \models \phi$ holds for some $\mathcal{A}$-assignment μ. In this case we say that the algebra $\mathcal{A}$ is a *model of* ϕ.

(viii) The notions of satisfiability and model are lifted to sets of formulae by viewing them as (possibly infinite) conjunctions of formulae.

(ix) A formula ϕ is *valid*, denoted by $\models \phi$, if ϕ holds in every interpretation. Alternatively we say that ϕ is a *tautology*.

(x) A clause is called a *syntactic tautology* if it contains a literal $t \simeq t$, or an atom along with its negation.

(xi) A set Γ of formulae *entails* a formula ϕ, written as $\Gamma \models \phi$, if every model of Γ is a model of ϕ. The *deductive closure* $\mathrm{Ded}(\Gamma)$ is the set of all the formulae that Γ entails.

(xii) Two formulae ϕ and ψ are *equivalent*, denoted by $\phi \models\!\models \psi$, if they have the same models; and so are two sets of formulae. Two formulae, or sets thereof, are *equisatisfiable* if they are either both satisfiable, or both unsatisfiable.

Assignments will be denoted by μ and ν. The interpretation of sentences is independent of the assignment at hand, which motivates Def. 2.4 (vii). Formulae behave with respect to satisfiability like their existential closure, and with respect to validity like their universal closure. That is, we have $\Gamma \models \phi$ if and only if $\Gamma \models \forall X.\, \phi$.

Definition 2.5 Let $\mathcal{A}$ and $\mathcal{B}$ denote two Σ-algebras.

(i) A mapping $\varphi\colon \bigcup\{S^\mathcal{A}\colon S \in \mathscr{S}\} \to \bigcup\{S^\mathcal{B}\colon S \in \mathscr{S}\}$ is a *homomorphism from $\mathcal{A}$ to $\mathcal{B}$* if
  (a) $\varphi(S^\mathcal{A}) \subseteq \varphi(S^\mathcal{B})$ for every $S \in \mathscr{S}$,
  (b) $\varphi(f^\mathcal{A}(\vec{\alpha})) = f^\mathcal{B}(\varphi(\vec{\alpha}))$ for every $f \in \mathscr{F}$, $\tau(f) = \vec{S}\,T$, $\vec{\alpha} \in \vec{S}^\mathcal{A}$,
  (c) $P^\mathcal{A}(\vec{\alpha})$ implies $P^\mathcal{B}(\varphi(\vec{\alpha}))$ for every $P \in \mathscr{P}$, $\tau(P) = \vec{S}$, $\vec{\alpha} \in \vec{S}^\mathcal{A}$.

The algebra $\mathcal{A}$ is called *homomorphic to* $\mathcal{B}$ if such a homomorphism exists.

(ii) A homomorphism φ from $\mathcal{A}$ to $\mathcal{B}$ is an *isomorphism* if there exists an inverse homomorphism from $\mathcal{B}$ to $\mathcal{A}$. We say that $\mathcal{A}$ and $\mathcal{B}$ are *isomorphic* if there is an isomorphism from one to the other.

Proposition 2.6 Consider an isomorphism φ from $\mathcal{A}$ to $\mathcal{B}$, an $\mathcal{A}$-assignment μ and the $\mathcal{B}$-assignment $\nu = \varphi \circ \mu$. Then $\mathcal{A}_\mu \models \phi$ iff $\mathcal{B}_\nu \models \phi$, for every Σ-formula ϕ.

Proof: First one shows that $\varphi \circ \mathcal{A}_\mu = \mathcal{B}_\nu$ on $\mathscr{T}(\Sigma)$, by induction over the term structure: Regarding variables, we evidently have $\varphi(\mathcal{A}_\mu(x)) = \varphi\mu(x) = \nu(x) = \mathcal{B}_\nu(x)$. As to complex terms, we get the chain of identities $\varphi(\mathcal{A}_\mu(f(\vec{t}))) = \varphi(f^{\mathcal{A}}(\mathcal{A}_\mu(\vec{t}))) = f^{\mathcal{B}}(\varphi(\mathcal{B}_\nu(\vec{t}))) = f^{\mathcal{B}}(\mathcal{B}_\nu(\vec{t})) = \mathcal{B}_\nu(f(\vec{t}))$ because φ is a homomorphism and by induction hypothesis.

The next step is to prove the statement at hand for atomic formulae ϕ. For an equation $s \simeq t$ we can argue that $\mathcal{A}_\mu(s) = \mathcal{A}_\mu(t)$ iff $\varphi(\mathcal{A}_\mu(s)) = \varphi(\mathcal{A}_\mu(t))$ iff $\mathcal{B}_\nu(s) = \mathcal{B}_\nu(t)$ by injectivity of φ and by the preceding paragraph. For predicative atoms we have $P^{\mathcal{A}}(\mathcal{A}_\mu(\vec{t}))$ iff $P^{\mathcal{B}}(\varphi(\mathcal{A}_\mu(\vec{t})))$ iff $P^{\mathcal{B}}(\mathcal{B}_\nu(\vec{t}))$ because φ is an isomorphism and because $\varphi \circ \mathcal{A}_\mu = \mathcal{B}_\nu$ on terms.

Finally, we induct over the formula structure. If ϕ is $\top$ or $\bot$, then we are done. In the case of disjunctions we have $\mathcal{A}_\mu \models \phi \wedge \psi$ iff both $\mathcal{A}_\mu \models \phi$ and $\mathcal{A}_\mu \models \psi$, which inductively is equivalent to $\mathcal{B}_\nu \models \phi$ and $\mathcal{B}_\nu \models \psi$. Concerning universal quantifications, note first that $\varphi \circ \mu^x_\alpha = \nu^x_{\varphi(\alpha)}$. This gives the chain of equivalences $\mathcal{A}_\mu \models \forall x.\, \phi$ iff $\mathcal{A}_{\mu^x_\alpha} \models \phi$ for every $\alpha \in \tau(x)^{\mathcal{A}}$ iff, inductively, $\mathcal{B}_{\nu^x_{\varphi(\alpha)}} \models \phi$ for every such α. The latter is the same as $\mathcal{B}_{\nu^x_\beta} \models \phi$ for every $\beta \in \tau(x)^{\mathcal{B}}$ because φ is surjective. The remaining cases are similar. $\square$

Definition 2.7 Let $\Sigma = (\mathscr{S}, \mathscr{P}, \mathscr{F}, \mathscr{V}, \tau)$ and $\Sigma' = (\mathscr{S}', \mathscr{P}', \mathscr{F}', \mathscr{V}', \tau')$ denote two signatures such that $(\mathscr{S}, \mathscr{P}, \mathscr{F}, \tau) \subseteq (\mathscr{S}', \mathscr{P}', \mathscr{F}', \tau')$ and $\mathscr{V} = \tau^{-1}(\mathscr{S}) \cap \mathscr{V}'$ hold.[1]

(i) In this case Σ is called a *subsignature of* Σ' and in turn Σ' a *supersignature of* Σ.

(ii) We say that a Σ-algebra $\mathcal{A}$ is a Σ-*reduct* of a Σ'-algebra $\mathcal{B}$ and that $\mathcal{B}$ is a Σ'-*expansion* of $\mathcal{A}$ if $\mathcal{A} = \mathcal{B}$ on $\mathscr{S} \cup \mathscr{P} \cup \mathscr{F}$.

Clearly every Σ'-algebra has one and only one Σ-algebra which is a Σ-reduct of it, the latter being obtained from the former simply by restriction to $\mathscr{S} \cup \mathscr{P} \cup \mathscr{F}$.

[1] The second condition ensures that variables of the sorts contained in $\mathscr{S}$ are common to Σ and Σ'.

2.3 Theory Reasoning Bits

Definition 2.8 Furthermore, we need a number of notions around theory reasoning:

(i) A Σ-*theory* $\mathcal{T}$ is a set of sentences over Σ.[2]

(ii) A formula ϕ or a set Γ of formulae is $\mathcal{T}$-*satisfiable* if ϕ or Γ is satisfiable in a $\mathcal{T}$-*model*, that is, in a model of $\mathcal{T}$.

(iii) A formula ϕ is $\mathcal{T}$-*valid* if $\mathcal{T} \models \phi$.

(iv) Two formulae ϕ and ψ are $\mathcal{T}$-*equivalent* if $\mathcal{T} \models \phi \leftrightarrow \psi$ is true. In that case we write $\phi \models\!\models_{\mathcal{T}} \psi$.

(v) A binary relation $\Longrightarrow$ on formulae *preserves* $\mathcal{T}$-*equivalence* if it is contained in $\models\!\models_{\mathcal{T}}$.

(vi) Consider a signature Σ with supersignature Σ'. For a Σ-theory $\mathcal{S}$ and a Σ'-theory $\mathcal{T}$, if $\mathrm{Ded}(\mathcal{S}) \subseteq \mathrm{Ded}(\mathcal{T})$ holds, then we say that $\mathcal{S}$ is a *subtheory of* $\mathcal{T}$, and $\mathcal{T}$ is an *extension* or a *supertheory of* $\mathcal{S}$.

(vii) Given signatures Σ and Σ', a *formula transformation* is any mapping from $\mathscr{F}(\Sigma)$ to $\mathscr{F}(\Sigma')$. For brevity, a formula transformation from one theory to another is any mapping between the formulae of the respective signatures.

(viii) Let Ξ denote a formula transformation from a Σ-theory $\mathcal{S}$ to a Σ'-theory $\mathcal{T}$, and Γ a set of Σ-formulae. Then we say that Ξ on the set Γ is:
   (a) *sound* if $\mathcal{T} \models \Xi(\phi)$ entails $\mathcal{S} \models \phi$ for every $\phi \in \Gamma$,
   (b) *complete* if $\mathcal{S} \models \phi$ implies $\mathcal{T} \models \Xi(\phi)$ for every $\phi \in \Gamma$, and
   (c) *correct* if it is both sound and complete.
   If omitted, Γ defaults to $\mathscr{F}(\Sigma)$.

(ix) A Σ-theory $\mathcal{T}$ is *complete* if for every Σ-sentence ϕ either $\mathcal{T} \models \phi$ or $\mathcal{T} \models \neg\phi$ holds.

Theories will be denoted by $\mathcal{S}$ and $\mathcal{T}$. Note that the notions of $\mathcal{T}$-satisfiability, $\mathcal{T}$-validity, $\mathcal{T}$-equivalence reduce to the standard notions of satisfiability, validity, equivalence in case the theory $\mathcal{T}$ is empty.

We formulate a variant of the compactness theorem for our setting:

Lemma 2.9 Let $\mathcal{T}$ denote a Σ-theory and Γ a set of Σ-formulae. Then the following are equivalent:

(i) Γ is $\mathcal{T}$-satisfiable.

(ii) Every finite subset of Γ is $\mathcal{T}$-satisfiable.

Proof: The compactness theorem of first-order logic is also valid in the many-sorted setting (see e. g. [End02, p. 298]). Therefore proving the implication

[2] Some authors require additionally that a theory is deductively closed. The looser definition as employed here can be found for example in [CK73, p. 36].

(ii)→(i) reduces to showing that every finite subset Δ of $\mathcal{T} \cup \Gamma$ is satisfiable. Now, the set $\Delta' = \Delta \setminus \mathcal{T}$ is a finite subset of Γ, hence $\mathcal{T}$-satisfiable. In other words, $\Delta' \cup \mathcal{T} = (\Delta \setminus \mathcal{T}) \cup \mathcal{T}$ is satisfiable, and hence also the subset Δ is. □

Definition 2.10 The following notions are often used as preconditions when combining theories:

(i) A theory $\mathcal{T}$ is called *stably infinite* if the following are equivalent for every quantifier-free formula ϕ:
   (a) ϕ is $\mathcal{T}$-satisfiable.
   (b) ϕ is satisfiable in an infinite $\mathcal{T}$-model.

(ii) We say that a theory $\mathcal{T}$ is *convex* if the first condition implies the second for every non-negative clause $C \equiv \neg A_1 \vee \ldots \vee \neg A_m \vee B_1 \vee \ldots \vee B_n$:
   (a) $\mathcal{T} \models \forall X.\ \bigwedge_i A_i \rightarrow \bigvee_j B_j$
   (b) $\bigvee_j \mathcal{T} \models \forall X.\ \bigwedge_i A_i \rightarrow \quad B_j$

For example, any theory with only infinite models is trivially stably infinite. If a theory is closed under products, meaning that the Cartesian product of two theory models is a theory model again, then it is convex; this applies in particular to Horn theories.

In order to relate the two notions, we start with a proposition on the cardinality of the models of convex theories:

Proposition 2.11 If $\mathcal{T}$ is a convex theory and ϕ is a quantifier-free formula, then every $\mathcal{T}$-model of ϕ is a singleton or infinite.

Proof: Without loss of generality ϕ is a conjunction of literals: It is equivalent to its disjunctive normal form $\psi_1 \vee \ldots \vee \psi_k$; and every ϕ-model is also a model of some ψ_i. So let $\phi \equiv \neg A_1 \wedge \ldots \wedge \neg A_m \wedge B_1 \wedge B_1 \wedge \ldots \wedge B_n$; hence we have $\neg\phi \models\!\models \bigwedge_j B_j \rightarrow \bigvee_i A_i$. Let us note that $\mathcal{T} \not\models \forall X.\ \bigwedge_j B_j \rightarrow \bigvee_i A_i$ holds, because otherwise $\mathcal{T} \models \forall X.\ \neg\phi$ were true and ϕ had no $\mathcal{T}$-models at all. We assume now that $\mathcal{T}, \exists X.\ \phi$ has only finite models, and have to show that these are singletons.

By compactness of first-order logic there exists an upper bound N of the size of the models (see for example [CK73, p. 67]). Hence of any $N+1$ variables at least two coincide, such that $\mathcal{T}, \exists X.\ \phi$ satisfies $\forall X.\ \bigvee_{i \neq j} x_i \simeq x_j$ where i, j range over $[0; N]$ and all variables x_i are fresh. This can be rephrased as $\mathcal{T} \models \forall X.\ (\bigwedge_j B_j) \rightarrow (\bigvee_i A_i \vee \bigvee_{i \neq j} x_i \simeq x_j)$. By convexity and $\mathcal{T} \not\models \forall X.\ \bigwedge_j B_j \rightarrow \bigvee_i A_i$ we now obtain $\mathcal{T} \models \forall X.\ \bigwedge_j B_j \rightarrow x_i \simeq x_j$; therefore we also have $\mathcal{T} \models (\forall X.\ \neg\phi) \vee (\forall X.\ x_i \simeq x_j)$ and finally $\mathcal{T}, \exists X.\ \phi \models \forall x.\ \forall y.\ x \simeq y$. □

This classification of model cardinalities gives rise to the following connection:

Corollary 2.12 ([BDS02]) Every convex theory without singleton models is stably infinite.

Finally we model extending a theory by means of definitions.

Assumption 2.13 For the rest of this subsection we fix some signature Σ from which we obtain a signature Σ' by adding a new function symbol f, the latter mapping from $S_1 \dots S_n$ to T. Furthermore, let $x_1, \dots, x_n, y$ denote distinct variables of sort $S_1, \dots, S_n, T$, respectively. We will study a particular relationship between a Σ-theory $\mathcal{T}$ and a Σ'-theory $\mathcal{T}'$.

Imagine that for every value of $\vec{x}$ some Σ-formula ϕ is true for one and only one value of y. Then we can give a name to this particular y, for example $f(\vec{x})$. This is the idea behind the following notion.

Definition 2.14 Consider a Σ-formula ϕ such that $\mathrm{free}(\phi) \subseteq \{\vec{x}, y\}$, and that $\mathcal{T} \models \forall \vec{x}.\, \exists y.\, \phi \wedge \forall z.\, \phi\{y \mapsto z\} \rightarrow y \simeq z$. Then we call $\mathcal{T}'$ a *definitional extension of* $\mathcal{T}$ if it is equivalent to $\mathcal{T} \cup \{\forall X.\, \phi \leftrightarrow y \simeq f(\vec{x})\}$.

Proposition 2.15 Let us study a definitional extension $\mathcal{T}'$ of $\mathcal{T}$.

(i) $\mathcal{T}$-validity and $\mathcal{T}'$-validity coincide on Σ-formulae.
(ii) For every $\mathcal{T}$-model, there is exactly one expansion to a $\mathcal{T}'$-model.
(iii) Σ'-formulae can effectively be transformed to Σ-formulae such that
  (a) the output is $\mathcal{T}$-valid if the input is $\mathcal{T}'$-valid,
  (b) the transformation preserves $\mathcal{T}'$-equivalence, and
  (c) it is invariant on $\mathcal{T}$-formulae.
(iv) Two $\mathcal{T}'$-models $\mathcal{A}'$ and $\mathcal{B}'$ are isomorphic iff their Σ-reducts $\mathcal{A}$ and $\mathcal{B}$ are.

Proof:

(i) On the one hand $\mathcal{T} \models \phi$ implies $\mathcal{T} \cup \{\forall X.\, \phi \leftrightarrow y \simeq f(\vec{x})\} \models \phi$ and $\mathcal{T}' \models \phi$, for any Σ-formula ϕ. On the other hand, let $\mathcal{T}' \models \phi$ and consider an arbitrary $\mathcal{T}$-model $\mathcal{A}_\mu$ with expansion $\mathcal{B}_\mu$. Inductively $\mathcal{A}_\mu$ and $\mathcal{B}_\mu$ agree on Σ-terms and Σ-formulae. So $\mathcal{B}_\mu \models \phi$ implies $\mathcal{A}_\mu \models \phi$.
(ii) Given a $\mathcal{T}$-model $\mathcal{A}$, an expansion $\mathcal{B}$ is determined by the interpretation of f. Let us study $f^{\mathcal{B}}(\vec{\alpha})$. If $\mu(\vec{x}) = \vec{\alpha}$, then $\mathcal{A}_{\mu^y_\beta} \models \phi$ for one and only one domain element β by the requirement on $\mathcal{T}$. Since $\mathcal{A}$ and $\mathcal{B}$ agree on Σ-formulae, the same is true for $\mathcal{B}_{\mu^y_\beta} \models \phi$. Hence we obtain by the condition on $\mathcal{T}'$ that $\mathcal{B}_{\mu^y_\beta} \models y \simeq f(\vec{x})$, and therefore $\beta = f^{\mathcal{B}}(\vec{\alpha})$.

(iii) Consider a formula $\psi[f(\vec{t})]$. Let us assume that y does not show up in ψ, because one can use a renamed variant of ϕ otherwise. We simply abstract out the occurrence of f with the variable y bound by ϕ. More precisely, the formulae $\psi[f(\vec{t})]$ and $\exists y.\, \phi\{\vec{x} \mapsto \vec{t}\} \wedge \psi[y]$ are $\mathcal{T}'$-equivalent by the assumption on $\mathcal{T}'$. If $f(\vec{t})$ was located at an innermost position, i. e. without any $f(\vec{s})$ within the t_i, then the second formula has one symbol f less than the right one. This way iterating the step eventually terminates with a Σ-formula ψ' which is $\mathcal{T}'$-equivalent to ψ, proving (b). We obtain (a) by (i) and (c) by construction of the transformation.

(iv) If $\mathcal{A}'$ and $\mathcal{B}'$ are isomorphic, then immediately $\mathcal{A}$ and $\mathcal{B}$ are because Σ' contains all the symbols of Σ. In turn, let us assume that φ is an isomorphism from $\mathcal{A}$ to $\mathcal{B}$. In order to show that it is also one from $\mathcal{A}'$ to $\mathcal{B}'$ we only need to prove that $\varphi(f^{\mathcal{A}'}(\vec{\alpha}))$ equals $f^{\mathcal{B}'}(\varphi(\vec{\alpha}))$ for any $\vec{\alpha} \in \vec{S}^{\mathcal{A}}$. Let μ denote an assignment such that $\vec{x} \overset{\mu}{\longmapsto} \vec{\alpha}$ and $y \overset{\mu}{\longmapsto} f^{\mathcal{A}'}(\vec{\alpha})$. Then we have $\mathcal{A}'_\mu \models y \simeq f(\vec{x})$ and hence $\mathcal{A}_\mu \models \phi$, which by Prop. 2.6 implies that $\mathcal{B}_{\varphi\mu} \models \phi$ and $\mathcal{B}'_{\varphi\mu} \models y \simeq f(\vec{x})$. This means that $\varphi(\mu(y)) = f^{\mathcal{B}'}(\varphi(\mu(\vec{x})))$ and finally $\varphi(f^{\mathcal{A}'}(\vec{\alpha})) = f^{\mathcal{B}'}(\varphi(\vec{\alpha}))$.

□

The above proposition states in particular that one can effectively reduce $\mathcal{T}'$-validity to $\mathcal{T}$-validity. For the relation between $\mathcal{T}$ and $\mathcal{T}'$ pointed out in (i) it is customary to say that $\mathcal{T}'$ is a *conservative extension of* $\mathcal{T}$. The notion of definitional extension can straightforwardly be extended to deal with predicate symbols.

A simple form of definitional extension is when a new function symbol is *explicitly defined* via an equation, assigning just an expression in the original signature:

Proposition 2.16 If $\mathcal{T}' = \mathcal{T} \cup \{\forall \vec{x}.\, f(\vec{x}) \simeq t\}$ where $t \in \mathscr{T}(\Sigma)$ and $\mathrm{free}(t) \subseteq \{\vec{x}\}$, then $\mathcal{T}'$ is a definitional extension of $\mathcal{T}$.

Proof: The formula $\forall \vec{x}.\, f(\vec{x}) \simeq t$ is equivalent to $\forall y.\, y \simeq t \leftrightarrow y \simeq f(\vec{x})$; so $\mathcal{T}'$ has an equivalent presentation as demanded provided ϕ denotes the formula $y \simeq t$. The condition on $\mathcal{T}$ then spells out as $\mathcal{T} \models \forall \vec{x}.\, \exists y.\, y \simeq t \wedge \forall z.\, z \simeq t \to y \simeq z$ and is obviously valid. □

Note that the transformation of $\mathcal{T}'$-formulae to $\mathcal{T}$-formulae given in the proof of Prop. 2.15 can be done simpler in this case, namely based on the $\mathcal{T}'$-equivalence of $\psi[f(\vec{s})]$ and $\psi[t\{\vec{x} \mapsto \vec{s}\}]$.

2.4 Rewriting and Orderings

Definition 2.17 The following notions apply to *abstract reduction systems* $\longrightarrow$, that is to binary relations over arbitrary sets M.

(i) The *composition* $\longrightarrow_1 \circ \longrightarrow_2$ of two relations $\longrightarrow_1$ and $\longrightarrow_2$ is given by $a \longrightarrow_1 \circ \longrightarrow_2 c$ if $a \longrightarrow_1 b \longrightarrow_2 c$ for some $b \in M$. The *n-fold iteration* $\longrightarrow^n$ is defined recursively via $\longrightarrow^0 = =$ and $\longrightarrow^{n+1} = \longrightarrow \circ \longrightarrow^n$. The *n-bounded iteration* $\longrightarrow^{\leq n}$ is given by $\bigcup_{i=0}^{n} \longrightarrow^i$.

(ii) We define various *closures* of $\longrightarrow$ as follows:
  (a) *symmetric:* $\longleftrightarrow = \longleftarrow \cup \longrightarrow$, where $\longleftarrow$ is the inverse of $\longrightarrow$
  (b) *transitive:* $\longrightarrow^+ = \bigcup_{i>0} \longrightarrow^i$
  (c) *reflexive-transitive:* $\longrightarrow^* = \longrightarrow^0 \cup \longrightarrow^+$
  (d) *reflexive-transitive-symmetric:* $\longleftrightarrow^* = (\longleftrightarrow)^*$

(iii) Two elements a and b are *convertible* if $a \longleftrightarrow^* b$, and *joinable*, denoted by $a \downarrow b$, if $a \longrightarrow^* \circ {}^*\!\longleftarrow b$. We say that $\longrightarrow$ is *Church-Rosser* if $\longleftrightarrow^* \subseteq \downarrow$, and *confluent* if ${}^*\!\longleftarrow \circ \longrightarrow^* \subseteq \downarrow$. The *local* versions of these properties are $\longleftrightarrow \subseteq \downarrow$ and $\longleftarrow \circ \longrightarrow \subseteq \downarrow$, respectively.

(iv) The relation $\longrightarrow$ is *terminating* or *well-founded* in case there is no infinite chain $c_1 \longrightarrow c_2 \longrightarrow \ldots$, and *convergent* provided it is confluent as well. The element b is a *normal form* of a, denoted by $a \longrightarrow^! b$, if $a \longrightarrow^* b$, but $b \longrightarrow c$ for no c. In that case b is called *irreducible*. Furthermore let $a{\downarrow} = b$ if b is the unique normal form of a.

(v) An *ordering* $\succ$ on M is a binary relation over M which is irreflexive and transitive. We denote its inverse by $\prec$ and its reflexive closure by $\succeq$. The ordering $\succ$ is *total* if $\succeq \cup \preceq = M \times M$.

(vi) A *quasi-ordering* $\succsim$ on M is a binary relation over M which is reflexive and transitive. Its *equivalence part* is $\approx \; = \; \succsim \cap \precsim$; its *strict part* is $\succ = \succsim \setminus \precsim$; and it is called *terminating* if its strict part is a terminating relation.

If $\succsim$ is a quasi-ordering, then $\succ$ is an ordering and $\approx$ an equivalence relation. So in the minimal case $\approx$ is just equality; and such quasi-orderings are preferrably denoted by $\succeq$, because they coincide with the reflexive closure of their strict part. Every quasi-ordering $\succsim$ satisfies $\succsim \circ \succ \; = \; \succ \circ \succsim \; = \; \succ$; and the relations $\succ$, $\approx$ and $\prec$ are mutually disjoint. The *lexicographic combination* of two quasi-orderings $\succsim_1$ and $\succsim_2$ is the relation $\succ_1 \cup (\approx_1 \cap \succsim_2)$.

Proposition 2.18 Quasi-orderings are closed under lexicographic combination, and so are terminating quasi-orderings. The equivalence part of the combination is always the intersection of those of the components.

Proof: Consider quasi-orderings $\succsim_1$ and $\succsim_2$ with lexicographic combination $\succsim$. The latter is transitive: $a \succ b \succ c$ implies $a \succsim_1 b \succsim_1 c$; and either one

of the steps is strict, entailing $a \succ_1 c$, or otherwise we have $a \approx_1 b \approx_1 c$ and $a \succsim_2 b \succsim_2 c$. The combination is reflexive because of $a\ (\succsim_1 \cap \precsim_1)\ a \succsim_2 a$. Since $\succsim = \succ_1 \cup (\approx_1 \cap \succsim_2) = (\succ_1 \cup \approx_1) \cap (\succ_1 \cup \succsim_2) = \succsim_1 \cap (\succ_1 \cup \succsim_2)$ is true, the equivalence part of $\succsim$ is just $\succsim \cap \precsim = \succsim_1 \cap (\succ_1 \cup \succsim_2) \cap \precsim_1 \cap (\prec_1 \cup \precsim_2) = \approx_1 \cap (\succ_1 \cup \succsim_2) \cap (\prec_1 \cup \precsim_2) = \approx_1 \cap \succsim_2 \cap \precsim_2 = \approx_1 \cap \approx_2$. Assume now that both quasi-orderings $\succsim_i$ are terminating. We show via induction with respect to $\succ_1$ that the existence of an infinite descending chain $a_1 \succ a_2 \succ \ldots$ leads to a contradiction: By termination of $\succ_2$, not all of the decreasing steps are with $\succ_2$. Hence, there is a minimal index i such that $a_1 \approx_1 \ldots \approx_1 a_i \succ_1 a_{i+1} \succ \ldots$ holds. Because of $a_1 \succ_1 a_{i+1}$, inductively there is no infinite descending chain from a_{i+i}. □

Remark 2.19 If a reduction system $\longrightarrow$ is terminating, then $\longrightarrow^+$ is an ordering and can be used for Noetherian induction. We obtain for example that in a terminating reduction system every element has at least one normal form.

Definition 2.20 Now we turn to reduction systems $\longrightarrow$ over the set $\mathscr{T}(\Sigma)$ of terms.

(i) We say that $\longrightarrow$ is *stable under substitutions* if $s \longrightarrow t$ implies $s\sigma \longrightarrow t\sigma$ for all terms s, t and substitutions σ. Furthermore $\longrightarrow$ is dubbed a *rewrite relation* if it is stable under substitutions and under contexts (cf. Def. 2.2 (xii)), and *ground confluent* if it is confluent on ground terms.

(ii) A *reduction ordering* $\succ$ is an ordering on terms that is a terminating rewrite relation. It is called *ground total* if it is total on ground terms. For finite signatures, $\succ$ is a *simplification ordering* if it also has the *subterm property:* $s[t]_{i.p} \succ t$ for all terms $s[t]$ and positions $i.p$.

(iii) A *rewrite rule* is a pair (l, r) of terms which have the same sort, and usually written $l \to r$ or $l \Rightarrow r$. A set R of rewrite rules is called a *rewrite system.* A rewrite relation $\longrightarrow$ *contains* R if $l \longrightarrow r$ holds for all $l \to r \in R$. The smallest such relation is denoted by $\longrightarrow_R$. We say that R is terminating, confluent etc. whenever $\longrightarrow_R$ is.

(iv) A *congruence* is an equivalence relation on $\mathscr{T}(\Sigma)$ which is closed under contexts.

Definition 2.21 We define a number of relations on terms as smallest ones satisfying some characteristic property, which must be quantified universally:

(i) *subterm ordering:* $s[t] \geq_{\text{subt}} t$
(ii) *subsumption ordering:* $t\sigma \gtrsim_{\text{subs}} t$
(iii) *encompassment ordering:* $s[t\sigma] \gtrsim_{\text{enc}} t$
(iv) *homeomorphic embedding:* $\geq_{\text{emb}}$ has the subterm property, and is transitive and closed under contexts

Each of these relations is a terminating quasi-ordering. Further properties are as follows, where α means equality up to variable renaming:

	$\geq_{\text{subt}}$	$\gtrsim_{\text{subs}}$	$\gtrsim_{\text{enc}}$	$\geq_{\text{emb}}$
equivalence part	$\equiv$	α	α	$\equiv$
context stability	no	yes	no	yes
substitution stability	yes	no	no	yes
subterm property	yes	no	yes	yes

Interestingly, the homeomorphic embedding relation $\geq_{\text{emb}}$ coincides with the reflexive-transitive closure of the rewrite relation generated by all rewrite rules $f(x_1, \ldots, x_n) \to x_i$, where f ranges over all operators, and all sorts are identified. The relation $\geq_{\text{emb}}$ plays a key rôle in what is known as *Kruskal's theorem:*

Theorem 2.22 ([Kru60]) If $t_1, t_2, \ldots$ is an infinite sequence of ground terms over a finite signature, then $t_i \leq_{\text{emb}} t_j$ holds for some indices $i < j$.

Definition 2.23 Let M denote a set with an ordering $\succ$. The following tools will be used in the construction of reduction orderings:

(i) A *multiset over* M is a function from M to the natural numbers $\mathbb{N}$ that differs from 0 only on finitely many elements of M. The *empty multiset* $\emptyset$ satisfies $\emptyset(m) = 0$ for all $m \in M$. Multisets can be written in a set-like notation like $\{a, a, b\}$ that explicitly specifies their elements with their respective multiplicities.

(ii) For multisets A and B over M, we define *union, intersection* and *set difference* via $(A \cup B)(m) = A(m) + B(m)$, $(A \cap B)(m) = \min(A(m), B(m))$ and $(A \setminus B)(m) = \max(0, A(m) - B(m))$. Let furthermore $A \subseteq B$ if $A = A \cap B$ holds.

(iii) The *multiset extension* $\succ^{\text{mul}}$ of $\succ$ is given by $A \succ^{\text{mul}} B$ if B has a presentation $B = (A \setminus X) \cup Y$ where $\emptyset \neq X \subseteq A$ and for each $x \in X$ exists an element $y \in Y$ such that $x \succ y$ is true.

(iv) Given an equivalence relation $\sim$ on M, the *multiset extension* $\succ^{\text{mul}}_{\sim}$ *with respect to* $\sim$ is defined by $A \succ^{\text{mul}}_{\sim} B$ if A and B have presentations $A = A' \cup \{a_1, \ldots, a_n\}$ and $B = B' \cup \{b_1, \ldots, b_n\}$ such that $A' \succ^{\text{mul}} B'$ and $a_i \sim b_i$ for all i. Furthermore let $\succsim = \succ \cup \sim$.

(v) The *lexicographic extension* $\succ_{\sim}^{\text{lex}}$ *with respect to an equivalence* $\sim$ is defined on vectors of equal length via $a_1, \ldots, a_n \succ_{\sim}^{\text{lex}} b_1, \ldots, b_n$ if $a_1 \sim b_1$, …, $a_{i-1} \sim b_{i-1}$ and $a_i \succ b_i$ for some i. We speak of *the lexicographic extension* $\succ^{\text{lex}}$ if $\sim$ is the identity.

Notably the multiset extension $\succ^{\text{mul}}$ is an ordering, and terminating if and only if the ordering $\succ$ is [DM79]. The same applies to the lexicographic extension. Similarly, both extensions preserve totality.

Next, the two reduction orderings most commonly used in today's theorem provers will be introduced.

Definition 2.24 Consider a *precedence* $\succ_{\mathscr{F}}$, that is, a well-founded ordering on the set $\mathscr{F}$ of function symbols, and assume that each function symbol f has a unique *status* $\iota(f)$ which is either lex(icographic) or mul(tiset). These statuses induce a *permutation congruence* $\sim$ as smallest congruence that identifies $f(s_1, \ldots, s_n)$ and $f(s_{\pi(1)}, \ldots, s_{\pi(n)})$ for any permutation π provided $\iota(f) = \text{mul}$. The *recursive path ordering* $\succ_{\text{rpos}}$ *with status* [Der79, KL80] is defined on $\mathscr{T}(\Sigma)$ as follows:

(i) $s \equiv f(\vec{s}) \succ_{\text{rpos}} g(\vec{t}) \equiv t$ if one of the following holds:
- $s_i \ (\succ_{\text{rpos}} \cup \sim) \ t$ for some i
- $f \succ_{\mathscr{F}} g$ and $s \succ_{\text{rpos}} t_j$ for all j
- $f = g$, $\iota(f) = \text{mul}$, and $\{\vec{s}\} \ (\succ_{\text{rpos}})_{\sim}^{\text{mul}} \ \{\vec{t}\}$
- $f = g$, $\iota(f) = \text{lex}$, $\vec{s} \ (\succ_{\text{rpos}})_{\sim}^{\text{lex}} \ \vec{t}$, and $s \succ_{\text{rpos}} t_j$ for all j

(ii) $s \succ_{\text{rpos}} x$ if $s \not\equiv x$ and $x \in \text{var}(s)$

The *lexicographic path ordering* $\succ_{\text{lpo}}$ is obtained if every symbol has lexicographic status, and the *recursive path ordering* $\succ_{\text{rpo}}$ if there are only symbols with multiset status.

Definition 2.25 Given a precedence $\succ_{\mathscr{F}}$, a *weight function* φ is a mapping from $\mathscr{F} \cup \mathscr{V}$ to the natural numbers such that φ returns a positive constant μ on variables, $\varphi(c) \geq \mu$ on nullary function symbols, and $\varphi(f) = 0$ for unary symbols only in case they are $\succ_{\mathscr{F}}$-greatest. It is extended to terms via $\varphi(f(\vec{t})) = \varphi(f) + \Sigma_i \varphi(t_i)$. The *Knuth-Bendix ordering* $\succ_{\text{kbo}}$ [KB70] is defined on $\mathscr{T}(\Sigma)$ as follows:

(i) $s \equiv f(\vec{s}) \succ_{\text{kbo}} g(\vec{t}) \equiv t$ if $|s|_x \geq |t|_x$ for all $x \in \mathscr{V}$,
and one of the following holds:
- $\varphi(s) > \varphi(t)$
- $\varphi(s) = \varphi(t)$ and $f \succ_{\mathscr{F}} g$
- $\varphi(s) = \varphi(t)$, $f = g$ and $\vec{s} \succ_{\text{kbo}}^{\text{lex}} \vec{t}$

(ii) $s \succ_{\text{kbo}} x$ if $s \not\equiv x$ and $x \in \text{var}(s)$

Both orderings are simplification orderings and can be employed for infinite signatures [MZ94]. If the precedence $\succ_{\mathscr{F}}$ is total, they are total on ground terms, the recursive path ordering with status however only up to the permutation congruence $\sim$.

It is customary to have a reduction ordering deal not only with terms, but also with predicative atoms, by interpreting predicate symbols as function symbols that map to a fresh sort.

Definition 2.26 A given reduction ordering $\succ$ on terms (and predicative atoms, if present) is successively extended to literals, clauses and clause multisets as follows:

(i) To every literal, one assigns a term multiset called its *complexity* via $s \simeq t \mapsto \{s, t\}$, $s \not\simeq t \mapsto \{s, s, t, t\}$, $P(\vec{t}) \mapsto \{P(\vec{t})\}$ and $\neg P(\vec{t}) \mapsto \{P(\vec{t}), P(\vec{t})\}$.

(ii) The *ordering* $\succ$ *on literals* is defined by comparing the complexities of the literals in the multiset extension of $\succ$.

(iii) The *clause ordering* $\succ$ takes the multisets of the literal complexities and compares them in the two-fold multiset extension of $\succ$.

(iv) The *clause multiset ordering* $\succ$ is the multiset extension of the clause ordering $\succ$.

By construction, the clause ordering $\succ$ is total on ground clauses if the reduction ordering is total on ground terms and atoms.

2.5 Calculi and Derivations

Clausal calculi will be described by rule patterns of three different types in a fraction-like notation. Clauses occurring in the numerator are generally called *premises*, and in the denominator *conclusions*. As usually, premises are assumed to be variable-disjoint. Finite clause sequences $C_1, \ldots, C_m$ where $m \geq 0$ are abbreviated as $\vec{C}$. If C denotes a clause and M a clause set, then M, C is shorthand notation for $M \cup \{C\}$.

(i) *Inference rules:* $\mathcal{I}\,\dfrac{\vec{C}}{D}$ if *condition*

denotes any transition from a clause set $M, \vec{C}$ to $M, \vec{C}, D$ provided *condition* is fulfilled. Occasionally the rightmost of the premises is named *main premise*, and the remaining ones are the *side premises*.

(ii) *Reduction rules:* $\mathcal{R}\,\dfrac{C}{\vec{D}}\,N$ if *condition*

stands for any transition from a clause set M, C, N to a clause set $M, \vec{D}, N$ whenever *condition* holds. In essence, the clause C is replaced by the clauses $\vec{D}$, the sequence of which may be empty.

(iii) *Split rules:* $\mathcal{S}\,\dfrac{C}{D \mid D'}$ if *condition*

describes any transition from a clause set M, C to the pair of clause sets $(M, C, D \mid M, C, D')$ constrained by *condition*. Note that the premise is part of each of the descending clause sets.

In the *condition* part of inference rules, frequently some terms, say s and t, are required to have a most general unifier σ; we stipulate that σ satisfies $\operatorname{dom}\sigma \cup \operatorname{cdom}\sigma \subseteq \operatorname{var}(s,t)$. Furthermore, occurrences of terms or of literals may be restricted to maximal ones. In the former case this maximality shall refer to the enclosing literal, and in the latter to the enclosing clause. Maximality means that no other occurrence is greater, and is strict if none is greater or equal. Correspondingly we will speak of greatest occurrences, which are greater than or equal to the remaining ones, and of strictly greater ones, that are greater than all the rest. There is no difference between being greatest or maximal in case the underlying ordering is total, as happens in the case of ground clauses and a reduction ordering total on ground terms. An application of one of the above rules is called an *inference,* a *reduction* or a *split*, respectively. Given an inference with premises $\vec{C}$ and conclusion D, then an *instance* of this inference is every inference with premises $\vec{C}\sigma$ and conclusion $D\sigma$.

A *derivation* from a (not necessarily finite) clause set M with respect to a calculus specified that way is a finitely branching tree such that (i) the nodes are sets of clauses, (ii) the root is M, and (iii) if a node N has the immediate descendants $N_1, \ldots, N_k$, respectively, then there is a transition from N to $N_1, \ldots, N_k$ in the calculus. The derivation tree degenerates into a sequence in case there is no split rule. If N and N_i are known and the transition is via an inference or a split, then we occasionally write $\vec{C} \vdash D$ to indicate the premises $\vec{C}$ from N and the conclusion D which is added to N_i. A *complete path* $N_1, N_2, \ldots$ in a derivation tree starts from the root, ends in a leaf in case the path is finite, and has the *limit* $N_\infty = \bigcup_i \bigcap_{j \geq i} N_j$. Clauses in N_∞ are called *persistent*, and such in $(\bigcup_i N_i) \setminus N_\infty$ *non-persistent.* Given a redundancy notion for inferences and clauses, a derivation is said to be *fair* if for every complete path $N_1, N_2, \ldots$ the following applies to the transitions from N_∞: (i) Every inference is redundant in some N_i, and (ii) in case splitting is mandatory in the calculus, then for every split, one of its conclusions is in some N_i or redundant with respect to it. A clause set M is *saturated* if it satisfies conditions (i) and (ii) with N_i replaced by M.

3 Superposition modulo a Shostak Theory

3.1 Introduction

Deduction in the combination of built-in and free theories has been an important topic of research for more than twenty years. One main motivation stems from program analysis, where one has to reason about standard theories like numbers as well as about free function symbols that occur naturally when abstracting over subroutines.

Deduction methods for combined theories can be roughly split into two groups: On the one hand there are white-box approaches, where theory knowledge is tightly integrated into individual inference rules of the general deductive system. This has been investigated for numerous algebraic theories, such as AC ([Plo72], among others), AC1 (e. g., [JM92]), cancellative Abelian monoids [Wal02], or Abelian groups [GN00]. While this kind of combination offers great flexibility, the requirements on termination orderings, the mathematical effort to prove the completeness of any single combined system and the technical effort to implement it are considerable. On the other hand, in black-box approaches a theory module is linked in a modular fashion to a general deductive system and the interaction between both is limited to a relatively small interface. Examples include theory resolution [Sti85], constraint resolution [Bür90], hierarchic superposition [BGW94], and the combination procedures of Nelson and Oppen [NO79] and Shostak [Sho84].

Shostak introduced a congruence closure procedure that can be combined with decision procedures for other theories, provided that these theories have canonizers and solvers. A solver is essentially a unitary unification algorithm, that transforms an equation $s \simeq t$ either into an equivalent set of equations with variables on the left-hand side, or into $\bot$ (if the equation is unsatis-

fiable). A canonizer is a procedure that transforms every term into some normal form with respect to the given theory. Here we will use these main ingredients of Shostak's method to integrate a canonizable and solvable theory into a refutationally complete theorem proving calculus for equational first-order clauses: the superposition calculus of Bachmair and Ganzinger [BG94].

From the Shostak point of view, our calculus can be seen as an extension from a calculus for (dis-)equations to a calculus working on arbitrary clauses over mixed terms. From the superposition point of view, the canonizer and the solver become simplification devices that allow us to replace theory equality by syntactic equality and to deal efficiently with overlaps between theory axioms and other axioms. The effect is thus similar to the symmetrization technique used in theory completion and theory superposition calculi [Che86, Mar94, Stu00]. However, while symmetrization tries to make coherence pairs between theory axioms and other axioms convergent by adding additional rules, the canonizer and the solver transform equations in such a form that coherence pairs become trivial.

In order to capture the behaviour of defined functions in applications like program verification, it is of course desirable that a calculus is not limited to ground formulas (corresponding to the universal fragment), but works also on non-ground formulas. In order to keep the technical contents manageable, we restrict ourselves mainly to the ground case here. Still, extending the calculus to non-ground clauses is possible to some extent, at least if variables occur only below free function symbols. This is considerably simplified by the fact that splitting disjunctions into individual formulas in a tableau-like manner is allowed (and for ground literals often useful), but not required, so that inferences between formulas remain local. Variables occurring below theory symbols are critical, however. Here, the limits of a pure black-box approach are reached, and further information on the internals of the canonizer and the solver is required in order to keep the inferences finitely branching.

3.2 Preliminaries

In this chapter we consider two signatures Δ and Φ over a single, common sort, but with disjoint function symbols, which we call *theory function symbols* and *free function symbols*, respectively. We assume that equality is the only predicate symbol and that a common set of variables is used. Terms over Δ are called *theory terms*. Every term u can be written as $u[s_1, \ldots, s_m]_{p_1,\ldots,p_m} \equiv u[\vec{s}]_{\vec{p}}$ where the s_i are the maximal non-theory subterms, i. e., every s_i has a free top symbol and any of its superterms has

a theory top symbol. Note that $\vec{s}$ may be empty, and $\vec{p}$ may be the root position. This presentation is unique up to arrangement. If u has a theory top symbol, then we use the notation $u[\![\vec{s}]\!]_{\vec{p}}$ in that situation. As usually, positions will be omitted whenever they are not relevant. A term headed by a free symbol will be called a Φ-*top term* and denoted as $f(\vec{s})$. Correspondingly $u[\![\vec{s}]\!]$ is called a Δ-*top term*.

The semantics of the theory symbols is given by a Δ-theory $\mathcal{T}$ which in this chapter we identify with a class of Δ-algebras that is closed under isomorphisms. We assume that $\mathcal{T}$ is *convex* (cf. Def. 2.10 (ii)) and contains all its σ-*models* $\sigma(\mathcal{T})$ in the sense of [RS01], that is, all Δ-structures $\mathcal{M}$ such that (i) $\mathcal{M} \models \forall X.\, s \simeq t$ whenever $s =_{\mathcal{T}} t$, and (ii) $\mathcal{M} \models s \not\simeq t$ for $s \neq_{\mathcal{T}} t$ where s and t are ground. The theory models are considered in contexts where additional free functions from Φ exist. To that end, by $\mathcal{T}^{\Phi}$ we denote the class of those $\Delta \cup \Phi$-structures the restriction of which to Δ is in $\mathcal{T}$. As an abbreviation for $\mathcal{T} \models \forall X.\, s \simeq t$, where all the free variables of s and t are bound, we simply write $s =_{\mathcal{T}} t$. Correspondingly $s \neq_{\mathcal{T}} t$ denotes $\mathcal{T} \models \forall X.\, s \not\simeq t$.

3.3 Basic Components

3.3.1 The Canonizer and its Extension

Every Shostak theory $\mathcal{T}$ comes with a so-called canonizer σ, a special simplification device that decides the word problem of $\mathcal{T}$.[1] Following [KC03], we briefly recall the formalization of this concept and describe how it can be used to decide the word problem for the extension of $\mathcal{T}$ with free function symbols.

Assumption 3.1 The canonizer σ fulfills the following properties:

(i) soundness: $t =_{\mathcal{T}} \sigma(t)$
(ii) completeness: $s =_{\mathcal{T}} t$ implies $\sigma(s) \equiv \sigma(t)$
(iii) idempotence: $\sigma^2(t) \equiv \sigma(t)$
(iv) subcanonicity: $s[t] \equiv \sigma(s[t]) \Rightarrow t \equiv \sigma(t)$
(v) variable conditions: $\sigma(x) \equiv x$, $\mathrm{var}(u) \supseteq \mathrm{var}(\sigma(u))$, and $\sigma(t\pi) \equiv \sigma(t)\pi$ for any renaming π of variables that preserves a fixed total ordering of the set of variables

A term t is called *canonical* if $t \equiv \sigma(t)$. The ordering condition imposed in (v) is less restrictive than full invariance under variable renamings. For

[1] Unfortunately this naming collides with the usage of σ as identifier of a substitution.

example, it still allows to canonize $y + x$ to $x + y$, provided that, say, $y \succ x$. This weaker form of invariance is required in [KC03] in order to get the extension of σ to a canonizer $\bar{\sigma}$ for terms over the combined signature well-defined.

The usual proceeding is, given some Δ-top term $u[\![\vec{s}]\!]$, to: (i) recursively canonize $\vec{s}$, say such that $u[\![\bar{\sigma}(\vec{s})]\!] \equiv u[\![\vec{t}]\!]$, (ii) build an *abstraction* $u[\vec{x}]$ of $u[\![\vec{t}]\!]$ where the maximal non-theory terms are replaced by variables, (iii) canonize the theory term $u[\vec{x}]$, and (iv) reinsert the Φ-top terms t_i for the x_i in the result. In step (ii) the choice of variables may matter, because the canonizer need not be stable under arbitrary renamings of the variables. So an order-preserving bijection between terms and variables has to be given. We choose a simpler presentation: When dealing with theory terms, we directly consider Φ-top terms $f(\vec{u})$ as variables. We name them *canonizer variables*, as opposed to *schematic variables*. The notation allows formulations like this:

Proposition 3.2 For theory terms s and t, that may contain canonizer variables, $s =_{\mathcal{T}} t$ implies $s =_{\mathcal{T}^\Phi} t$. Furthermore $s \neq_{\mathcal{T}} t$ entails $s \neq_{\mathcal{T}^\Phi} t$.

Proof: Let $I^\Phi \in \mathcal{T}^\Phi$, μ an assignment of the schematic variables of s and t, and μ' the extension of μ to the canonizer variables at hand such that $\mu'(f(\vec{u})) = I^\Phi_\mu(f(\vec{u}))$. Then $I^\Phi_\mu = I_{\mu'}$ on s and t. □

Now the extension of σ to terms over the combined signature can easily be expressed in an innermost fashion, as for example in [Kap02] or [SR02]:

Definition 3.3 $\bar{\sigma}(u[\![\vec{s}]\!]) :\equiv \sigma(u[\bar{\sigma}(\vec{s})])$, $\bar{\sigma}(f(\vec{u})) :\equiv f(\bar{\sigma}(\vec{u}))$, and $\bar{\sigma}(x) :\equiv \sigma(x)$.

Consider as an example the theory of linear rational arithmetic. A simple canonizer for $\mathcal{T}$ is obtained by increasingly arranging addends according to some fixed order on all variables and combining like terms. To compute $\bar{\sigma}(f(c) + 3g(1) - 5f(c + 0))$, the canonizer variables $f(c)$, $g(1)$ and $f(c + 0)$ are extracted and recursively canonized to $f(c)$, $g(1)$ and $f(c)$, respectively. The final result is $\sigma(f(c) + 3g(1) - 5f(c)) \equiv 3g(1) - 4f(c)$ if, say, $f(c)$ is greater than $g(1)$.

The above definition is well-formed in that (i) $\bar{\sigma}$ is terminating, for example by the multiset extension of the subterm ordering; (ii) it is total; and (iii) σ is applied to theory terms only. Evidently, $\bar{\sigma}$ on theory terms is just σ. Soundness of the extension is easy to see. But completeness need not hold for non-convex theories: In the signature $\Delta = \{1/0, 2/0\}$, let $\mathcal{T}$ denote the class of Δ-algebras that satisfy the formulae $\forall x.\, x \simeq 1 \vee x \simeq 2$ and $1 \not\simeq 2$. The signature contains only constants. Up to variable renamings, only finitely

many equations can be constructed: the syntactic tautologies $x \simeq x$, $1 \simeq 1$, $2 \simeq 2$ and the non-theorems $x \simeq y$, $x \simeq 1$, $x \simeq 2$ and $1 \simeq 2$. Hence the identity serves as canonizer for the structure at hand. With $\Phi = \{f/1\}$ we then have $f^3(x) =_{\mathcal{T}^\Phi} f(x)$ because on a domain of size 2 there are just four unary functions: a negation-like, two constant ones, and the identity, and each of these satisfies $f^3 = f$. However, left- and right-hand side are canonical. As shown in [KC03], the incompleteness vanishes if convexity is given:

Theorem 3.4 The extension of σ to $\bar{\sigma}$ preserves soundness, completeness, and computability for convex $\mathcal{T}$.

3.3.2 The Black-Box Path Ordering

In the sequel, we will need an ordering on mixed ground terms that comes as close as possible to a ground total reduction ordering, is compatible with canonizer applications, and has the so-called multiset property. We will define such an ordering in three steps: First, we define an auxiliary signature Φ^κ, whose symbols are, one the one hand, the free symbols in Φ, and on the other hand, the contexts over Δ. Then we define a variant of the lexicographic path ordering for terms over Φ^κ and prove its essential properties. Finally, we define a mapping from terms over $\Phi \cup \Delta$ to terms over Φ^κ and obtain an ordering for terms over $\Phi \cup \Delta$ via this mapping.

We assume a fixed set of canonizer variables $CV_{\text{norm}} = \{ c_i \mid i \in \mathbb{N} \}$ with the ordering $c_0 \prec_{\text{norm}} c_1 \prec_{\text{norm}} \ldots$ A ground term u over theory symbols and CV_{norm} is CV_{norm}-*normalized* if the set of canonizer variables in u equals $\{ c_i \mid i < k \}$ for some $k \in \mathbb{N}$. Let Φ^κ be the signature that contains all function symbols from Φ and a new symbol κ_u for every CV_{norm}-normalized term u. Function symbols from Φ keep their arities; the arity of any κ_u is the number of occurrences of canonizer variables in u.

A well-founded total ordering $\succ_{\text{cont}}$ on Φ^κ is called a σ-*context precedence* if it satisfies the following conditions for all function symbols $f \in \Phi$ and CV_{norm}-normalized terms u and v:

(a) $f \succ_{\text{cont}} \kappa_u$,

(b) if u and v contain identical multisets of canonizer variables, and if the number of non-canonical subterms in u is greater than the number of non-canonical subterms in v, then $\kappa_u \succ_{\text{cont}} \kappa_v$,

(c) if v is a proper subterm of u, then $\kappa_u \succ_{\text{cont}} \kappa_v$.

It is easy to construct a σ-context precedence for an arbitrary canonizer σ, for instance by comparing first the origin of the symbols (Φ vs. $\Phi^\kappa \setminus \Phi$), then, for κ_u, the number of non-canonical subterms in u, followed by the size of u,

and finally by comparing the symbols according to some arbitrarily chosen total and well-founded ordering.

Let $\sim_\kappa$ be the congruence that is induced by the permutativity axioms $\kappa_u(x_1, \ldots, x_m) \simeq \kappa_u(x_{\pi(1)}, \ldots, x_{\pi(m)})$ for all κ_u.

The auxiliary ordering $\succ_\kappa$ on ground Φ^κ-terms induced by the σ-context precedence $\succ_{\text{cont}}$ is defined as follows: $s \succ_\kappa t$ if and only if

(a) $s \equiv h(s_1, \ldots, s_m)$, $h \in \Phi^\kappa$, and $s_i \succsim_\kappa t$ for some $i \in \{1, \ldots, m\}$, or
(b) $s \equiv f(s_1, \ldots, s_m)$, $f \in \Phi$, $t \equiv h(t_1, \ldots, t_n)$, $h \in \Phi^\kappa$ with $f \succ_{\text{cont}} h$ and $s \succ_\kappa t_j$ for all $j \in \{1, \ldots, n\}$, or
(c) $s \equiv f(s_1, \ldots, s_m)$, $t \equiv f(t_1, \ldots, t_m)$, $f \in \Phi$, $s \succ_\kappa t_j$ for all $j \in \{1, \ldots, n\}$, and $s_1, \ldots, s_m$ $(\succ_\kappa)^{\text{lex}}_{\sim_\kappa}$ $t_1, \ldots, t_m$, or
(d) $s \equiv \kappa_u(s_1, \ldots, s_m)$, $t \equiv \kappa_v(t_1, \ldots, t_n)$, and $\{s_1, \ldots, s_m\}$ $(\succ_\kappa)^{\text{mul}}_{\sim_\kappa}$ $\{t_1, \ldots, t_n\}$, or
(e) $s \equiv \kappa_u(s_1, \ldots, s_m)$, $t \equiv \kappa_v(t_1, \ldots, t_m)$, the multisets $\{s_1, \ldots, s_m\}$ and $\{t_1, \ldots, t_m\}$ agree up to $\sim_\kappa$, and $\kappa_u \succ_{\text{cont}} \kappa_v$.

The relation $\succ_\kappa$ differs from a traditional recursive path ordering with status only in that it compares two terms with top symbols in $\Phi^\kappa \setminus \Phi$ by comparing lexicographically first the multisets of arguments and then the two top symbols. One can show in essentially the same way as for the lexicographic path ordering that $\succ_\kappa$ is a simplification ordering on ground Φ^κ-terms which is compatible with $\sim_\kappa$ and total and irreflexive up to $\sim_\kappa$.

Lemma 3.5 If $\succ_{\text{cont}}$ is a σ-context precedence, then $\succ_\kappa$ has the following properties for all ground Φ^κ-terms r, s, t, $s_1, \ldots, s_m$:

(i) Transitivity and compatibility with $\sim_\kappa$: $r \succsim_\kappa s$ and $s \succ_\kappa t$ imply $r \succ_\kappa t$; $r \succ_\kappa s$ and $s \succsim_\kappa t$ imply $r \succ_\kappa t$.
(ii) Subterm property: If t is a proper subterm of s, then $s \succ_\kappa t$.
(iii) Compatibility with contexts: $s \succ_\kappa t$ implies $h(\ldots, s, \ldots) \succ_\kappa h(\ldots, t, \ldots)$ for every $h \in \Phi^\kappa$.
(iv) Irreflexivity: $s \sim_\kappa t$ implies $s \not\succ_\kappa t$.
(v) $h_1(s_1, \ldots, s_m) \succ_\kappa h_2(s_{i_1}, \ldots, s_{i_n})$ if $m \geq n$, $h_1 \succ_{\text{cont}} h_2$, and either $1 \leq i_1 < \ldots < i_n \leq m$ or $m = n = 0$.
(vi) $\succ_\kappa$ is a simplification ordering[2] on ground terms and therefore well-founded.
(vii) Totality: $s \succ_\kappa t$ or $s \prec_\kappa t$ or $s \sim_\kappa t$.

Proof: The proof proceeds in essentially the same way as for the lexicographic path ordering (cf. [BN98, Chap. 5.4.2]). Part (i) is proved by induction on $|r| + |s| + |t|$ and a somewhat lengthy case analysis on the rules that are used to show $r \succ_\kappa s$ and/or $s \succ_\kappa t$. Property (ii) follows from (a) and

[2]In the sense of Middeldorp and Zantema [MZ94] for possibly infinite signatures.

transitivity, property (iii) from (c) for $h \in \Phi$ and from (d) for $h \in \Phi^\kappa \setminus \Phi$. Part (iv) is proved by induction on $|s|$, and (v) follows from (b) for $h_1 \in \Phi$ and from (d) and (e) for $h_1 \in \Phi^\kappa \setminus \Phi$. Now (vi) is an immediate consequence of (i)–(v). Note that the fact that we consider only ground terms does not influence termination. Property (vii) is again proved by induction on $|s|+|t|$. □

Let $\succ$ be an ordering on terms, and let $s \equiv u[\![s_1, \ldots, s_n]\!]$ be a term such that $\succ$ is total on the set $\{s_1, \ldots, s_n\}$. Then the *CV_{norm}-normalized form of* s is the term $\mathrm{norm}(s, \succ)$ that is obtained from s by replacing the smallest term of $\{s_1, \ldots, s_n\}$ by c_0, the second smallest by c_1, and so on. (Multiple occurrences of the same term are replaced by the same canonizer variable from CV_{norm}.)

We can now define a binary relation $\succ_{\mathrm{bb}}$ on ground terms over $\Phi \cup \Delta$ and a mapping κ from ground terms over $\Phi \cup \Delta$ to ground terms over Φ^κ by mutual recursion:

(a) $s \succ_{\mathrm{bb}} t$ if and only if $\kappa(s)$ and $\kappa(t)$ are defined and $\kappa(s) \succ_\kappa \kappa(t)$.
(b) If $s \equiv f(s_1, \ldots, s_m)$, $f \in \Phi$, and $\kappa(s_i)$ is defined for all $i \in \{1, \ldots, m\}$, then $\kappa(s) \equiv f(\kappa(s_1), \ldots, \kappa(s_m))$.
(c) If $s \equiv u[\![s_1, \ldots, s_m]\!]$, $\kappa(s_i)$ is defined for all $i \in \{1, \ldots, m\}$, and $\succ_{\mathrm{bb}}$ is total on the set $\{s_1, \ldots, s_m\}$, then $\kappa(s) \equiv \kappa_{\mathrm{norm}(s, \succ_{\mathrm{bb}})}(\kappa(s_1), \ldots, \kappa(s_m))$.

It is easy to check that κ is injective. Furthermore, if $t \equiv \kappa(s)$ for some term s, then the order of arguments of every κ_u occurring in t corresponds to the order of context variables in u. It is thus clear that the set of all terms that are the image of some term under κ contains at most one element of any $\sim_\kappa$-congruence class. This is the key argument in proving the following theorem:

Theorem 3.6 If $\succ_{\mathrm{cont}}$ is a σ-context precedence, then the relation $\succ_{\mathrm{bb}}$ and the mapping κ have the following properties for all ground Φ^κ-terms r, s, t, $t_1, \ldots, t_n$:

(i) Transitivity: $r \succ_{\mathrm{bb}} s$ and $s \succ_{\mathrm{bb}} t$ imply $r \succ_{\mathrm{bb}} t$
(ii) Irreflexivity: $s \not\succ_{\mathrm{bb}} s$.
(iii) Well-foundedness: There exists no infinite decreasing $\succ_{\mathrm{bb}}$-chain.
(iv) Totality of $\succ_{\mathrm{bb}}$: $s \succ_{\mathrm{bb}} t$ or $s \prec_{\mathrm{bb}} t$ or $s \equiv t$.
(v) Totality of κ: $\kappa(s)$ and $\kappa(t)$ are defined.
(vi) Subterm property: If t is a proper subterm of s, then $s \succ_{\mathrm{bb}} t$.
(vii) Compatibility with free contexts: $s \succ_{\mathrm{bb}} t$ implies $f(\ldots, s, \ldots) \succ_{\mathrm{bb}} f(\ldots, t, \ldots)$ for every $f \in \Phi$.
(viii) Partial compatibility with theory contexts: $s \succ_{\mathrm{bb}} t$ implies $u[\![\ldots, s \ldots]\!] \succ_{\mathrm{bb}} u[\ldots, t, \ldots]$ if s is a Φ-top term.

(ix) Multiset property: If s, $t_1, \ldots, t_n$ are Φ-top terms, $s \succ_{\mathrm{bb}} t_i$ for every $i \in \{1, \ldots, m\}$, and u is a theory context, then $s \succ_{\mathrm{bb}} u[\![t_1, \ldots, t_n]\!]$.

Proof: Properties (i), (ii), and (iii) follow directly from the corresponding properties of $\succ_\kappa$. Properties (iv) and (v) are proved by simultaneous induction over $\max(|s|, |t|)$ using the fact that $\kappa(s') \sim_\kappa \kappa(t')$ entails $\kappa(s') \equiv \kappa(t')$ and thus $s' \equiv t'$. To prove (vi), it is sufficient to show that $s \equiv h(\ldots, t, \ldots) \succ_{\mathrm{bb}} t$ for every $h \in \Phi^\kappa$ by transitivity. For $h \in \Phi$, this is obvious, but the case $h \in \Phi^\kappa \setminus \Phi$ is more difficult: If t is a Φ-top term, then s has the form $u[\![\ldots, t, \ldots]\!]$ for some theory context u, hence $\kappa(s) \equiv \kappa_{\mathrm{norm}(s,\succ_{\mathrm{bb}})}(\ldots, \kappa(t), \ldots) \succ_\kappa \kappa(t)$. Otherwise, $t \equiv v[\![\vec{t}]\!]$ and $s \equiv h(\ldots, v[\![\vec{t}]\!], \ldots) \equiv u[\![\vec{s}]\!]$, and therefore $\kappa(s) \equiv \kappa_{\mathrm{norm}(s,\succ_{\mathrm{bb}})}(\kappa(\vec{s}))$ and $\kappa(t) \equiv \kappa_{\mathrm{norm}(t,\succ_{\mathrm{bb}})}(\kappa(\vec{t}))$. Now either $\vec{t}$ is a strict sublist of $\vec{s}$, then $\kappa(s) \succ_\kappa \kappa(t)$ by part (d) of the definition of $\succ_\kappa$, or $\vec{t} \equiv \vec{s}$, then $\kappa(s) \succ_\kappa \kappa(t)$ by part (e) of the definition of $\succ_\kappa$ and by part (c) of the definition of a σ-context precedence. Part (vii) follows again directly from the corresponding property of $\succ_\kappa$. For (viii) notice that $u[\ldots, t \ldots]$ can be written as $v[\![\ldots, \vec{t} \ldots]\!]$, where either $t \equiv \vec{t}$ or $t \equiv v'[\![\vec{t}]\!]$. In any case, $s \succ_{\mathrm{bb}} t_j$ for every t_j in $\vec{t}$, hence $u[\![\ldots, s \ldots]\!] \succ_{\mathrm{bb}} v[\![\ldots, \vec{t} \ldots]\!]$ by part (d) of the definition of $\succ_\kappa$. Finally, property (ix) follows from part (b) of the definition of $\succ_\kappa$. □

As can be seen from this collection of properties, what the *black-box path ordering* $\succ_{\mathrm{bb}}$ lacks to become a ground total reduction ordering is full compatibility with theory contexts.

In order to ensure compatibility with canonizer applications, we have to impose an additional condition:

Assumption 3.7 From now on we require that the canonizer σ is *multiset decreasing* in the sense that $\sigma(u[\![\vec{s}]\!]) \equiv v[\![\vec{t}]\!]$ implies $\{\vec{s}\}\ (\succeq_{\mathrm{bb}})^{\mathrm{mul}}\ \{\vec{t}\}$, and $\sigma(u[\![\vec{s}]\!]) \equiv t$ implies $\{\vec{s}\}\ (\succeq_{\mathrm{bb}})^{\mathrm{mul}}\ \{t\}$ if t is a Φ-top term.

At first glance, this assumption looks a bit restrictive, in that it excludes canonizers which duplicate variables; e. g. when applying distributivity to turn $x(y+z)$ into $xy+xz$. However, the canonizer is free to internally employ a fresh theory operator $f_{\lambda uvw.uv+uw}$ with the semantics $f_{\lambda uvw.uv+uw}(x, y, z) =_{\mathcal{T}} xy+xz$ to resolve this non-linearity. This is possible, because in our calculus, theory operators must be known only to the ordering, the canonizer, and the solver that will be introduced in Sect. 3.3.4. For the ordering, dealing with infinite signatures is unproblematic, and canonizer and solver can always expand the new operator on the fly whenever they encounter it, continue as usual, and finally linearize again. For the remainder of the superposition

calculus, the new theory operator is practically irrelevant: it treats theory contexts as black boxes anyway.

Theorem 3.8 If σ is multiset decreasing, then $s[u[\![\vec{t}]\!]] \succeq_{\mathrm{bb}} s[\sigma(u[\![\vec{t}]\!])]$.

Proof: By parts (vii) and (viii) of Thm. 3.6 it is sufficient to consider the case that σ is applied within a theory context at the top of s. That is, $s \equiv v[\![\vec{s}]\!] \equiv v''[\ldots, u[\![\vec{t}]\!], \ldots]$, where $\vec{t}$ is some sublist of $\vec{s}$. We have to show that $s \succeq_{\mathrm{bb}} s' \equiv v''[\ldots, \sigma(u[\![\vec{t}]\!]), \ldots] \equiv v'[\![\vec{r}]\!]$, where $\vec{r}$ is obtained from $\vec{s}$ by replacing $\vec{t}$ by the list of free subterms in $\sigma(u[\![\vec{t}]\!])$. As σ is multiset decreasing, the multiset of terms in $\vec{r}$ is smaller than or equal to the multiset of terms in $\vec{s}$. If it is strictly smaller, then $s \succ_{\mathrm{bb}} s'$ by part (d) of the definition of $\succ_\kappa$. If the two multisets are equal, then $\mathrm{norm}(s, \succ_{\mathrm{bb}})$ and $\mathrm{norm}(s', \succ_{\mathrm{bb}})$ contain identical multisets of canonizer variables. Now there are two possibilities: Either $u[\![\vec{t}]\!] \equiv \sigma(u[\![\vec{t}]\!])$, then trivially $s \equiv s'$. Or the number of non-canonical subterms in $\mathrm{norm}(s, \succ_{\mathrm{bb}})$ is strictly larger than the number of non-canonical subterms in $\mathrm{norm}(s', \succ_{\mathrm{bb}})$, hence $\kappa_{\mathrm{norm}(s,\succ_{\mathrm{bb}})} \succ_{\mathrm{cont}} \kappa_{\mathrm{norm}(s',\succ_{\mathrm{bb}})}$ by part (b) of the definition of σ-context precedence. By part (e) of the definition of $\succ_\kappa$, we obtain $s \succ_{\mathrm{bb}} s'$. □

Since $\bar{\sigma}(t)$ is obtained from t by iterated application of σ to Δ-top subterms, the following theorem is an obvious consequence:

Theorem 3.9 If σ is multiset decreasing, then $s[t] \succeq_{\mathrm{bb}} s[\bar{\sigma}(t)]$.

Multiset decreasingness is crucial here. For canonizers σ that are not multiset decreasing, there need not exist any total ordering on mixed terms that is both compatible with canonizer applications and (partially) compatible with theory contexts. As an example consider $\Delta = \{h/2\}$, $\Phi = \{a/0, b/0\}$, and a canonizer that transforms any Δ-term t into $h(s, s)$, where s is the greatest canonizer variable in t. If $\succ$ is (partially) compatible with theory contexts and, say, $a \succ b$, then $h(a, a) \succ h(a, b)$, but on the other hand, σ transforms $h(a, b)$ back into $h(a, a)$.

Similarly, it is in general not possible to extend property (viii) of Thm. 3.6 to full compatibility with theory contexts while keeping totality and compatibility with canonizer applications. As an example, consider $\Delta = \{a/0, b/0, f/1, g/1\}$ and a canonizer σ that transforms $f(a)$ into $f(b)$ and $g(b)$ into $g(a)$. If $\succ$ were total and fully compatible with theory contexts, then either $a \succ b$ or $b \succ a$, but $a \succ b$ would imply $g(a) \succ g(b)$ and similarly $b \succ a$ would imply $f(b) \succ f(a)$.

In the following $\succ$ will denote the black-box path ordering $\succ_{\mathrm{bb}}$ to some arbitrary σ-context precedence $\succ_{\mathrm{cont}}$. It is lifted to literals and clauses as usually (cf. Def. 2.26).

Since the publication of the results of this chapter in [GHW03], a related ordering construction has been presented in [FGR05].

3.3.3 Canonizing and Rewriting

By now we have extended the canonizer to terms over the combined signature; and with the black-box path ordering we have developed a tool for canonizer-related termination argumentations. In this subsection we will first of all rephrase the extended canonizer as a rewrite system, i. e., in terms of equational deduction, second employ the black-box path ordering to show that the resulting system is convergent, and third study convergence of special extensions. The rewrite system will be used as an ubiquitous simplification device and therefore later become part of the model construction.

Definition 3.10 With an extended canonizer $\bar{\sigma}$, we associate $\widehat{\sigma} = \bar{\sigma} \setminus \mathrm{id}$ as a ground rewrite system.

Hence the rewrite system $\widehat{\sigma}$ consists of rules $t \to \bar{\sigma}(t)$ where t is non-canonical. The rewrite relation $\longrightarrow_{\widehat{\sigma}}$ is just the context closure thereof, and it has a very simple structure: If $s \longrightarrow_{\widehat{\sigma}} t$ is a top-level reduction, then $\bar{\sigma}(s) \equiv t$ and $s \not\equiv t$ and t is irreducible. By soundness and completeness of the extended canonizer (Thm. 3.4) the equality $\longleftrightarrow^*_{\widehat{\sigma}}$ generated by the rewrite system coincides with $=_{\mathcal{T}^\Phi}$.

The extended canonizer $\bar{\sigma}$ actually decides the word problem for $\mathcal{T}^\Phi$, via computation and syntactical comparison of normal forms. So does the rewrite-based reformulation:

Proposition 3.11 The rewrite system $\widehat{\sigma}$ is terminating and confluent.

Proof: Concerning termination, every single rewrite step can be presented as $s[t]_p \longrightarrow_{\widehat{\sigma}} s[\bar{\sigma}(t)]_p$. Since the ordering $\succeq$ is strict, Thm. 3.9 implies that every such step is decreasing. Finally, $\succ$ is transitive and well-founded.

Because of termination, confluence coincides with local confluence. By the Critical Pair Lemma, local confluence of rewrite systems is in turn equivalent to joinability of critical pairs. So the fact that $\succ$ is not precisely a reduction ordering is not relevant here. Consider now $l[r'] \;{}_{\widehat{\sigma}}\!\longleftarrow l[l'] \longrightarrow_{\widehat{\sigma}} r$. Since the right rewrite step is top-level, r is canonical. By completeness and idempotence of $\bar{\sigma}$ we already have $\bar{\sigma}(l[r']) \equiv r$. □

Termination holds because the rewrite steps can be embedded in a black-box path ordering (Thm. 3.9) which itself is terminating (Thm. 3.6). Hence confluence is equivalent to local confluence, which reduces to joinability

of critical pairs. The latter can be derived from the completeness of $\bar{\sigma}$ (Thm. 3.4).

We now study convergence of $\widehat{\sigma}$ extended by a single rule $l \Rightarrow r$ where l is a Φ-top term. Roughly this is the shape of equations after application of the solver. Within the analysis of confluence, critical pairs from such rules $l \Rightarrow r$ into canonizer rules will occur. For arbitrary terms t, l and r, we define recursively the *parallel reduction* $t[l \mapsto r]$ as r if $t \equiv l$, as x if $t \equiv x \not\equiv l$, and as $h(\vec{s}[l \mapsto r])$ if $t \equiv h(\vec{s}) \not\equiv l$. Clearly every parallel reduction can be computed by an iterated sequential one. The following, somewhat technical proposition states that under specific requirements canonization and parallel reduction commute, and will be an important ingredient to the subsequent confluence proof.

Proposition 3.12 Canonization and parallel reduction are related as follows:

(i) For terms over Δ, $v[x \mapsto u] =_{\mathcal{T}} \sigma(v)[x \mapsto u]$.

(ii) For terms over Δ and Φ, $s[l \mapsto r] =_{\mathcal{T}^\Phi} \bar{\sigma}(s)[l \mapsto r]$ in case that l is a canonical Φ-top term, and the strict subterms of s are canonical.

Proof: (i) Soundness of σ means $v =_{\mathcal{T}} \sigma(v)$. Since variables are universally quantified, we have $v[x \mapsto u] =_{\mathcal{T}} \sigma(v)[x \mapsto u]$.

(ii) is shown via reduction to (i). If $s \equiv l$, then $s \equiv \bar{\sigma}(s)$, by canonicity of l. Otherwise, in case $s \equiv x$, we have $\bar{\sigma}(x)[l \mapsto r] \equiv \sigma(x)[l \mapsto r] \equiv x[l \mapsto r]$. For $s \equiv f(\vec{u})$ holds $\bar{\sigma}(s) \equiv \bar{\sigma}(f(\vec{u})) \equiv f(\bar{\sigma}(\vec{u})) \equiv f(\vec{u}) \equiv s$ because the subterms of s are canonical. Finally consider $s \equiv v[\![\vec{t}]\!]$. By the subterm canonicity $\bar{\sigma}(s) \equiv \sigma(v[\bar{\sigma}(\vec{t})]) \equiv \sigma(v[\vec{t}]) \equiv \sigma(s)$. Now we have by Prop. 3.12 (i) for theory terms that $s[l \mapsto r] =_{\mathcal{T}} \sigma(s)[l \mapsto r]$ because the top symbol of l is free. By Prop. 3.2, this implies $s[l \mapsto r] =_{\mathcal{T}^\Phi} \sigma(s)[l \mapsto r]$. □

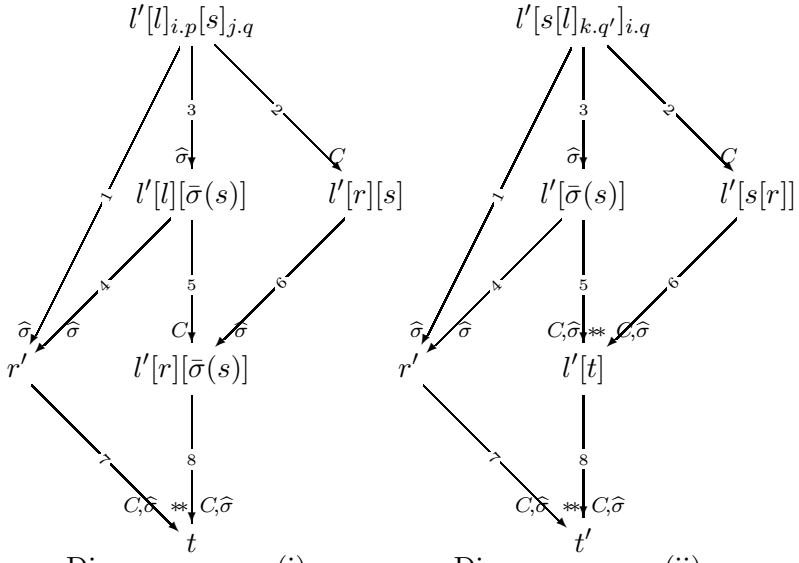
The first statement is a consequence of the soundness of σ. The second one relies on the subcanonicity of σ and $\bar{\sigma}$, and follows from the first one by Prop. 3.2. It is an important ingredient to the proof of the following confluence result. This theorem is at the heart of our approach; we therefore give a detailed proof.

Theorem 3.13 For every ground rewrite rule $l \Rightarrow r$ where l is a canonical Φ-top term and $l \succ r$, the combined system $\widehat{\sigma} \cup \{l \Rightarrow r\}$ is terminating and confluent.

Proof: Because of Thm. 3.6 we have that $t[l] \succ t[r]$ for every term t; therefore $\widehat{\sigma} \cup \{l \Rightarrow r\}$ is terminating. Hence the system is confluent if all the critical pairs are joinable. We prove the latter by induction on the peak term with respect to $\succ$.

The critical pairs of $\widehat{\sigma}$ are joinable by Prop. 3.11. The rule $l \Rightarrow r$ is ground and overlaps with itself trivially only. So we only need to study critical pairs between $l \Rightarrow r$ and some $l' \Rightarrow r' \in \widehat{\sigma}$. For the sake of brevity, let $\longrightarrow_C = \longrightarrow_{\{l \Rightarrow r\}}$ and $\longrightarrow_{C,\widehat{\sigma}} = \longrightarrow_{\{l \Rightarrow r\} \cup \widehat{\sigma}}$.

Canonizer steps below rewrite steps like in $r \;{}_C\!\longleftarrow l[l'] \longrightarrow_{\widehat{\sigma}} l[r']$ are not possible because $l[l']$ is canonical by assumption and hence also the subterm l' is. Now let us analyze vice versa rewrite steps below canonizer steps, i. e., $r' \;{}_{\widehat{\sigma}}\!\longleftarrow l'[l]_{i.p} \longrightarrow_C l'[r]$. Since the subterms of l' need not be canonical, we distinguish the following cases:

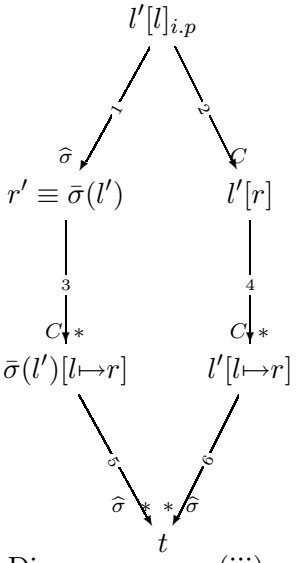


Divergence case (i) Divergence case (ii)

(i) l' might contain a $\longrightarrow_{\widehat{\sigma}}$-redex at a position $j.q$ parallel to $i.p$. The divergency corresponds to the reduction steps numbered 1 and 2 in the diagram, whereas the $\longrightarrow_{\widehat{\sigma}}$-redex leads to the one numbered 3. Step 4 exists because $l'[l][s]$, r' and $l'[l][\bar{\sigma}(s)]$ are $\mathcal{T}^{\Phi}$-equal and r' is canonical. Steps 5 and 6 are possible because reductions 2 and 3 have been parallel. Steps 7 and 8 must exist inductively because reduction 3 is decreasing.

(ii) Since l is canonical, l' has no $\longrightarrow_{\widehat{\sigma}}$-redex below $i.p$, but there may exist one strictly above, say at $i.q$ where $p = q.k.q'$. This corresponds to reduction 3 in the diagram. As above, Step 4 exists because $l'[s[l]]$, r' and $l'[\bar{\sigma}(s[l])]$ are $\mathcal{T}^{\Phi}$-equal and r' is canonical. Steps 5 and 6 are justified inductively because s is a strict subterm of l', such that $l' \succ s$ holds. Steps 7 and 8 are possible because reduction 3 is decreasing.

(iii) Finally consider l' without any $\longrightarrow_{\widehat{\sigma}}$-redex but on top-level. Left-hand and right-hand side of the divergency can sequentially be reduced by $\longrightarrow_C$ to their respective images under the parallel reduction $[l \mapsto r]$. By Prop. 3.12, these are $\mathcal{T}^{\Phi}$-equal and hence have the same canonical form.



Divergence case (iii)

□

3.3.4 The Solver, and its Extension and Application

Besides the canonizer, the solver is the second procedural component of Shostak's method, basically a unitary unification algorithm. From a conceptual point of view, for Shostak's method itself the solver is even of higher importance, because there the canonizer can be seen as a facultative simplification device. The solver is characterized as follows (cf. [Gan02]):

Assumption 3.14 For any equation $s \simeq t$ over Δ, the solver returns

$$\mathsf{solve}(s \simeq t) = \begin{cases} \{\} & s =_{\mathcal{T}} t \\ \bot & s \neq_{\mathcal{T}} t \\ \{\vec{x} \simeq \vec{u}\} & \text{otherwise} \end{cases}$$

where in the last case the following conditions additionally hold:

(i) $\{\vec{x} \simeq \vec{u}\} \neq \{\}$
(ii) $x_i \not\equiv x_j$ for $i \neq j$
(iii) $\{\vec{x}\} \subseteq \mathrm{var}(s \simeq t)$
(iv) $\{\vec{x}\} \cap \mathrm{var}(\vec{u}) = \{\}$
(v) $\mathcal{T} \models \forall X.\, (s \simeq t \leftrightarrow \exists Y.\, \vec{x} \simeq \vec{u})$ with $Y = \mathrm{var}(\vec{u}) \setminus \mathrm{var}(\vec{s} \simeq \vec{t})$ and $X \cap Y = \{\}$

Condition (iv) corresponds to the characterization of idempotence for substitutions. Note that solutions can be parameterized by new variables, namely those that occur freshly in $\vec{u}$. Consequently these are existentially quantified in condition (v). For example, in the case of linear arithmetic over the integers, calling $\mathsf{solve}(3x_1 + 5x_2 \simeq 1)$ returns the solution $\{x_1 \simeq -3 + 5y,\, x_2 \simeq 2 - 3y\}$.

We want to employ this special unification algorithm to solve equations over the combined signature with respect to one (or more) of their theory subterms. To this end, we extend it like the canonizer by considering all maximal Φ-top terms as canonizer variables.

Definition 3.15 For equations $s \simeq t$ over the combined signature $\Delta \cup \Phi$, we define $\overline{\mathsf{solve}}(s \simeq t) := \mathsf{solve}(s \simeq t)$, where $s \simeq t$ on the right-hand side is read as an equation over Δ.

In our example of linear arithmetic, let us consider for instance a call of $\overline{\mathsf{solve}}$ on the equation $f(c) + 3g(1) - 5f(c+0) \simeq 0$. The canonizer variables in this equation are $f(c)$, $g(1)$ and $f(c+0)$. So the extended solver can for example return the equation $f(c+0) \simeq \frac{1}{5}f(c) + \frac{3}{5}g(1)$.

With arguments similar to those required in the proof of Prop. 3.2, we get the following property which shows that the extension serves the intended purpose:

Proposition 3.16 The extended solver is sound in the sense that
$\overline{\mathsf{solve}}(s \simeq t) = \{\}$ implies $s =_{\mathcal{T}^\Phi} t$,
$\overline{\mathsf{solve}}(s \simeq t) = \bot$ implies $s \neq_{\mathcal{T}^\Phi} t$, and
$\overline{\mathsf{solve}}(s \simeq t) = \{\vec{x} \simeq \vec{u}\}$ implies $\mathcal{T}^\Phi \models \forall X_0.\, (s \simeq t \leftrightarrow \exists Y.\, \vec{x} \simeq \vec{u})$,
where X_0 is the set of schematic variables of X of Ass. 3.14.

Proof: The first two cases are directly covered by Prop. 3.2. For the third, we use as in the proof of Prop. 3.2 that, given an interpretation $I^\Phi \in \mathcal{T}^\Phi$ and an assignment μ of schematic variables, there exists an extension μ' of μ to canonizer variables such that $I^\Phi_\mu = I_{\mu'}$ on the terms at hand. As to the left-to-right implication of the equivalence, consider I^Φ_μ with $I^\Phi_\mu(s) = I^\Phi_\mu(t)$, and an extension μ' as before. Then by Ass. 3.14 (v) there is an assignment ν of the variables Y such that $I_{\mu' \cup \nu}(\vec{x}) = I_{\mu' \cup \nu}(\vec{u})$, and hence also $I^\Phi_{\mu \cup \nu}(\vec{x}) = I^\Phi_{\mu \cup \nu}(\vec{u})$. The right-to-left implication holds by a similar argumentation. □

Now we study the application of the solver to equations in clausal contexts. The purpose of the solver is to extract maximal Φ-top subterms from theory contexts. Indeed, if we apply it to a ground equation and obtain a solution $\{\vec{x} \simeq \vec{u}\}$, then each of the x_i is actually a canonizer variable, i. e. a Φ-top term, whereas the variables within $\vec{u}$ are existentially quantified. We formalize this as follows:

Definition 3.17 Consider a ground clause $C = D \vee s \simeq t$. Depending on the outcome of $\overline{\mathsf{solve}}(s \simeq t)$, the *solved form of* C *at* $s \simeq t$ is defined as:
(i) if the result is $\{\}$, the tautological clause $\top$;
(ii) in case of $\bot$, the subclause D;

(iii) for $\{\vec{x} \simeq \vec{u}\}$, each of the clauses $D \vee x_i \simeq \widehat{u}_i$ where $\widehat{\vec{u}}$ denotes $\vec{u}$ after consistently replacing existentially quantified variables with fresh Skolem functions.

Note that because of the restriction to ground clauses, only Skolem constants will occur in (iii); they are shared between different solved forms only if they arise from the same Skolemization. We consider them as part of the free signature Φ. The effect of the solving operation on semantics still needs to be characterized. Proposition 3.16 and the fact that the interpretation of the Skolem constants can be chosen suitably imply that a clause is satisfiable if and only if the set of its solved form is:

Proposition 3.18 If an interpretation $I^\Phi \in \mathcal{T}^\Phi$ satisfies $D \vee s \simeq t$, then there is an extension of I^Φ to the Skolem constants occurring in any of the solved forms such that the extension satisfies every solved form. Conversely, if there is such an extension of I^Φ, then I^Φ satisfies $D \vee s \simeq t$.

Proof: If solving does not change the clause, or if $I^\Phi \models D$, then the statement is obvious. Otherwise we have $I^\Phi \models s \simeq t$. So $\overline{\mathsf{solve}}(s \simeq t)$ cannot return $\bot$, but only $\{\vec{x} \simeq \vec{u}\}$. The first property follows from Prop. 3.16 and the fact that the interpretation of the Skolem constants can be chosen suitably. The converse part is shown analogously. □

The solving operation still has to be related to the ordering structure that is imposed on terms by the black-box path ordering: First, we want to ensure that solving and canonizing a clause can be done in finite time, i. e., solving should be non-increasing. Second, the outcome $x_i \simeq \widehat{u}_i$ shall be interpreted as a form of definition of x_i and applied in left-to-right direction; and in order to achieve this, the definition must be decreasing.

Assumption 3.19 We require that whenever $D \vee x_i \simeq \widehat{u}_i$ is a solved form of $D \vee s \simeq t$, then $s \simeq t \succeq x_i \simeq \widehat{u}_i$ and $x_i \succ \widehat{u}_i$ hold.

The restriction is always met if the complex solutions $\{\vec{x} \simeq \vec{u}\}$ that $\overline{\mathsf{solve}}(s \simeq t)$ returns are unitary and x is the maximal canonizer variable of the equation, as is for instance the case for linear arithmetic: We get $x \succ \widehat{u}$ by the multiset property of the black-box path ordering $\succ$, and $s \simeq t \succeq x \simeq \widehat{u}$ additionally by its subterm property.

Finally we stipulate how to apply the solver to literals in clausal contexts: eagerly, but to strictly maximal equations only.

Convention 3.20 Every clause $C \equiv D \vee s \simeq t$ where $s \simeq t$ and s are strictly maximal, is without loss of generality presented such that

(i) C equals its solved form at $s \simeq t$, and
(ii) s is canonical.

Furthermore, for every clause $C \equiv D \vee s \not\simeq t$ where $s \not\simeq t$ is maximal and s is strictly so, we stipulate that s shall be canonical.

Indeed every clause C which does not adhere to the conditions can be transformed into an equisatisfiable set of clauses that does: Until the condition is met, C is repeatedly replaced by the canonized set of all the corresponding solved forms, which by Ass. 3.14 is always finite. By Ass. 3.19 and König's Lemma this replacing process terminates; and Prop. 3.18 guarantees equisatisfiability. The unit clause $f(c) + 3g(1) - 5f(c + 0) \simeq 0$ of linear arithmetic is first solved to $f(c + 0) \simeq \frac{1}{5}f(c) + \frac{3}{5}g(1)$, then canonized to $f(c) \simeq \frac{1}{5}f(c) + \frac{3}{5}g(1)$ and finally solved to $f(c) \simeq \frac{3}{4}g(1)$. For a clause D that explicitly needs to be reshaped, as is the case for conclusions in inference rules, by $\Gamma(D)$ we denote the application of such a reshaping postprocessor. Since a clause can have more than one solved form, we may view Γ as a non-deterministic function returning one of them.

3.4 Superposition

3.4.1 Ground Case

In the following we restrict our attention to ground clauses. With the machinery developed so far, extending the superposition calculus to work modulo a Shostak theory $\mathcal{T}^\Phi$ is more or less straightforward, in that the argumentations for the standard calculus smoothly carry over. Mainly we have to integrate the canonizer into the model construction process and ensure that the coherence pairs are trivial. This will be possible because maximal equations within clauses are solved such that they have a Φ-top left-hand side. As was shown in Sect. 3.3.3, extending $\widehat{\sigma}$ by such rules preserves confluence. The presentation of our calculus follows the lines of [NR01].

Definition 3.21 The *superposition calculus modulo* $\mathcal{T}^\Phi$ for ground clauses consists of the following inference rules:

Superposition

$$\mathcal{I}\,\frac{C \vee l \simeq r \quad s[l] \bowtie t \vee D}{\Gamma(C \vee s[r] \bowtie t \vee D)} \qquad \text{if } \begin{array}{l} \cdot\ l \simeq r,\ l \text{ and } s \text{ are strictly maximal} \\ \cdot\ s \bowtie t \text{ is maximal,} \\ \quad \text{and strictly maximal for } \bowtie = \simeq \end{array}$$

Equality resolution

$$\mathcal{I}\,\frac{C \vee t \not\simeq t}{\Gamma(C)} \qquad \text{if } \cdot\ t \not\simeq t \text{ is maximal}$$

Equality factoring

$$\mathcal{I}\,\frac{C \vee s \simeq t \vee s \simeq t'}{\Gamma(C \vee t \not\simeq t' \vee s \simeq t')} \qquad \text{if } \begin{array}{l} \cdot\ s \simeq t \text{ is maximal} \\ \cdot\ s \text{ is strictly maximal in } s \simeq t \end{array}$$

To get a flavour of how derivations in this calculus proceed, consider the clause set $\{f(a-1)-1 \simeq a+1; f(b)-1 \not\simeq a+1 \vee f(b) \simeq b-2; b+1 \simeq a\}$. Assume $f(b) \succ a \succ b$. Reshaping the clauses according to our convention we obtain (1) $\underline{f(a-1)} \simeq a+2$, (2) $\underline{f(b)} \not\simeq a+2 \vee f(b) \simeq b-2$ and (3) $\underline{a} \simeq b+1$, where maximal terms in maximal literals are underlined. From (1) and (3) we obtain by the superposition rule (4) $\underline{f(b)} \simeq a+2$, which with (2) gives $a+2 \not\simeq a+2 \vee \underline{f(b)} \simeq b-2$. This clause and (4) lead to $\underline{a+2} \not\simeq \underline{a+2} \vee a+2 \simeq b-2$. From this, equality factoring deduces $\underline{a} \simeq b-4$. With (3) we obtain the empty clause.

We now employ the proof technique of *model generation* of [BG94] to show that every saturated clause set free of the empty clause has a model. This model will be a congruence induced by a convergent ground rewrite system. The canonizer $\widehat{\sigma}$ is part of that system. In the sequel S denotes an arbitrary, fixed set of ground clauses according to Conv. 3.20. Every contained clause may have to contribute a rule to become valid in the model:

Definition 3.22 For $C = C' \vee l \simeq r \in S$, we define $\mathrm{Gen}(C) = \{l \Rightarrow r\}$ if the following conditions hold, and $\mathrm{Gen}(C) = \{\}$ otherwise:

(i) $R^*_{C,\widehat{\sigma}} \not\models C$,
(ii) $l \succ r$, and $l \simeq r$ is strictly maximal in C,
(iii) l is irreducible with respect to R_C,
(iv) $R^*_{C,\widehat{\sigma}} \not\models t \simeq r$ for all $l \simeq t$ in C';

with the shorthand notations $R_C = \bigcup_{D \prec C} \mathrm{Gen}(D)$ and $R_{C,\widehat{\sigma}} = R_C \cup \widehat{\sigma}$, and $(\_)^*$ denoting closure under congruence axioms. If C has no positive literal, then let $\mathrm{Gen}(C) = \{\}$.
Furthermore, R_S shall stand for $\bigcup_{D \in S} \mathrm{Gen}(D)$, and $R_{S,\widehat{\sigma}}$ for $R_S \cup \widehat{\sigma}$.

Note that in this definition l is canonical because of Conv. 3.20. Now the shape of a generated rule $l \Rightarrow r$ is the following: By Conv. 3.20, the containing clause is solved at $l \simeq r$. By construction of $\overline{\mathsf{solve}}$, it is not possible that both l and r are ground theory terms. Because of $l \succ r$ and Ass. 3.19, the left-hand side l must be a Φ-top term. We summarize:

Proposition 3.23 If $l \Rightarrow r$ is a generated rule, then l is a canonical Φ-top term.

The next object of study is $R_{S,\widehat{\sigma}}$ which comprises the canonizer-related rules of $\widehat{\sigma}$ and the generated rules. By construction it is embedded in the black-box path ordering $\succ$ and therefore terminating. This allows to deduce confluence from the joinability of critical pairs. Divergencies between rules of $\widehat{\sigma}$ converge due to Prop. 3.11. Pairs between $\widehat{\sigma}$ and generated rules are joinable by Thm. 3.13; in that sense, R_S and $\widehat{\sigma}$ are coherent. R_S is confluent by the same argumentation as for standard superposition.

Lemma 3.24 The rewrite system $R_{S,\widehat{\sigma}}$ is terminating and confluent.

Proof: Rewrite steps in $\longrightarrow_{\widehat{\sigma}} \subseteq \succ$ are decreasing by Prop. 3.11. For steps in $\longrightarrow_{\{l\Rightarrow r\}}$ where $l \Rightarrow r$ is a generated rule, l is a canonical Φ-top term by Prop. 3.23; and as discussed in the proof of Thm. 3.13, such reductions are also decreasing. Now termination allows to deduce confluence from the joinability of critical pairs. Pairs between rules of $\widehat{\sigma}$ have been proven joinable in Prop. 3.11. Were there a non-joinable critical pair between $\widehat{\sigma}$ and a generated rule $l \Rightarrow r$, then $\widehat{\sigma} \cup \{l \Rightarrow r\}$ would not be confluent, contradicting Thm. 3.13. In that sense R_S and $\widehat{\sigma}$ are coherent. Finally, if R_S had non-joinable critical pairs, then also some finite subset thereof. We show via induction on clauses with respect to $\succ$ that, however, every $R_C \cup \mathrm{Gen}(C)$ is confluent. Assume that clause C generates $l \Rightarrow r$. Overlaps into l à la $r \;{}_{\{l\Rightarrow r\}}\!\longleftarrow l[l']_p \longrightarrow_{R_C} l[r']$ contradict condition (iii) of Def. 3.22 that l is irreducible with respect to R_C. For overlaps from l like $r' \;{}_{R_C}\!\longleftarrow l'[l]_{i.p} \longrightarrow_{\{l\Rightarrow r\}} l'[r]$, assume that $l' \Rightarrow r'$ has been produced by clause $D \prec C$. That implies $l \succ l'$ and contradicts the subterm property of $\succ$. □

Proposition 3.25 Validity from $R^*_{C,\widehat{\sigma}}$ to $R^*_{S,\widehat{\sigma}}$.
(i) If $C \equiv l \simeq r \vee D$ generates $l \Rightarrow r$, then $R^*_{C,\widehat{\sigma}} \not\models D$ and $R^*_{S,\widehat{\sigma}} \not\models D$.
(ii) $R^*_{C,\widehat{\sigma}} \models C$ implies $R^*_{S,\widehat{\sigma}} \models C$.

Proof: As to (i), $R^*_{C,\widehat{\sigma}} \not\models D$ just holds by the definition of the generation of rules, and implies that $s \downarrow_{R_{C,\widehat{\sigma}}} t$ for all $s \not\simeq t$ in D. Then $R_{C,\widehat{\sigma}} \subseteq R_{S,\widehat{\sigma}}$ yields that $s \downarrow_{R_{S,\widehat{\sigma}}} t$ and $R^*_{S,\widehat{\sigma}} \not\models D$.

Concerning (ii), let $C \equiv l \simeq r \vee D$. Either we have $R^*_{C,\widehat{\sigma}} \not\models D$; then $R^*_{C,\widehat{\sigma}} \models l \simeq r$ and $R^*_{S,\widehat{\sigma}} \models l \simeq r$ since $R_{C,\widehat{\sigma}} \subseteq R^*_{S,\widehat{\sigma}}$. Or $R^*_{C,\widehat{\sigma}} \models D$, i. e. $s \not\downarrow_{R_{C,\widehat{\sigma}}} t$ for some $s \not\simeq t$ in D. Then $R^*_{S,\widehat{\sigma}} \not\models D$ would require $s \downarrow_{R_{S,\widehat{\sigma}}} t$ and hence if wlog. $s \succeq t$ that s becomes reducible by some rule $l' \Rightarrow r'$ generated by a clause $C' \succ C$. But then by construction of $\succ$ we would get $l' \succ l \succeq s[l'] \succeq l'$. □

Finally, the congruence $R^*_{S,\widehat{\sigma}}$ generated by this rewrite system exhibits a $\mathcal{T}^\Phi$-model for S under appropriate conditions: namely S being closed under superposition and not containing the empty clause. By Prop. 3.23 the restriction of $R^*_{S,\widehat{\sigma}}$ to Δ is just $\widehat{\sigma}^*$, which is in $\mathcal{T}$ because $\mathcal{T}$ contains its σ-models. Similar to the proof in [NR01, p. 388f] we now will show that $R^*_{S,\widehat{\sigma}} \models C$ holds for every clause $C \in S$, essentially via induction on clauses with respect to $\succ$ and case distinction as to the occurrence of the maximal literal. Additionally one has to take into account that the postprocessor turns the syntactic conclusion into an equisatisfiable set of solved forms which are smaller or equal.

Lemma 3.26 If S is closed under superposition and does not contain the empty clause, then $R^*_{S,\widehat{\sigma}}$ is a $\mathcal{T}^\Phi$-model of S.

Proof: Proposition 3.23 implies that the restriction of $R^*_{S,\widehat{\sigma}}$ to Δ is just $\widehat{\sigma}^*$, which is in $\mathcal{T}$ because $\mathcal{T}$ contains its σ-models. So we have to show that $R^*_{S,\widehat{\sigma}} \models C$ for every clause $C \in S$, which is done via induction on clauses. We split with respect to the occurrence of its maximal literal $s \bowtie s$, or $s \bowtie t$ where $s \succ t$.

- $s \not\simeq t \vee D$: If $R^*_{S,\widehat{\sigma}} \models s \not\simeq t$, then we are already done. Otherwise, we need to show $R^*_{S,\widehat{\sigma}} \models D$. Confluence of $R_{S,\widehat{\sigma}}$ (Lem. 3.24) implies that $s \downarrow_{R_{S,\widehat{\sigma}}} t$. Because of $s \succ t$, the term s must be $R_{S,\widehat{\sigma}}$-reducible. Since s is canonical by Conv. 3.20, the reduction is say by $l \Rightarrow r \in R_S$ generated by some clause $l \simeq r \vee D' \in S$. That clause overlaps with C, satisfying the ordering restrictions. Since S is closed under superposition, all the conclusions $C'' \equiv \Gamma(C') \equiv \Gamma(s[r] \not\simeq t \vee D \vee D')$ are also contained in S. Then by construction of $\succ$ and non-increasingness of solving we have $C \succ C' \succeq C''$ and hence inductively $R^*_{S,\widehat{\sigma}} \models C''$. Because of the equisatisfiability of C' and the set of all the $\Gamma(C')$ (see Prop. 3.18 and the discussion of Conv. 3.20), $R^*_{S,\widehat{\sigma}} \models C'$ is also true. Since $l \Rightarrow r$ has been generated, we have $R^*_{S,\widehat{\sigma}} \models l \simeq r$ and $R^*_{l \simeq r \vee D',\widehat{\sigma}} \not\models D'$, as well as $R^*_{S,\widehat{\sigma}} \not\models D'$ by Prop. 3.25 (i). So $R^*_{S,\widehat{\sigma}} \models C'$ spells out as $R^*_{S,\widehat{\sigma}} \models s[r] \not\simeq t \vee D$, which implies $R^*_{S,\widehat{\sigma}} \models C$.
- $s \simeq t \vee D$: Unless $R^*_{S,\widehat{\sigma}} \models s \simeq t$ we have to show $R^*_{S,\widehat{\sigma}} \models D$. Evidently C has not generated $s \Rightarrow t$, such that one of the conditions of Def. 3.22 (i)-(iv) must have been violated. If this applies to condition (i), then $R^*_{C,\widehat{\sigma}} \models C$ is true, and so is $R^*_{S,\widehat{\sigma}} \models C$ according to the monotonicity stated in Prop. 3.25 (ii).
  If condition (iv) is not satisfied, then $R^*_{C,\widehat{\sigma}} \models t \simeq t'$ holds for one positive literal $s \simeq t'$ of D, such that the clause C has a presentation $s \simeq t \vee s \simeq t' \vee D'$. Furthermore, it serves as premise in equality

resolution inferences with conclusions $C'' \equiv \Gamma(C') \equiv \Gamma(t \not\simeq t' \vee s \simeq t' \vee D')$. The set S is saturated and contains all these clauses C''. From $s \succ t$ and $s \simeq t \succeq s \simeq t'$ follow $s \simeq t \succ t \not\simeq t'$ and $C \succ C' \succeq C''$, such that we obtain inductively $R^*_{S,\widehat{\sigma}} \models C''$, and like in the above case by equisatisfiability also $R^*_{S,\widehat{\sigma}} \models C'$. Because of $R^*_{C,\widehat{\sigma}} \models t \simeq t'$ and Prop. 3.25 (ii) this can more precisely be stated as $R^*_{S,\widehat{\sigma}} \models s \simeq t' \vee D'$, which entails $R^*_{S,\widehat{\sigma}} \models C$.
In the sequel we me assume that condition (iv) is fulfilled, which means that $R^*_{C,\widehat{\sigma}} \models t \not\simeq t'$ holds for all positive literals $s \simeq t'$ of D. In particular D has no further occurrence of $s \simeq t$. Hence $s \simeq t$ and s are strictly maximal, such that condition (ii) is met, and (iii) must be violated. Therefore s is R_c-reducible, say by $l \Rightarrow r \in R_C$ generated by some clause $l \simeq r \vee D'$ smaller than C. From these two clauses, the inference rule superposition right produces the conclusions $C'' \equiv \Gamma(C') \equiv \Gamma(s[r] \simeq t \vee D \vee D')$, which are in S by saturatedness. From $\{s \simeq t\} \succeq \{l \simeq r\} \succ D'$ follows $C \succ C' \succeq C''$. Inductively we get $R^*_{S,\widehat{\sigma}} \models C''$, and then by equisatisfiability $R^*_{S,\widehat{\sigma}} \models C'$. Since $l \Rightarrow r$ has been generated, we know that $R^*_{l \simeq r \vee D',\widehat{\sigma}} \not\models D'$, such that $R^*_{S,\widehat{\sigma}} \not\models D'$ by Prop. 3.25 (i). Therefore $R^*_{S,\widehat{\sigma}} \models C'$ implies $R^*_{S,\widehat{\sigma}} \models s[r] \simeq t \vee D$, from which $R^*_{S,\widehat{\sigma}} \models C$ follows via $R^*_{S,\widehat{\sigma}} \models l \simeq r$.

- $s \not\simeq s \vee D$: In this case S must contain all conclusions $C' \equiv \Gamma(D)$ from equality resolution inferences. Like above these are smaller than C, such that we have inductively $R^*_{S,\widehat{\sigma}} \models C'$ for every C', next by equisatisfiability $R^*_{S,\widehat{\sigma}} \models D$, and finally $R^*_{S,\widehat{\sigma}} \models C$.
- $s \simeq s \vee D$: Then C is trivially satisfiable.

□

Summing it up, unsatisfiable clause sets are just those that cannot be closed under superposition without producing the empty clause. Hence we have in the usual sense:

Theorem 3.27 Superposition modulo $\mathcal{T}^\Phi$ is refutationally complete for ground clauses with respect to $\mathcal{T}^\Phi$.

But recall that this requires that the ordering restriction imposed in Ass. 3.19 has been met for any call of the solver. For situations in which this cannot be achieved, a possible way of circumventing the problem may consist in considering some of the coherence pairs.

Let us briefly remark that the concept of *redundancy* smoothly carries over from the ordinary superposition calculus, as well as the concept of *selection*. Just like the standard inference system, superposition modulo $\mathcal{T}^\Phi$ is turned into a decision procedure for $\mathcal{T}^\Phi$-satisfiability of ground Horn clauses

by means of eager selection (cf. [NR01, p. 411f]). Via splitting of clauses, this can be extended to a decision procedure for arbitrary ground clauses, similar to what is sketched for the standard calculus at the end of Sect. 6.2.

3.4.2 Towards Non-ground Clauses

The calculus that we have considered so far works on ground clauses. In non-ground clauses, schematic variables are universally quantified within their clauses, and each clause represents the set of all its canonized ground instances. If we want to extend our calculus to non-ground clauses, then inferences between these clauses must represent all necessary inferences between their instances in a finite way. In the usual superposition calculus, this happens essentially by replacing equality tests in the inference rules by unification. In our case, it is also necessary to extend the other basic operations of the calculus to non-ground terms or formulas.

Extending the black-box path ordering to non-ground terms is somewhat tedious but possible in principle. Of course, the extended ordering is not total on non-ground terms, so that maximal terms or literals in a clause are no longer uniquely determined. One has to beware of the following pitfall, though: If a schematic variable occurs directly below a theory context, then in a ground instance, the variable may be replaced by a Δ-top term, thus enlarging the theory context. Such a context is therefore smaller than free function symbols, but essentially incomparable with other contexts (unless required by the subterm property).

Can we also extend the canonizer and the solver to non-ground terms and literals? If schematic variables occur only directly below free function symbols, the answer is again positive. If the ordering is not total on the subterms of a theory context, a case split on the ordering relations possible may lead to more than one result, but the number of resulting terms or formulas is always finite. On the other hand, schematic variables occurring directly below theory contexts will now lead to failure: in a ground instance, they may be replaced by Δ-top terms, thus enlarging the theory context, and the set of all canonizations or solved forms of these ground instances is in general not finitely representable. This case can only be handled if more information on the internals of the canonizer and the solver is available. Note however that canonization and solving are relevant only for maximal or selected literals; so schematic variables in other literals are always unproblematic.

3.5 Summary

In this chapter, we have integrated a Shostak theory into the superposition calculus. The Shostak-style components for deciding the clausal validity problem, namely canonizer and solver, have been employed as simplification devices, so that no coherence pairs between theory axioms and other axioms have to be considered. Under the assumption that the solver meets some ordering restriction, as is the case for linear arithmetic, our calculus is refutationally complete on mixed ground clauses.

4 A Superposition View on Nelson-Oppen

4.1 Introduction

Following a line of research to prove decidability and combination results in a resolution framework, we reinspect the Nelson-Oppen combination procedure [NO79, Opp80, TH96] and re-establish its correctness as an instance of the completeness of the superposition calculus [BG94].

In doing so, we employ superposition as a particular means of generating models. The starting point is an observation by Bachmair that given two clause sets over disjoint signatures without isolated variable occurrences, if each of the sets is saturated and does not contain the empty clause, so is their union, which is therefore satisfiable. With a suitable fine-tuning of the calculus parameters, this result can be adapted to the Nelson-Oppen setting where the signatures share a finite number of free constants.

Our goal in this research is two-fold. Firstly, a number of attempts have been undertaken to generalize the Nelson-Oppen procedure, relaxing either the signature disjointness condition or the stably infiniteness requirement. Can these also be rephrased and maybe extended within the superposition framework? Secondly, extensions of superposition, like the chaining calculus for transitive relations [BG98b], are promising candidates for stretching the limits of the combination scheme, say for reasoning about data types that are based on some ordering.

4.2 Reconstructing Nelson-Oppen

The component theories of the Nelson-Oppen procedure are *stably infinite* (Def. 2.10 (i)): On quantifier-free formulae, $\mathcal{T}$-satisfiability is equivalent to $\mathcal{T}$-satisfiability in infinite models. A related property of a theory is *convexity*

(Def. 2.10 (ii)), holding in case that if a clause is $\mathcal{T}$-valid, then also some Horn clause is with the same disequations, but only one of the equations. Every convex theory without singleton models is stably infinite (Cor. 2.12). Convexity is also an important requirement to Shostak-style theory reasoning (cf. Chap. 3).

4.2.1 Obtaining a Clausal Theory Presentation

We have defined theories to consist of arbitrary first-order formulae, whereas the superposition calculus deals with formulae in clause normal form. Hence our first task is to find a clausal theory presentation. To ease the presentation of the Nelson-Oppen procedure, we will assume that theories are not only stably infinite, but also convex. We only need a single sort and assume that equality is the only predicate symbol.

In the superposition framework, when combining clause sets over disjoint signatures, inferences different from the superposition rule are unary and do not produce conclusions in the combined signature. The superposition rule behaves the same as long as the terms in the clauses that are to be unified both start with a function symbol. On the other hand, clauses with positive isolated occurrences of variables are harmful. Consider for example the variable x in the clause $x \simeq a \vee C$ where x does not appear in C: Some instances of x are strictly maximal then, such that the inference is necessary. But if we apply some splitting rule, the malicious clause is broken into C and $x \simeq a$, the latter clause contradicting stable infiniteness.

We will exploit convexity in order to present the theory as a set of Horn clauses. Let us then call a Horn clause C *unshielding* if $C \equiv x \simeq t \vee C'$ where x occurs neither in t nor in C'. Still $x \simeq t$ contradicts stable infiniteness. However, if we want to replace the clause C by C' within a superposition derivation, we have to apply the concept of redundancy, which demands that the replacement be not also smaller, but also logically implied. The implication holds if the theory contains an axiom $\forall x \exists y.\, x \not\simeq y$, or a Skolemized version thereof. It turns out that this can be required in our context without loss of generality. Summing it up, stable infiniteness allows us to get rid of unshielded variable occurrences.

The following lemma makes our considerations precise.

Lemma 4.1 Let $\mathcal{T}$ denote a stably infinite and convex Σ-theory and $\#$ a unary operator not in Σ. Then there exists a set T of Horn clauses over $\Sigma \cup \{\#\}$ such that, going from $\mathcal{T}$ to T,

(i) stable infiniteness is preserved,

(ii) convexity is preserved,

(iii) the sets of theory-satisfiable quantifier-free formulae over Σ coincide,

(iv) and every non-trivial $\mathcal{T}$-model can be extended to a T-model.

Moreover the operator $\#$ occurs in exactly one clause of T, which is $x \not\approx \#(x)$.

Proof: We will develop T in several steps, namely via a sequence of theories $\mathcal{T}_i$, and will show that, whenever going from $\mathcal{T}_i$ to $\mathcal{T}_{i+1}$, the properties given from (i) through (iv) are preserved. We start start with $\mathcal{T}_0 = \mathcal{T}$, and $\mathcal{T}_1$ is the set of $\mathcal{T}_0$-valid universal formulae. The theory $\mathcal{T}_2$ is a standard clause normal form of $\mathcal{T}_1$. By convexity, for every non-Horn clause $\bigvee_i A_i \to \bigvee_j B_j \in \mathcal{T}_2$ there is a $\mathcal{T}_2$-equivalent Horn subclause $\bigvee_i A_i \to B_j$. Replacing all clauses that way gives the set $\mathcal{T}_3$. Adding the clause $x \not\approx \#(x)$, we obtain the theory $\mathcal{T}_4 = T$. Step-wise preservation of properties (i) through (iv) remains to be proved.

The first step is from $\mathcal{T}_0$ to $\mathcal{T}_1$:

(iv) Because of $\mathcal{T}_0 \supseteq \mathcal{T}_1$, every $\mathcal{T}_0$-model is also a $\mathcal{T}_1$-model.

(iii) Let ϕ denote a quantifier-free Σ-formula. Because of (iv), if ϕ is $\mathcal{T}_0$-satisfiable, it is also $\mathcal{T}_1$-satisfiable. Conversely, if ϕ is $\mathcal{T}_1$-satisfiable, then we have $\mathcal{T}_1 \cup \{\exists X.\, \phi\} \not\models \bot$ and $\mathcal{T}_1 \not\models \forall X.\, \neg\phi$, such that in particular $\forall X.\, \neg\phi \notin \mathcal{T}_1$ holds. By construction of $\mathcal{T}_1$ and quantifier-freeness of ϕ we know that $\mathcal{T}_0 \not\models \forall X.\, \neg\phi$ is true, which implies $\mathcal{T}_0 \cup \{\exists X.\, \phi\} \not\models \bot$ and $\mathcal{T}_0$-satisfiability of ϕ.

(i) Given a $\mathcal{T}_1$-satisfiable quantifier-free formula ϕ, we need to exhibit an infinite $\mathcal{T}_1$-model of ϕ. Now, ϕ is $\mathcal{T}_0$-satisfiable by (iii), even in an infinite model because $\mathcal{T}_0$ is stably infinite. By (iv) this model is also a $\mathcal{T}_1$-model.

(ii) Let $C \equiv \bigwedge_i A_i \to \bigvee_j B_j$ denote a non-negative clause such that $\mathcal{T}_1 \models \forall X.\, C$ is valid. By construction of $\mathcal{T}_1$ we obtain $\mathcal{T}_0 \models \forall X.\, C$, and by convexity of $\mathcal{T}_0$ furthermore $\mathcal{T}_0 \models \forall X.\, \bigwedge_i A_i \to B_j$ for some j. Hence $\forall X.\, \bigwedge_i A_i \to B_j$ is contained in $\mathcal{T}_1$ and therefore $\mathcal{T}_1$-valid.

Going from $\mathcal{T}_1$ to $\mathcal{T}_2$, we note that CNF-transforming universal formulae preserves equivalence, such that the two theories are equivalent. Moving on to $\mathcal{T}_3$, every formula in $\mathcal{T}_2$ is $\mathcal{T}_3$-valid and vice versa, such that equivalence is given again. Adding the clause $x \not\approx \#(x)$, we arrive at $\mathcal{T}_4$. Property (iv) is established by suitable interpretation of $\#$, which is always possible in non-trivial models. Convexity (ii) holds because $\mathcal{T}_4$ is a Horn theory. Stable infiniteness (i) follows from the absence of trivial models and Cor. 2.12. Property (iii) holds because of $\mathcal{T}_4 \supseteq \mathcal{T}_3$ and (iv). $\square$

4.2.2 Combining Convex Theories

To simplify the presentation, we restrict ourselves to the important special case that the component theories are not only stably infinite, but also convex. The Nelson-Oppen procedure then becomes simpler in that no initial guessing of equivalences between the shared variables is needed, or in other terms: branching is not required.

The procedure deals with finitely many theories $\mathcal{T}_1, \ldots, \mathcal{T}_n$, each $\mathcal{T}_i$ being built over some signature Σ_i. The signatures are disjoint. Every theory $\mathcal{T}_i$ comes with a decision procedure for $\mathcal{T}_i$-satisfiability of quantifier-free formulae. Let $\mathcal{T}$ denote the union of the theories, and Σ the union of the signatures. Our aim is to decide $\mathcal{T}$-satisfiability of a finite set Γ of Σ-literals. This straightforwardly extends to a procedure for arbitrary quantifier-free formulae. In order to accommodate to the superposition setting, we replace the variables in Γ by free constant symbols from an additional, finite signature Σ_0, such that from now on Γ is ground.

We exploit as usually that Γ can be *purified*, in linear time, with the help of free constants from Σ_0 into an equivalent presentation such that each literal is built from symbols of Σ_0 and just one component signature Σ_i, $i > 0$. We partition Γ accordingly into $\bigcup_{i \geq 0} \Gamma_i$ where Γ_0 gathers the literals over Σ_0 (and variables), which are common to all component signatures.

Let us introduce two notions to be employed within the description of the procedure: We say that Γ *is component satisfiable* if every $\Gamma_i \cup \Gamma_0$ is $\mathcal{T}_i$-satisfiable, and that Γ *is component unsatisfiable* otherwise. Furthermore, a Σ_0-*conclusion of* Γ shall be any Σ_0-equation $a \simeq a'$ such that the literal set $\Gamma \cup \{a \not\simeq a'\}$ is component unsatisfiable.

Algorithm 4.2 *Nelson-Oppen procedure [NO79]* for convex theories:

(0) Purify and partition Γ.
(1) Exhaustively, unless Γ is component unsatisfiable,
extend Γ with Σ_0-conclusions.
(2) Return satisfiable if Γ is component satisfiable,
and unsatisfiable otherwise.

Of the properties we will need to prove the combination procedure correct, let us briefly gather the evident ones:

Proposition 4.3 The notions of component satisfiability and Σ_0-conclusion are subject to the following properties:

(i) Component satisfiability is decidable.
(ii) For Γ, component unsatisfiability entails $\mathcal{T}$-unsatisfiability.

(iii) If $a \simeq a'$ is a Σ_0-conclusion of Γ, then Γ and $\Gamma \cup \{a \simeq a'\}$ are $\mathcal{T}$-equivalent.
(iv) The set of all Σ_0-conclusion of Γ is computable.

Proof: Statement (i) follows directly from the definition of component satisfiability. Regarding (ii), the entailment holds because any $\mathcal{T}$-model of Γ would also be a $\mathcal{T}$-model of Γ_i and therefore a $\mathcal{T}_i$-model. Concerning (iii), the unsatisfiability condition in the definition of Σ_0-conclusion corresponds, in terms of entailment, to $\mathcal{T}_i \models \Gamma_0 \wedge \Gamma_i \rightarrow a \simeq a'$ in case we read the literal sets as conjunctions of their elements. Hence $\mathcal{T} \models \Gamma \rightarrow a \simeq a'$. Statement (iv) holds because $\mathcal{T}_i$-satisfiability is decided by the given procedure for $\mathcal{T}_i$ and because Σ_0 is finite. □

The less evident property is that, if the procedure claims satisfiability, one can indeed exhibit some $\mathcal{T}$-model. We will use a superposition-based model construction to this end.

Lemma 4.4 If Γ is closed under adding Σ_0-conclusions, then component satisfiability implies $\mathcal{T}$-satisfiability.

Proof: By Lem. 4.1, for every theory $\mathcal{T}_i$ over Σ_i there is a Horn clause set T_i over $\Sigma_i \cup \{\#_i\}$ with respect to which the same quantifier-free formulae over $\Sigma_i \cup \Sigma_0$ are satisfiable. Every non-trivial $\mathcal{T}_i$-model can be extended to a T_i-model, and T_i contains the operator $\#_i$ exactly in the clause $x \not\simeq \#_i(x)$.

Consider any theory $\mathcal{T}_i$. Since the literal set $\Gamma_i \cup \Gamma_0$ is $\mathcal{T}_i$-satisfiable, it is also T_i-satisfiable. Equivalently, the clause set $\Gamma_i \cup \Gamma_0 \cup T_i$ is satisfiable, and therefore has a logically equivalent presentation S_i which is saturated, in the sense of the superposition calculus (cf. [NR01, Sect. 3–4]). Our aim is to show that with a suitable fine-tuning of the calculus parameters the union $S := \bigcup_i S_i$ is immediately saturated and, since it does not contain the empty clause, thereby satisfiable according to the completeness of the superposition calculus.

The parameter setting is the following:

(i) We choose a lexicographic path ordering with some total precedence over $\Sigma \cup \Sigma_0 \cup \{\#_1, \ldots, \#_n\}$ such that symbols from Σ_0 are minimal.
(ii) Eager selection is applied to negative equations in Σ_0-clauses.
(iii) Every unshielding clause $y \simeq t \vee C$ produces, with a clause $x \not\simeq \#_i(x)$ under the unifier $\{y \mapsto \#_i(x)\}$, the superposition conclusion $x \not\simeq t \vee C$, which in turn produces C by equality resolution. Thereby the unshielding clause becomes redundant such that it can be dropped. This is done eagerly.

We now discuss which inference rule can produce conclusions in which symbols from different Σ_i are mixed. The rule *equality factoring* is not applicable since every clause is Horn. The rule *equality resolution* is unary. Every conclusion is already contained in that clause set S_i which holds the premise, or is redundant there, and therefore need not be added to S. Likewise, applications of the binary *superposition rule* are not critical if both premises belong to the same S_i.

So it remains to study an inference between a clause $C \vee l \simeq r \in S_i$ and another clause $D \vee s[l'] \bowtie t \in S_j$. By definition of the superposition rule, the subterm l' is not a variable. If l were a variable x, then any further occurrence of x within the first premise would violate the strict maximality condition on l. Hence the first premise were an unshielding clause; but such clauses are dropped.

Therefore both l and l' are headed by function symbols, which have to coincide since l and l' are unifiable. This common symbol has to belong to the signature part common to S_i and S_j, namely Σ_0. Since Σ_0 is minimal in the precedence, the maximality conditions imply that both premises contain function symbols only from Σ_0.

Because eager selection is applied to negative equations in Σ_0-clauses, the first premise $C \vee l \simeq r \in S_i$ of the superposition inference is simply an equation $l \simeq r$, or more precisely $a \simeq a'$ because unshielding clauses are dropped.

Since every non-trivial model of $\Gamma_i \cup \Gamma_0 \cup \mathcal{T}_i$ extends to one of S_i, and since $a \simeq a'$ holds in every model of S_i, we obtain that $a \simeq a'$ must be valid in $\Gamma_i \cup \Gamma_0 \cup \mathcal{T}_i$. Hence $a \simeq a'$ is a Σ_0-conclusion of Γ and contained in Γ_0, as well as in every S_k. In particular the first premise is also an element of S_j, so that the superposition inference is not needed. □

Theorem 4.5 For convex component theories, Alg. 4.2 decides $\mathcal{T}$-satisfiability of quantifier-free formulae.

Proof: As outlined in the beginning of the subsection, Step (0) is computable and transforms Γ into a $\mathcal{T}$-equisatisfiable set of literals. By Prop. 4.3(i),(iv) every iteration of Step (1) is computable; by Prop. 4.3(iii) it produces another $\mathcal{T}$-equisatisfiable presentation of Γ. Since Σ_0 is finite, Step (1) terminates, and so does the procedure. After Step (1), the resulting literal set Γ is closed under adding Σ_0-conclusions. If Γ is component satisfiable, then by Lem. 4.4 a $\mathcal{T}$-model exists. Otherwise it is $\mathcal{T}$-unsatisfiable by Prop. 4.3(ii). □

5 Dealing with Bits and Vectors thereof

5.1 Datapath Verification with SPASS

In the context of the Verisoft project [Ver08], the datapath of a simple DLX-style processor [HP96] has successfully been verified with SPASS, as a case study to assess the potential of first-order theorem proving for hardware verification. The processor, a circuit diagram of which is depicted in Fig. 5.1, is a 32-bit architecture. Its state comprises 2^5 registers (see GPR in diagram), 2^{32} bytes of memory (m), and a program counter (PC), thereby creating a state space of size $2^{2^{37}}$. For the sake of simplicity, the machine comes without interrupts and without pipelining. Three addressing modes are provided: immediate, register and displacement. The instructions either load, store, compute, branch, or jump.

Let us have a look at the execution of a particular load instruction where the 32-bit instruction word is partitioned into operation code OPC, source register RS1, destination register RD and immediate operand imm as follows:

OPC_6	$RS1_5$	RD_5	imm_{16}

The bit pattern for the operation code OPC here is 100011. The number of the destination register is given by RD. The memory address to be read from is the sum of the immediate operand imm and of the contents of register number RS1. Symbolically this intended semantics can be denoted as:

$$GPR[RD] := m[GPR[RS1] + imm]$$

The execution of each single instruction is arranged in five subsequent stages, such that the processor could be refined to work in a pipelined fashion. In the first stage (instruction fetch), the program counter PC puts its contents onto the address line of the instruction memory IM, which in turn outputs

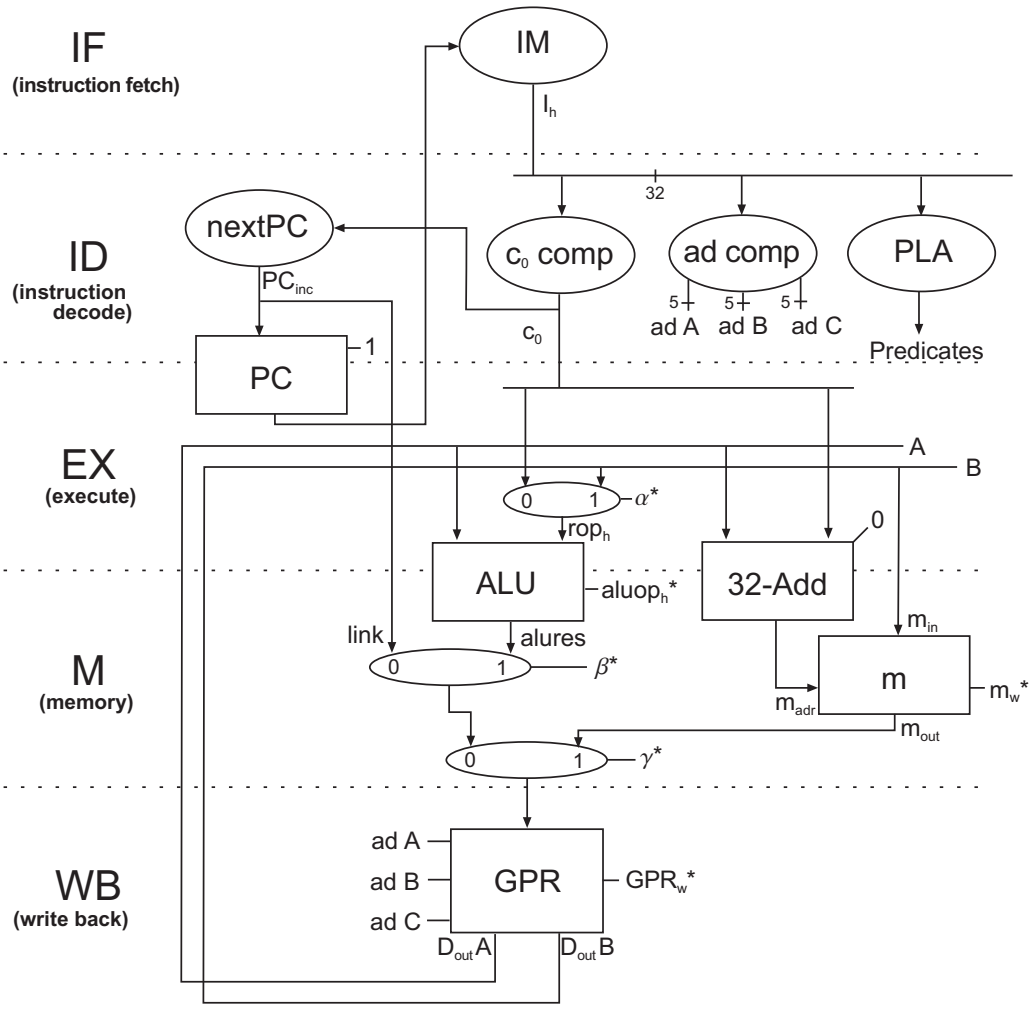


Figure 5.1: DLX hardware configuration, arranged in execution stages

the current instruction word $\mathsf{I_h}$, in our case a load instruction. In the second stage (instruction decode), this instruction word is split according to the above scheme, and the resulting parts are forwarded to the components they are relevant for. The operation code OPC is decoded within the PLA circuit; the latter part realizes the control logic of the processor. The source register address RS1 is fed into adA and the destination register address RD into adC, whilst the immediate operand imm goes into $\mathsf{c_0}$. In the third stage (execute), the register bank GPR puts its contents at address adA onto bus A, from where it jointly goes into the adder 32−Add with $\mathsf{c_0}$. In the fourth stage (memory), the sum of both, which is the address GPR[RS1] + imm, is put on the address line of memory m, and the corresponding memory contents arrives on $\mathsf{m_{out}}$. In the fifth stage (write back), this result is written into the register bank GPR at address adC; and as an epilogue, the program counter PC is incremented.

The functionality of the processor has been specified via a state transition function, where state means the contents of registers, memory and program counter modelled as bitvectors with a fixed length of 32 bits, and each transition describes the effect of executing a single instruction. The arithmetical-logical unit ALU and the adder 32−Add have been assumed as correct. A correspondence relation has been declared between abstract state and hardware state, simply as coincidence of contents. In this setting, processor verification means to prove that the correspondence is maintained when executing one arbitrary instruction.

Proving the processor correct with pen and paper turned out to be tedious. The statement needed to be split up into 3 lemmas and 15 propositions, not all of which were trivial. Using the Isabelle system [NPW02] was quite laborious as well; the completed proof script was 500 lines long. This motivated the automation of proof.

A straightforward choice would have been to employ dedicated decision procedures. Such have been obtained in various ways. For example, in [CMR97] the theory of fixed-sized bitvectors is decided via Shostak-style reasoning, by construction of a canonizer and of a solver. However, the signature is fixed, and Shostak's method is restricted to validity in the universal fragment. Evidently, fixed-size bitvectors are not convex; however, the authors axiomatize them by a set of positive unit equations. A similar Shostak-style treatment is given in [BP98]. Bitvectors are encoded in second-order monadic logic on strings in [BK95], resulting in an automata-based decision procedure for an expressive fragment that includes quantification over bitvectors. In [Pic03], bitvectors of symbolic lengths are considered. This extension allows to reason about parameterized hardware descriptions. Tableau-style decision procedures are presented for various fragments, up to

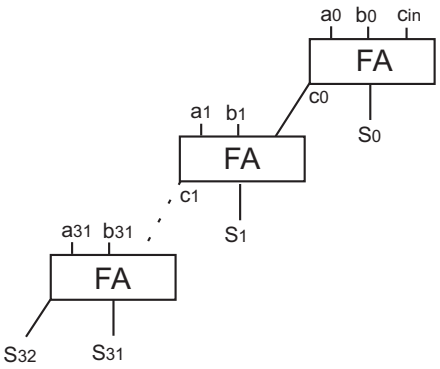


Figure 5.2: Carry-chain adder

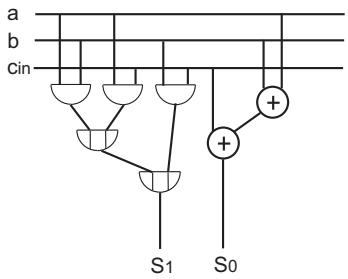


Figure 5.3: Full adder

some arithmetic on indices. The annual competition on satisfiability modulo theories (SMT-COMP, [BdMS05, SD07]) includes two categories related to fixed-size bitvectors: in terms of validity, the universal fragment with and without free functions. In 2007, three systems were competing in these categories (SPEAR, YICES and Z3). They all relied on variants of bit-blasting, reducing bit-vector formulae to the propositional level.

Contrary to these established approaches, the focus in our case study was on the question what could be achieved with first-order methods. Since these are less specialized, weaker performance was to be expected, and a priori those methods are only semi-decision procedures. On the other hand, first-order methods offer higher expressiveness: (i) Arbitrary first-order formulae can be used instead of only universal ones. (ii) Furthermore, non-theory operators can be introduced, and one can even give axioms for them. This allows to abstract over implementation details, like in the following simple example: Consider the 32-bit carry-chain adder in Fig. 5.2, composed of bit-level full adders FA as shown in Fig. 5.3, and assume we want to show that the input vectors $\vec{a}$ and $\vec{b}$ may interchanged without affecting the sum $\vec{S}$. With a bit-blasting approach, one goes down to the complete bit-level specification of the adder and proves two variants thereof equivalent. With axiomatized non-theory operators, however, one can first prove that the full adder ignores the interchange of the input bits a and b, and then prove the actual conjecture for an abstracted version of the carry-chain adder where FA is a black box except that each a_i and b_i can be interchanged. (iii) As another advantage of first-order methods, let us remark that bitvectors can freely be combined with arbitrary other sorts that capture for example other data types.

In the beginning of the case study, various options of representing bitvectors were pursued: using lists over constants 0 and 1 like in $\mathrm{cons}(1, \mathrm{cons}(1, \mathrm{cons}(0, \mathrm{nil})))$, or words over the unary functions 0 and 1 as in $1(1(0(\epsilon)))$, also

arrays like in $\mathrm{wrt}(2, \mathrm{wrt}(1, \mathrm{wrt}(0, \mathrm{empty}(), 0), 1), 1)$, and finally constructor functions as in $\mathrm{bv}_3(1, 1, 0)$. Only the last of these approaches turned out viable in practice: For all the others, inductive lemmas like associativity of append had to be added, which produced loads of resolvents. Roughly, the former approaches rely on fragments of recursive axiomatizations, whereas the latter is complete in first-order logic already. It was hence decided to use fixed-length bitvectors in a many-sorted setting. Since in concrete hardware the size of any bitvector is always given, this was perfectly suited for the application at hand.

The base level of this axiomatization is constituted by a sort B that represents bits, using the constants 0 and 1. These are distinct and represent the domain; and operations like negation $^-$, conjunction $\cdot$ and disjunction $+$ are defined as expected:

$$\begin{array}{l} 0 \not\simeq 1 \\ \forall x \in B.\ x \simeq 0 \vee x \simeq 1 \\ \quad \overline{0} \simeq 1 \qquad \overline{1} \simeq 0 \\ 0 \cdot 0 \simeq 0 \quad 0 \cdot 1 \simeq 0 \quad 1 \cdot 0 \simeq 0 \quad 1 \cdot 1 \simeq 1 \\ 0 + 0 \simeq 0 \quad 0 + 1 \simeq 1 \quad 1 + 0 \simeq 1 \quad 1 + 1 \simeq 1 \end{array}$$

On the vector level, for every length m there is an own sort B^m and an m-place constructor function $(\_, \ldots, \_)^m$ from bits to B^m. In turn, there are m projection functions P_i^m for the extraction of the i-th bit, where i ranges from 0 to $m - 1$. Construction and projection are related as follows:

$$\begin{array}{rl} & \forall x \in B^m.\ (P_{m-1}^m(x), \ldots, P_0^m(x))^m \simeq x \\ \forall x_{m-1}, \ldots, x_0 \in B. & P_i^m((x_{m-1}, \ldots, x_0)^m) \simeq x_i \end{array}$$

On top of these, one can easily introduce operations like concatenation $:_n^m$, which joins a vector of length m and another of length n, and slicing P_{ji}^m, which from a vector of length m extracts the subvector from position j down to position i:

$$\begin{array}{l} \forall x_{m-1}, \ldots, x_0, y_{n-1}, \ldots, y_0 \in B. \\ \quad (x_{m-1}, \ldots, x_0)^m :_n^m (y_{n-1}, \ldots, y_0)^n \simeq (x_{m-1}, \ldots, x_0, y_{n-1}, \ldots, y_0)^{m+n} \\ \quad \wedge \qquad\qquad P_{ji}^m((x_{m-1}, \ldots, x_0)^m) \simeq (x_j, \ldots, x_i)^{j-i+1} \end{array}$$

Next we give some examples how these constructs were employed in the case study. The state space for the transition function was modelled as non-bitvector sort C, its elements being thought of as configurations. For every configuration c, the expression $\mathrm{mem}(c, x)$ denotes the memory contents at address x, the term $\mathrm{PC}(c)$ stands for the current program counter, and $\mathrm{GPR}(\mathrm{c}, \mathrm{i})$ is the contents of the i-th register. Based on these definitions, the

instruction word $\mathrm{I}(c)$ is the memory contents of where the program counter points to, the operation code $\mathrm{opc}(c)$ is one subvector thereof, and the test bit $\mathrm{sw}(c)$ is set if and only if the operation code is 101011, which stands for "store word":

$$\begin{aligned}\forall c \in C. \qquad & \mathrm{I}(c) \simeq \mathrm{mem}(c, \mathrm{PC}(c)) \\ \wedge\ & \mathrm{opc}(c) \simeq P^{32}_{31,26}(\mathrm{I}(c)) \\ \wedge\ & \mathrm{sw}(c) \simeq \mathrm{eq}^6(\mathrm{opc}(c), (1,0,1,0,1,1)^6)\end{aligned}$$

The expression $\mathrm{eq}^6(x, y)$ in the definition of sw shall equal 1 in case the bitvectors x and y are equal, and 0 otherwise. – The instruction word has one field denoting a destination register, and another containing a direct operand. From these an effective address $\mathrm{ea}(c)$ is computed, which is the memory address to write to. The data stems from the source register $\mathrm{RS}(c)$, which is yet another part of the instruction word. Using a functional if-then-else $\_\,?\,\_ :^{32} \_$ on bitvectors, we are ready to specify the consecutive state of memory. Each memory location remains unchanged unless a store command is executed and affects that very address:

$$\begin{aligned}&\forall c \in C.\ \forall x \in B^{32}. \\ &\quad \mathrm{nextmem}(c, x) \simeq \\ &\qquad \mathrm{sw}(c) \cdot \mathrm{eq}^{32}(x, \mathrm{ea}(c))\ ? \\ &\qquad\quad \mathrm{GPR}(c, \mathrm{RS}(c)) :^{32} \mathrm{mem}(c, x)\end{aligned}$$

An empirical finding in the case study was that on the level of bits, the cardinality-constraining axiom $\forall x \in B.\, x \simeq 0 \vee x \simeq 1$ was difficult to reason with for the SPASS theorem prover: The unshielded variable x unifies with every variable-disjoint term. Hence this clause overlaps with any other at every non-variable subterm position for which the maximality conditions on the enclosing term and literal are met, and loads of resolvents are produced. Experimental results will be given at the end of this chapter; they demonstrate that reasoning with this clause is hopeless in practice.

Another observation was that unsatisfiability modulo the theory of bits can quickly be proved for unsatisfiable clause sets where all clauses are made up of positive literals like $s \simeq 0$ and $t \simeq 1$ with no Boolean operator on top of s or t. This phenomenon is even more dramatic in the ground case, the reason being that SPASS via splitting and standard simplifications then reduces to a DPLL-like procedure [DP60, DLL62]. The latter is the method of choice for propositional satisfiability problems.

Notably every clause set can be transformed into such a shape. First of all, modulo the theory of bits the clause $s \simeq t \vee C$ is equivalent to the conjunction of the two clauses $s \simeq 0 \vee t \simeq 1 \vee C$ and $s \simeq 1 \vee t \simeq 0 \vee C$. Symmetrically,

$s \not\simeq t \vee C$ is equivalent to $s \simeq 0 \vee t \simeq 0 \vee C$ and $s \simeq 1 \vee t \simeq 1 \vee C$. Secondly, getting rid of top-level theory operators is easy: For example, since in the theory of bits $s \cdot t \simeq 1$ is equivalent to $s \simeq 1 \wedge t \simeq 1$, we can replace the clause $s \cdot t \simeq 1 \vee C$ by the two clauses $s \simeq 1 \vee C$ and $t \simeq 1 \vee C$. Unsurprisingly, in the worst case formulae are blown up exponentially like in the standard computation of clause normal forms. There, given for example an $n \times 2$ matrix of propositional variables (P_{ij}) and a formula

$$\phi \equiv (P_{11} \wedge P_{12}) \vee \ldots \vee (P_{n1} \wedge P_{n2}) \equiv \bigvee_{i=1}^{n} \bigwedge_{j=1}^{2} P_{ij}$$

then its standard clause normal form is

$$\psi \equiv (P_{11} \vee \ldots \vee P_{n1}) \wedge \ldots \wedge (P_{12} \vee \ldots \vee P_{n2}) \equiv \bigwedge_{\vec{\jmath} \in [1;2]^n} \bigvee_{i=1}^{n} P_{ij_i}$$

Going from ϕ to ψ, the number of atom occurences increases from $2n$ to $2^n \cdot n$, hence by a factor of 2^{n-1}. Formula renaming is a good remedy here (cf. [NW01, Sect. 4]). Introducing a fresh abbreviation Q_i for every conjunction $P_{i1} \wedge P_{i2}$, we obtain the equisatisfiable formula

$$\begin{aligned} \chi \equiv (Q_1 \vee \ldots \vee Q_n) \wedge (\neg Q_1 \vee P_{11}) \wedge (\neg Q_1 \vee P_{12}) \\ \wedge \qquad\qquad \ldots \\ \wedge (\neg Q_n \vee P_{n1}) \wedge (\neg Q_1 \vee P_{n2}) \end{aligned}$$

Notably, the formula χ has only $5n$ atom occurences, which is just a linear increase from ϕ. This renaming technique should equally be applicable in our setting.

Consider now the case that the signature contains no other non-theory symbols than constants. Then a contradiction, if at all, can be shown without reference to the cardinality-constraining clause $x \simeq 0 \vee x \simeq 1$: For running the superposition calculus, assume that 0 and 1 are the smallest and the second smallest ground term, respectively. In this clause, the constant 1 can become strictly maximal only if $x\sigma \equiv 0$, but that instance is a tautology. Hence, non-redundant superposition inferences involve the maximal literal $x \simeq 1$ and the maximal term x such that x is bound neither to 0 nor to 1. Then with a side premise $C \vee c \simeq 0$ we obtain $C \vee c \simeq 0 \vee 1 \simeq 0$, which is subsumed by the side premise. With $C \vee c \simeq 1$ we get $C \vee c \simeq 0 \vee 1 \simeq 1$, which is a tautology. Hence, all inference conclusions of the cardinality-constraining clause are redundant.

Coming back to the general case, this observation extends to arbitrary non-theory symbols if we require that for every occurrence like in $f(t) \simeq 0 \vee C$, there is an additional clause $t \simeq 0 \vee t \simeq 1 \vee C$, or a solved form thereof if t has a theory operator on top. The reason is that the former and the latter clause entail $t \simeq 0 \vee f(1) \simeq 0 \vee C$, which is exactly what a superposition inference at t would have produced.

These considerations motivate an alternative axiomatization on the level of bits which performs the transformation of the clause set into the particular shape on the fly. Disregarding non-theory operators, the following is needed:

$$
\begin{array}{lll}
0 \not\simeq 1 & & \\
\forall x, y \in B. & \overline{x} \simeq 0 \leftrightarrow x \simeq 1 & \overline{x} \simeq 1 \leftrightarrow x \simeq 0 \\
& x \cdot y \simeq 0 \leftrightarrow x \simeq 0 \vee y \simeq 0 & x \cdot y \simeq 1 \leftrightarrow x \simeq 1 \wedge y \simeq 1 \\
& x + y \simeq 0 \leftrightarrow x \simeq 0 \wedge y \simeq 0 & x + y \simeq 1 \leftrightarrow x \simeq 1 \vee y \simeq 1
\end{array}
$$

A clausal version thereof is given by:

$$
\begin{array}{ll}
0 \not\simeq 1 & \\
\overline{x} \simeq 0 \rightarrow x \simeq 1 & \overline{0} \simeq 1 \\
\overline{x} \simeq 1 \rightarrow x \simeq 0 & \overline{1} \simeq 0 \\
 & 0 \cdot x \simeq 0 \\
x \cdot y \simeq 0 \rightarrow x \simeq 0 \vee y \simeq 0 & x \cdot 0 \simeq 0 \\
x \cdot y \simeq 1 \rightarrow x \simeq 1 & 1 \cdot x \simeq x \\
x \cdot y \simeq 1 \rightarrow y \simeq 1 & x \cdot 1 \simeq x \\
 & 0 + x \simeq x \\
x + y \simeq 0 \rightarrow x \simeq 0 & x + 0 \simeq x \\
x + y \simeq 0 \rightarrow y \simeq 0 & 1 + x \simeq 1 \\
x + y \simeq 1 \rightarrow x \simeq 1 \vee y \simeq 1 & x + 1 \simeq 1
\end{array}
$$

Assume we want to prove de Morgan's law $\overline{c \cdot d} \simeq \overline{c} + \overline{d}$ via superposition, say using the lexicographic path ordering induced by the precedence $\cdot \succ + \succ {}^{-} \succ c \succ d \succ 1 \succ 0$. We have to derive the empty clause $\bot$ from the rephrased axiomatization above and the two clauses $\overline{c \cdot d} \simeq 0 \vee \overline{c} + \overline{d} \simeq 0$ and $\overline{c \cdot d} \simeq 1 \vee \overline{c} + \overline{d} \simeq 1$. The former gives rise to just one inference, producing $c \cdot d \simeq 1 \vee \overline{c} + \overline{d} \simeq 0$, which in turn produces $c \simeq 1 \vee \overline{c} + \overline{d} \simeq 0$ and $d \simeq 1 \vee \overline{c} + \overline{d} \simeq 0$. Continuing this way, we finally obtain the two clauses $c \simeq 1$ and $d \simeq 1$. In a similar fashion, the clause $\overline{c \cdot d} \simeq 1 \vee \overline{c} + \overline{d} \simeq 1$ gives rise to $c \simeq 0 \vee d \simeq 0$, leading to $0 \simeq 1$ and $\bot$.

Compare this with a proof based on the original axiomatization. The negated conjecture is the clause $\overline{c \cdot d} \not\simeq \overline{c} + \overline{d}$. Each derivation starts with an overlap from $x \simeq 0 \vee x \simeq 1$ into the negated conjecture. In the maximal literal, there are four positions for doing so: $\overline{c \cdot d}$, $c \cdot d$, c and d. (i) From $\overline{c \cdot d}$ one obtains a valid clause. (iv) From d we obtain after simplification the clause $d \simeq 0 \vee \overline{c} \not\simeq \overline{c} + \overline{d}$. Continuing with an overlap at d within $\overline{c} + \overline{d}$, which is one of five choices, produces $d \simeq 0$, with which the negated conjecture can be rewritten into the contradiction $1 \not\simeq 1$. (iii) Starting with c is symmetric. (ii) From $c \cdot d$ one gets $\bot$ in twice as many steps as in the derivations (iv) and (iii).

Summing it up, it seems that the rephrased axiomatization induces a search space with less branching, and is less prolific in this sense. But proofs are longer in terms of inferences, because expressions with theory operators on top-level are solved. On the other hand, these particular inferences are actually simplifications and could be implemented as such.

The deeper the expressions in a problem formulation are, the more evident the difference between the two approaches becomes in practice. For example, the 4×4 Sudoku puzzle to the right has no solution. Such a Sudoku can be seen as a matrix A over $[1;4]$ such that every projection to a row, column or subsquare is (i) injective and (ii) surjective. Propositional encodings are obtained in 4^3 variables P_{ij}^d, each identified with the condition $a_{ij} = d$, and adding conditions on (iii) the given digits, (iv) the codomain of A, and (v) the functionality of A; see [LO06] and [HTW06] for details. If one compiles each of the conditions (i) through (v) into one equation over bits, then one obtains a challenging test problem. On a Dell PowerEdge server equipped with 3GHz Xeon processors, it took SPASS half an hour to derive the empty clause from the standard axiomatization of bits, whereas with the rephrased one less than 30 seconds were needed.

·	·	·	4
3	·	·	·
·	·	·	·
2	·	1	·

In this chapter, the rephrased axiomatization on the bit level is studied in detail. We show that it, with an appropriate treatment of conjectures, constitutes a sound and complete method for reasoning about bits. Interestingly, in the ground case, which corresponds to universal validity, superposition with standard simplifications is a decision procedure. We briefly discuss how our approach can be extended to small bounded domains. Note that here the adaptation is via reshaping of axiomatization and conjecture, whereas in Chap. 6 we study a refinement of superposition on the calculus level, but without theory operators. Finally we present and analyze a detailed first-order theory of fixed-size bitvectors, and sketch how it can be combined with the rephrased axiomatization of bits.

5.2 Axioms on Bits

For dealing with bits, we use a signature Σ that contains a single sort only. The set $\mathscr{F}$ of operators contains, besides arbitrary free elements, the symbols 0, 1, $^-$, $\cdot$, $+$, $\oplus, \ominus$ that are intended to denote falsum, verum, negation, conjunction, disjunction, antivalence, equivalence respectively, where 0 and 1 are constants, the operator $^-$ is unary and $\cdot$, $+$, $\oplus, \ominus$ are binary. The set $\mathscr{P}$ of predicate symbols is arbitrary.

Our target structure, the algebra of propositions, can be axiomatized

simply by stating that there are exactly two elements, and by exhibiting truth tables for the connectives:

Definition 5.1 The following Σ-formulae, universally closed, define the theory $\mathbb{B}$:

$$
\begin{array}{llll}
0 \not\simeq 1 & x \simeq 0 \vee x \simeq 1 & & \\
\overline{0} \simeq 1 & \overline{1} \simeq 0 & & \\
0 \cdot 0 \simeq 0 & 0 \cdot 1 \simeq 0 & 1 \cdot 0 \simeq 0 & 1 \cdot 1 \simeq 1 \\
0 + 0 \simeq 0 & 0 + 1 \simeq 1 & 1 + 0 \simeq 1 & 1 + 1 \simeq 1 \\
0 \oplus 0 \simeq 0 & 0 \oplus 1 \simeq 1 & 1 \oplus 0 \simeq 1 & 1 \oplus 1 \simeq 0 \\
0 \ominus 0 \simeq 1 & 0 \ominus 1 \simeq 0 & 1 \ominus 0 \simeq 0 & 1 \ominus 1 \simeq 1
\end{array}
$$

From these axioms follow, essentially by case splits, the laws of Boolean algebra, like $\forall x.\, 1 + x \simeq 1$. The axiomatization is the simplest one in the sense that it contains only one non-ground formula, and that clearly all operators are just definitional extensions of $\{0 \not\simeq 1, \forall x.\, x \simeq 0 \vee x \simeq 1\}$. Given any two models of $\mathbb{B}$, their reducts to the non-free symbols are isomorphic and correspond to the intended algebra of propositions. So without free symbols there is essentially one model, and then $\mathbb{B}$ is a complete theory.

Next we introduce a weak version of a rephrased axiomatization as presented informally in the preceding subsection. From $\mathbb{B}$ we drop the axiom that every element is 0 or 1. To compensate for this, we give solver-like axioms for the theory operators, in place of the truth tables. A similar axiom is necessary for any operator f that is free with respect to $\mathbb{B}$.

Definition 5.2 The following Σ-formulae, universally closed, define the theory $\mathbb{C}$:

$$
\begin{array}{ll}
0 \not\simeq 1 & \\
\overline{x} \simeq 0 \to x \simeq 1 & \overline{x} \simeq 1 \to x \simeq 0 \\
x \cdot y \simeq 0 \to x \simeq 0 \vee y \simeq 0 & x \cdot y \simeq 1 \to x \simeq 1 \wedge y \simeq 1 \\
x + y \simeq 0 \to x \simeq 0 \wedge y \simeq 0 & x + y \simeq 1 \to x \simeq 1 \vee y \simeq 1 \\
x \oplus y \simeq 0 \to (x \simeq 0 \vee y \simeq 1) \wedge (x \simeq 1 \vee y \simeq 0) & \\
x \oplus y \simeq 1 \to (x \simeq 0 \vee y \simeq 0) \wedge (x \simeq 1 \vee y \simeq 1) & \\
x \ominus y \simeq 0 \to (x \simeq 0 \vee y \simeq 0) \wedge (x \simeq 1 \vee y \simeq 1) & \\
x \ominus y \simeq 1 \to (x \simeq 0 \vee y \simeq 1) \wedge (x \simeq 1 \vee y \simeq 0) & \\
f(\vec{x}) \simeq 0 \vee f(\vec{x}) \simeq 1 \to \bigwedge_i (x_i \simeq 0 \vee x_i \simeq 1) \text{ for every free } f \in \mathscr{F} &
\end{array}
$$

If we instantiate the schematic axiom for free operators with a constant, then the conjunction becomes empty, i. e. just $\top$. Hence the axiom in this case is a tautology and can be left away.

Evidently $\mathbb{C}$ is a subtheory of $\mathbb{B}$, but not very useful by itself: For example we have $\mathbb{C} \models 1 \cdot 1 \not\simeq 0$, but already $\mathbb{C} \not\models 1 \cdot 1 \simeq 1$. The approximation is

meant to be the minimal one that still ensures completeness if combined with the transformation of conjectures which is still to be introduced. The idea behind the transformation is to replace every literal by a formula in which the top-level terms of the literal evaluate to 0 or 1, and to consider the literal true in a don't-care manner otherwise. But let us before have a closer look at the relation between $\mathbb{B}$ and $\mathbb{C}$.

Definition 5.3 The following notions are useful to describe $\mathbb{C}$ more abstractly:

(i) The set of *01-constraints for* $\vec{x}$ is the smallest set that contains all atoms $x_i \simeq 0$ and $x_j \simeq 1$, and is closed under conjunction and disjunction.

(ii) By definition (cf. Def. 5.2), the theory $\mathbb{C}$ for every operator $op \in \{^{-}, \cdot, +, \oplus, \ominus\}$ and every $\alpha \in \{0,1\}$ contains exactly one formula of the form $\forall X.\, op(\vec{x}) \simeq \alpha \to S^{op}_{\alpha}(\vec{x})$ where $S^{op}_{\alpha}(\vec{x})$ is a 01-constraint for $\vec{x}$. We call $S^{op}_{\alpha}(\vec{x})$ the *solution of op for* α. Let furthermore $S^{op}_{\alpha}(\vec{t}) \equiv (S^{op}_{\alpha}(\vec{x}))\{\vec{x} \mapsto \vec{t}\}$.

(iii) For every Σ-interpretation $\mathcal{A}_\mu$,
  (a) let $01^{\mathcal{A}}$ denote the set $\{0^{\mathcal{A}}, 1^{\mathcal{A}}\}$,
  (b) $\mathscr{T}[\mathcal{A}_\mu] = \mathcal{A}_\mu^{-1}(01^{\mathcal{A}})$ the set of terms that $\mathcal{A}_\mu$ maps into $01^{\mathcal{A}}$,
  (c) $\mathscr{A}[\mathcal{A}_\mu] = \{A \in \mathscr{A}(\Sigma) : A|_i \in \mathscr{T}[\mathcal{A}_\mu] \text{ for every } i\}$ the set of atoms over these terms, and
  (d) $\mathscr{Q}[\mathcal{A}_\mu]$ the set of quantifier-free formulae over these atoms.

(iv) Let $\vec{t} \in 01^*$ abbreviate the formula $\big(\bigwedge_i (t_i \simeq 0 \vee t_i \simeq 1)\big)$, and $\vec{t} \notin 01^*$ the formula $\big(\bigvee_i (t_i \not\simeq 0 \wedge t_i \not\simeq 1)\big)$.

Proposition 5.4 We compare how in $\mathbb{C}$ and in $\mathbb{B}$ the solutions $S^{op}_{\alpha}(\vec{x})$ are related to $op(\vec{x}) \simeq \alpha$, and show that validity of 01-constraints may carry over from $\mathbb{C}$-models to $\mathbb{B}$-models.

(i) For every solution $S^{op}_{\alpha}(\vec{x})$ the following hold:
  (a) $\mathbb{C} \models op(\vec{x}) \simeq \alpha \to S^{op}_{\alpha}(\vec{x})$
  (b) $\mathbb{B} \models op(\vec{x}) \simeq \alpha \leftrightarrow S^{op}_{\alpha}(\vec{x})$

(ii) Consider a $\mathbb{C}$-model $\mathcal{A}_\mu$, a $\mathbb{B}$-model $\mathcal{B}_\nu$, a 01-constraint ϕ for $\vec{x}$ and a substitution $\sigma = \{\vec{x} \mapsto \vec{t}\}$ such that $\mathcal{A}_\mu = \mathcal{B}_\nu$ on $\{0,1\} \cup (\{\vec{t}\} \cap \mathscr{T}[\mathcal{A}_\mu])$. Then $\mathcal{A}_\mu \models \phi\sigma$ entails $\mathcal{B}_\nu \models \phi\sigma$.

Proof:

(i) The first statement holds by definition of $S^{op}_{\alpha}(\vec{x})$. Since $\mathrm{Ded}(\mathbb{C}) \subseteq \mathrm{Ded}(\mathbb{B})$ this implies $\mathbb{B} \models op(\vec{x}) \simeq \alpha \to S^{op}_{\alpha}(\vec{x})$. What remains to show is the converse implication. For negation, conjunction and disjunction this is obvious. As to anti- and equivalence, let us for an example look at $S^{\oplus}_0$, which is $(x \simeq 0 \vee y \simeq 1) \wedge (x \simeq 1 \vee y \simeq 0)$ and $\mathbb{B}$-equivalent to $(x \simeq 1 \to y \simeq 1) \wedge (x \simeq 0 \to y \simeq 0)$ and $x \simeq y$. Note that this

definition of $S_0^{\oplus}$ is preferable over the perhaps more obvious one via $(x \simeq 0 \wedge y \simeq 0) \vee (x \simeq 1 \wedge y \simeq 1)$, in that it will produce half as many clauses during clausification.

(ii) We proceed by induction over the structure of 01-constraints.

- $\phi \equiv x_1 \simeq \alpha$, $\alpha \in \{0,1\}$: Since $\mathcal{A}_\mu \models t_1 \simeq \alpha$ we know that $\mathcal{A}_\mu(t_1) \in 01^{\mathcal{A}}$ and $t_1 \in \mathscr{T}[\mathcal{A}_\mu]$; hence $\mathcal{A}_\mu(t) = \mathcal{B}_\nu(t)$ and also by assumption $\alpha^{\mathcal{A}} = \alpha^{\mathcal{B}}$. So $\mathcal{A}_\mu(t_1) = \alpha^{\mathcal{A}}$ can simply be rewritten as $\mathcal{B}_\nu(t_1) = \alpha^{\mathcal{B}}$, such that $\mathcal{B}_\nu \models (x_1 \simeq \alpha)\{x_1 \mapsto t_1\}$.
- $\phi \equiv \phi_1 \wedge \phi_2$: From $\mathcal{A}_\mu \models (\phi_1 \wedge \phi_2)\sigma$ we get that both $\mathcal{A}_\mu \models \phi_1\sigma$ and $\mathcal{A}_\mu \models \phi_2\sigma$ hold. So do inductively $\mathcal{B}_\nu \models \phi_1\sigma$ and $\mathcal{B}_\nu \models \phi_2\sigma$, which implies $\mathcal{B}_\nu \models (\phi_1 \wedge \phi_2)\sigma$.
- $\phi \equiv \phi_1 \vee \phi_2$: Similarly $\mathcal{A}_\mu \models (\phi_1 \vee \phi_2)\sigma$ implies that one of $\mathcal{A}_\mu \models \phi_1\sigma$ and $\mathcal{A}_\mu \models \phi_2\sigma$ is true, hence inductively one of $\mathcal{B}_\nu \models \phi_1\sigma$ and $\mathcal{B}_\nu \models \phi_2\sigma$, and finally $\mathcal{B}_\nu \models (\phi_1 \vee \phi_2)\sigma$.

□

Lemma 5.5 For every $\mathbb{C}$-model $\mathcal{A}_\mu$ there exists a $\mathbb{B}$-model $\mathcal{B}_\nu$ such that $\mathcal{A}_\mu$ and $\mathcal{B}_\nu$ coincide on $\mathscr{T}[\mathcal{A}_\mu]$ and $\mathscr{Q}[\mathcal{A}_\mu]$.

Proof: We construct a Σ-structure $\mathcal{B}$ as follows: The carrier is $B := 01^{\mathcal{A}}$. The constants 0 and 1 are interpreted as $0^{\mathcal{A}}$ and $1^{\mathcal{A}}$, respectively. The Boolean operators $^-$, $\cdot$, $+$, $\oplus$ and $\ominus$ are interpreted as expected for the algebra of propositions. For free functions f of arity n, take as $f^{\mathcal{B}}$ any function from B^n to B that coincides with $f^{\mathcal{A}}$ whenever the latter maps into B, i. e. $f^{\mathcal{B}} = f^{\mathcal{A}}$ on $(f^{\mathcal{A}})^{-1}(B) \cap B^n$. Finally, for every n-place predicate symbol P, let $P^{\mathcal{B}}$ equal the restriction of $P^{\mathcal{A}}$ to B^n. The structure $\mathcal{B}$ is closed under every operation, hence a Σ-algebra. Let $\nu\colon \mathscr{V} \to B$ denote an arbitrary assignment such that $\nu = \mu$ on $\mu^{-1}(B)$. By construction of $\mathcal{B}$, the interpretation $\mathcal{B}_\nu$ is a model of $\mathbb{B}$.

We now show that $\mathcal{A}_\mu = \mathcal{B}_\nu$ on $\mathscr{T}[\mathcal{A}_\mu]$. The proof is by induction over the term structure. In each case we consider the situation that $\mathcal{A}_\mu(t) \in B$, and have to show that then $\mathcal{A}_\mu(t) = \mathcal{B}_\nu(t)$.

- $t \equiv x$: $\mathcal{A}_\mu(x) = \mu(x) = \nu(x) = \mathcal{B}_\nu(x)$ by construction of ν
- $t \equiv 0$: [$t \equiv 1$:] We have stipulated above that $0^{\mathcal{A}} = 0^{\mathcal{B}}$ [$1^{\mathcal{A}} = 1^{\mathcal{B}}$].
- $t \in \{\overline{t_1}, t_1 \cdot t_2, t_1 + t_2, t_1 \oplus t_2, t_1 \ominus t_2\}$: Let op denote the operator on top of t. Since $\mathcal{A}_\mu(t) \in B$ we have $\mathcal{A}_\mu \models op(\vec{t}) \simeq \alpha$ for some $\alpha \in \{0,1\}$, by Prop. 5.4 (i)(a) hence $\mathcal{A}_\mu \models S_\alpha^{op}(\vec{t})$ or $\mathcal{A}_\mu \models \phi\sigma$ for the 01-constraint $\phi \equiv S_\alpha^{op}(\vec{x})$ and the substitution $\sigma = \{\vec{x} \mapsto \vec{t}\}$. Inductively we obtain for every i that if t_i is mapped into B by $\mathcal{A}_\mu$, then $\mathcal{A}_\mu(t_i) = \mathcal{B}_\nu(t_i)$. Because of $0^{\mathcal{A}} = 0^{\mathcal{B}}$ and $1^{\mathcal{A}} = 1^{\mathcal{B}}$ we get $\mathcal{A}_\mu = \mathcal{B}_\nu$ on $\{0,1\} \cup (\{\vec{t}\} \cap \mathscr{T}[\mathcal{A}_\mu])$. So we can apply Prop. 5.4 (ii), which yields $\mathcal{B}_\nu \models \phi\sigma$. The

latter can be spelled out as $\mathcal{B}_\nu \models S^{op}_\alpha(\vec{t})$, which by Prop. 5.4 (i)(b) delivers $\mathcal{B}_\nu \models op(\vec{t}) \simeq \alpha$ and finally $\mathcal{B}_\nu(t) = \mathcal{A}_\mu(t)$.

- $t \equiv f(\vec{t})$: Similarly to the previous case we obtain that $\mathcal{A}_\mu \models f(\vec{t}) \simeq \alpha$ for some $\alpha \in \{0, 1\}$. From $\mathbb{C} \models f(\vec{t}) \simeq \alpha \to \vec{t} \in 01^*$ follows that $\mathcal{A}_\mu \models \vec{t} \in 01^*$, hence inductively $\mathcal{A}_\mu(\vec{t}) = \mathcal{B}_\nu(\vec{t})$. By construction of $\mathcal{B}$ holds $f^{\mathcal{A}} = f^{\mathcal{B}}$ on $(f^{\mathcal{A}})^{-1}(B) \cap B^n$. Therefore we get $f^{\mathcal{A}}(\mathcal{A}_\mu(\vec{t})) = f^{\mathcal{B}}(\mathcal{B}_\nu(\vec{t}))$ and $\mathcal{A}_\mu(f(\vec{t})) = \mathcal{B}_\nu(f(\vec{t}))$.

Next we come to $\mathcal{A}_\mu = \mathcal{B}_\nu$ on $\mathscr{A}[\mathcal{A}_\mu]$. Regarding equations, if $s \simeq t \in \mathscr{A}[\mathcal{A}_\mu]$, then $\{s, t\} \subseteq \mathscr{T}[\mathcal{A}_\mu]$, hence $\{\mathcal{A}_\mu(s), \mathcal{A}_\mu(t)\} \subseteq B$ and by the previous paragraph $\mathcal{A}_\mu(s) = \mathcal{B}_\nu(s)$ and $\mathcal{A}_\mu(t) = \mathcal{B}_\nu(t)$. Therefore $\mathcal{A}_\mu(s) = \mathcal{A}_\mu(t)$ iff $\mathcal{B}_\nu(s) = \mathcal{B}_\nu(t)$. Let us now come to atoms $P(\vec{t})$ where $\vec{t}$ is a vector of n terms. Like in the case of equations we get $\mathcal{A}_\mu(\vec{t}) \in B^n$ and $\mathcal{A}_\mu(\vec{t}) = \mathcal{B}_\nu(\vec{t})$. By construction of $\mathcal{B}$ we have $P^{\mathcal{B}} = P^{\mathcal{A}}|_{B^n}$. Therefore $\mathcal{A}_\mu \models P(\vec{t})$ iff $P^{\mathcal{A}}(\mathcal{A}_\mu(\vec{t}))$ iff $P^{\mathcal{B}}(\mathcal{A}_\mu(\vec{t}))$ iff $P^{\mathcal{B}}(\mathcal{B}_\nu(\vec{t}))$ iff $\mathcal{B}_\nu \models P(\vec{t})$.

Finally the identity of $\mathcal{A}_\mu$ and $\mathcal{B}_\nu$ on $\mathscr{Q}[\mathcal{A}_\mu]$ is proven by induction over the structure of quantifier-free formulae. The base case is covered in the previous paragraph. For the logical constants $\top$ and $\bot$, the identity is obvious. Regarding complex formulae, let us have a look at a negation $\neg\phi$. There $\mathcal{A}_\mu \models \neg\phi$ iff $\mathcal{A}_\mu \not\models \phi$ iff, inductively, $\mathcal{B}_\nu \not\models \phi$ iff $\mathcal{B}_\nu \models \neg\phi$. The remaining cases are correspondingly. □

The proof of Lem. 5.5 exhibits in the part on the identity of $\mathcal{A}_\mu$ and $\mathcal{B}_\nu$ on $\mathscr{T}[\mathcal{A}_\mu]$ why our axiomatization has to contain implications like $\overline{x} \simeq 0 \to x \simeq 1$: They allow to conclude, given that a compound term evaluates into $01^{\mathcal{A}}$, that some or all of the direct subterms also do, such that one can trace the correspondence of $\mathcal{A}_\mu$ and $\mathcal{B}_\nu$ down to the components. This were not possible if, say, instead of the above implication we had stated $\overline{0} \simeq 1$ and $\overline{1} \simeq 0$.

In practice one will use stronger approximations of $\mathbb{B}$ that extend $\mathbb{C}$ with useful simplification rules like $x \cdot 0 \simeq 0$ and $x + 1 \simeq 1$. Since we do not want to fix a particular such approximation of $\mathbb{B}$ yet, let us just briefly stipulate the following:

Assumption 5.6 From now on, $\mathbb{D}$ denotes an arbitrary Σ-theory meeting the condition $\mathrm{Ded}(\mathbb{B}) \supseteq \mathrm{Ded}(\mathbb{D}) \supseteq \mathrm{Ded}(\mathbb{C})$.

5.3 A Transformational Approach

We will introduce a transformation Ξ that maps formulae to formulae. Ideally, we would like that Ξ were a sound and complete transformation from

$\mathbb{B}$ to $\mathbb{D}$. Let us first present the transformation informally on the level of equational literals $s \simeq t$. There, the mapping amounts to:

$$s \simeq t \;\overset{\Xi}{\longmapsto}\; \begin{array}{l}(s \simeq 0 \to t \not\simeq 1)\\ \wedge\, (s \simeq 1 \to t \not\simeq 0)\end{array} \qquad\qquad s \not\simeq t \;\overset{\Xi}{\longmapsto}\; \begin{array}{l}(s \simeq 0 \to t \not\simeq 0)\\ \wedge\, (s \simeq 1 \to t \not\simeq 1)\end{array}$$

We introduce a notation that focuses on the relevant parts of these formulae: For a formula or subformula ϕ and a Boolean connective $\otimes$, the expression $\bigotimes (\phi[{}^{s_1}_{t_1}] \ldots [{}^{s_n}_{t_n}])$ shall serve as an abbreviation for $((\phi[s_1] \ldots [s_n]) \otimes (\phi[t_1] \ldots [t_n]))$. The mapping can then be written as follows:

$$s \simeq t \;\overset{\Xi}{\longmapsto}\; \bigwedge (s \simeq {}^{0}_{1} \to t \not\simeq {}^{1}_{0}) \qquad\qquad s \not\simeq t \;\overset{\Xi}{\longmapsto}\; \bigwedge (s \simeq {}^{0}_{1} \to t \not\simeq {}^{0}_{1})$$

The idea behind the transformation is to replace every literal $s \bowtie t$ by a formula that restricts the equality or disequality to the case that both s and t evaluate to 0 or 1, and to consider the literal true in a don't-care manner otherwise. Note also that $\Xi(s \simeq t)$ is equivalent to $\Xi(t \simeq s)$.

For predicative literals $P(\vec{t})$ this idea leads to a scheme which is a bit more elaborate:

$$P(\vec{t}) \;\overset{\Xi}{\longmapsto}\; \vec{t} \in 01^* \to P(\vec{t}) \qquad\qquad \neg P(\vec{t}) \;\overset{\Xi}{\longmapsto}\; \vec{t} \in 01^* \to \neg P(\vec{t})$$

Actually this scheme generalizes the previous one: If we instantiate P with $\simeq$, then for example on the left we obtain $(s \not\simeq 0 \wedge s \not\simeq 1) \vee (t \not\simeq 0 \wedge 1 \not\simeq 1) \vee s \simeq t$, which is equivalent to the lengthy expression $(s \simeq 0 \wedge t \simeq 0 \to s \simeq t) \wedge (s \simeq 0 \wedge t \simeq 1 \to s \simeq t) \wedge (s \simeq 1 \wedge t \simeq 0 \to s \simeq t) \wedge (s \simeq 1 \wedge t \simeq 1 \to s \simeq t)$. Modulo $0 \not\simeq 1$ this simplifies into $\neg(s \simeq 0 \wedge t \simeq 1) \wedge \neg(s \simeq 1 \wedge t \simeq 0)$, being equivalent to $\Xi(s \simeq t)$.

As an alternative to the transformation scheme for predicates, one could turn P into a function symbol and code $P(\vec{t})$ as $P(\vec{t}) \simeq 1$ and $\neg P(\vec{t})$ as $P(\vec{t}) \simeq 0$. Removing tautologies and contradictions, the transformation then gives rise to:

$$P(\vec{t}) \simeq 1 \;\overset{\Xi}{\longmapsto}\; P(\vec{t}) \not\simeq 0 \qquad\qquad P(\vec{t}) \simeq 0 \;\overset{\Xi}{\longmapsto}\; P(\vec{t}) \not\simeq 1$$

The result looks simpler, but note that turning P into a function symbol requires to add the axiom $P(\vec{x}) \simeq 0 \vee P(\vec{x}) \simeq 1 \to \bigwedge_i (x_i \simeq 0 \vee x_i \simeq 1)$.

5.3.1 The Formula Transformation

In order to lift this idea to the level of formulae, we have to take care in which way an atom contributes to the truth value of the overall formula. To

this end, we simply embed it into a polarity-based computation of a negation normal form. Then atoms with positive polarity become subject to the left transformation scheme, and atoms with negative polarity to the right one.

The actual work is done within a two-place auxiliary function named Ξ as well. It works on pairs containing a formula and either $+1$ or -1 to denote the polarity.

Definition 5.7 The formal definition of the *formula transformation* Ξ is as follows:

(i) The main function is Ξ: $\begin{array}{rcl} \mathscr{F}(\Sigma) & \longrightarrow & \mathscr{F}(\Sigma) \\ \phi & \longmapsto & \Xi(\phi, 1) \end{array}$.

(ii) The auxiliary function Ξ: $\mathscr{F}(\Sigma) \times \{+1, -1\} \to \mathscr{F}(\Sigma)$ is given recursively:

$$\begin{array}{ll}
\top, +1 \longmapsto \top & \top, -1 \longmapsto \bot \\
\bot, +1 \longmapsto \bot & \bot, -1 \longmapsto \top \\
s \simeq t, +1 \longmapsto \bigwedge (s \not\simeq_1^0 \vee t \not\simeq_0^1) & s \simeq t, -1 \longmapsto \bigwedge (s \not\simeq_1^0 \vee t \not\simeq_1^0) \\
P(\vec{t}), +1 \longmapsto (\vec{t} \notin 01^* \vee P(\vec{t})) & P(\vec{t}), -1 \longmapsto (\vec{t} \notin 01^* \vee \neg P(\vec{t})) \\
\multicolumn{2}{c}{\neg\phi, \pi \longmapsto \Xi(\phi, -\pi)} \\
\phi \wedge \psi, +1 \longmapsto \Xi(\phi, +1) \wedge \Xi(\psi, +1) & \phi \wedge \psi, -1 \longmapsto \Xi(\phi, -1) \vee \Xi(\psi, -1) \\
\phi \vee \psi, +1 \longmapsto \Xi(\phi, +1) \vee \Xi(\psi, +1) & \phi \vee \psi, -1 \longmapsto \Xi(\phi, -1) \wedge \Xi(\psi, -1) \\
\multicolumn{2}{c}{\phi \to \psi, \pi \longmapsto \Xi(\neg\phi \vee \psi, \pi)} \\
\multicolumn{2}{c}{\phi \leftrightarrow \psi, \pi \longmapsto \Xi((\phi \to \psi) \wedge (\psi \to \phi), \pi)} \\
\forall x.\, \phi, +1 \longmapsto \forall x.\, \Xi(\phi, +1) & \forall x.\, \phi, -1 \longmapsto \exists x.\, \Xi(\phi, -1) \\
\exists x.\, \phi, +1 \longmapsto \exists x.\, \Xi(\phi, +1) & \exists x.\, \phi, -1 \longmapsto \forall x.\, \Xi(\phi, -1)
\end{array}$$

Obviously, the transformation results are in negation normal form. Moreover, the transformation is linear as long as no equivalences occur. Regarding the latter, note that they would be resolved during clausification anyway.

Proposition 5.8 We gather some simple properties of the transformation Ξ:

(i) Ξ distributes over conjunctions, disjunctions and quantifications.

(ii) The following formulae have identical images under Ξ:
   (a) $\phi \vee \psi$ and $\neg(\neg\phi \wedge \neg\psi)$,
   (b) $\phi \to \psi$ and $\neg\phi \vee \psi$,
   (c) $\phi \leftrightarrow \psi$ and $(\phi \to \psi) \wedge (\psi \to \phi)$,
   (d) $\exists x.\, \phi$ and $\neg\forall x.\, \neg\phi$.

(iii) The sets $\Xi(\mathscr{F}(\Sigma))$ and $\Xi(\mathscr{F}_0(\Sigma))$ are equal.

(iv) Ξ commutes with substitutions.

Proof:

(i) The property is a consequence of the fact that, for the cases at hand, the polarity is not changed in the recursive call of the auxiliary function. We have $\Xi(\phi \wedge \psi) \equiv \Xi(\phi \wedge \psi, 1) \equiv \Xi(\phi, 1) \wedge \Xi(\psi, 1) \equiv \Xi(\phi) \wedge \Xi(\psi)$ and $\Xi(\forall x.\, \phi) \equiv \Xi(\forall x.\, \phi, 1) \equiv \forall x.\, \Xi(\phi, 1) \equiv \forall x.\, \Xi(\phi)$. The proofs for $\vee$ and $\exists$ are analogous.

(ii) The proofs are simply by unfolding and folding recursion steps of the auxiliary function.
   (a) $\Xi(\phi \vee \psi) \equiv \Xi(\phi, 1) \vee \Xi(\psi, 1) \equiv \Xi(\neg\phi, -1) \vee \Xi(\neg\psi, -1) \equiv \Xi(\neg\phi \wedge \neg\psi, -1) \equiv \Xi(\neg(\neg\phi \wedge \neg\psi))$
   (b) $\Xi(\phi \to \psi) \equiv \Xi(\neg\phi \vee \psi, 1) \equiv \Xi(\neg\phi \vee \psi)$
   (c) $\Xi(\phi \leftrightarrow \psi) \equiv \Xi((\phi \to \psi) \wedge (\psi \to \phi), 1) \equiv \Xi((\phi \to \psi) \wedge (\psi \to \phi))$
   (d) $\Xi(\exists x.\, \phi) \equiv \exists x.\, \Xi(\phi) \equiv \exists x.\, \Xi(\neg\phi, -1) \equiv \Xi(\forall x.\, \neg\phi, -1) \equiv \Xi(\neg\forall x.\, \neg\phi)$

(iii) The inclusion $\Xi(\mathscr{F}(\Sigma)) \supseteq \Xi(\mathscr{F}_0(\Sigma))$ follows from $\mathscr{F}(\Sigma) \supseteq \mathscr{F}_0(\Sigma)$. For the converse inclusion, we show the more general property that for every π and $\phi \in \mathscr{F}(\Sigma)$ there exists a formula $\psi \in \mathscr{F}_0(\Sigma)$ such that $\Xi(\phi, \pi) \equiv \Xi(\psi, \pi)$. The proof is by one induction over the formula structure. To obtain an induction ordering, we interpret formulae as terms and choose a lexicographic path ordering [KL80] based on a precedence $\succ$ such that $\leftrightarrow \succ \to \succ \vee \succ \{\neg, \wedge\}$ and $\exists \succ \{\forall, \neg\}$. Because $\mathscr{F}_0(\Sigma)$ and $\mathscr{F}(\Sigma)$ contain the same atomic formulae, there is nothing to prove for atoms. The same applies to the logical constants $\top$ and $\bot$. If $\phi \equiv \phi_1 \vee \phi_2$, then by Prop. 5.8 (ii) we have $\Xi(\phi) \equiv \Xi(\neg(\neg\phi_1 \wedge \neg\phi_2))$. Since $\neg(\neg\phi_1 \wedge \neg\phi_2)$ is smaller than ϕ we inductively obtain a formula $\psi \in \mathscr{F}_0(\Sigma)$ such that $\Xi(\psi) \equiv \Xi(\neg(\neg\phi_1 \wedge \neg\phi_2)) \equiv \Xi(\phi)$. The cases $\phi \equiv \phi_1 \to \phi_2$, $\phi \equiv \phi_1 \leftrightarrow \phi_2$ and $\phi \equiv \exists x.\, \phi_1$ are treated correspondingly. The following cases remain:
   - $\phi \equiv \neg\phi_1$: Inductively we get a formula $\psi_1 \in \mathscr{F}_0(\Sigma)$ such that $\Xi(\phi_1, -1) \equiv \Xi(\psi_1, -1)$. Then $\Xi(\neg\phi_1) \equiv \Xi(\psi_1, -1) \equiv \Xi(\neg\psi_1)$, which covers $\pi = 1$. For $\pi = -1$ correspondingly $\Xi(\neg\phi_1, -1) \equiv \Xi(\phi_1) \equiv \Xi(\psi_1) \equiv \Xi(\neg\psi_1, -1)$.
   - $\phi \equiv \phi_1 \wedge \phi_2$: Regarding $\pi = 1$ we obtain inductively ψ_1 and ψ_2 such that $\Xi(\phi_i) \equiv \Xi(\psi_i)$ and hence $\Xi(\phi_1 \wedge \phi_2) \equiv \Xi(\phi_1) \wedge \Xi(\phi_2) \equiv \Xi(\psi_1) \wedge \Xi(\psi_2) \equiv \Xi(\psi_1 \wedge \psi_2)$. For $\pi = -1$ analogously $\Xi(\phi_1 \wedge \phi_2, -1) \equiv \Xi(\phi_1, -1) \vee \Xi(\phi_2, -1) \equiv \Xi(\psi_1, -1) \vee \Xi(\psi_2, -1) \equiv \Xi(\psi_1 \wedge \psi_2, -1)$
   - $\phi \equiv \forall x.\, \phi_1$: similarly $\Xi(\forall x.\, \phi_1) \equiv \forall x.\, \Xi(\phi_1) \equiv \forall x.\, \Xi(\psi_1) \equiv \Xi(\forall x.\, \psi_1)$ and $\Xi(\forall x.\, \phi_1, -1) \equiv \exists x.\, \Xi(\phi_1, -1) \equiv \exists x.\, \Xi(\psi_1, -1) \equiv \Xi(\forall x.\, \psi_1, -1)$

(iv) One has to prove by induction over the formula structure that $\Xi(\phi\sigma, \pi) \equiv \Xi(\phi, \pi)\sigma$. For atoms this is obvious from the way they are transformed. Regarding compound formulae, one exploits that both Ξ and σ move more or less homomorphically over connectives and quantifiers. Let us look at the case where $\phi \equiv \forall x.\, \psi$, $\pi = +1$, $x \in \operatorname{dom}\sigma$,

and $\sigma' = \sigma \setminus \{x \mapsto x\sigma\}$. Then $\Xi((\forall x.\, \psi)\sigma, \pi) \equiv \Xi(\forall x.\, \psi\sigma', \pi) \equiv \forall x.\, \Xi(\psi\sigma', \pi) \equiv \forall x.\, \Xi(\psi, \pi)\sigma' \equiv (\forall x.\, \Xi(\psi, \pi))\sigma$. The remaining cases are correspondingly.

□

A useful consequence of Prop. 5.8 (iii) is that a statement on formulae holds for every $\Xi(\phi)$ iff it does for every $\Xi(\psi)$ where ψ is a basic formula.

The transformation Ξ does not distribute over negations. Instead the effect is on the formula level to swap disjunctions and conjunctions as well as universal and existential quantifications. On the literal level, negation swaps between transformations with opposite polarity. To capture this formally, we introduce a syntactical mirroring operation.

Definition 5.9 The *mirroring function* $\_^{\mathrm{M}}\colon \Xi(\mathscr{F}(\Sigma)) \to \Xi(\mathscr{F}(\Sigma))$ is defined by the following recursive transformation schemes, given in order of descending priority:

$$\begin{array}{rll}
\top \leftrightarrows \bot & & \\
\bigwedge (s \not\approx^{0}_{1} \vee\, t \not\approx^{\alpha}_{\beta}) \longmapsto \bigwedge (s \not\approx^{0}_{1} \vee\, t \not\approx^{\beta}_{\alpha}) & \text{if } \binom{\alpha}{\beta} \in \left\{\binom{0}{1}, \binom{1}{0}\right\} \\
\vec{t} \notin 01^* \vee P(\vec{t}) \leftrightarrows \vec{t} \notin 01^* \vee \neg P(\vec{t}) & & \\
\phi \otimes \psi \longmapsto \phi^{\mathrm{M}} \overline{\otimes}\, \psi^{\mathrm{M}} & \text{if } \otimes \in \{\wedge, \vee\} \\
Qx.\, \phi \longmapsto \overline{Q}x.\, \phi^{\mathrm{M}} & \text{if } Q \in \{\forall, \exists\}
\end{array}$$

We will show now that the images of $\_^{\mathrm{M}}$ indeed belong to $\Xi(\mathscr{F}(\Sigma))$, by relating $\_^{\mathrm{M}}$ with negation. This relation also leads to a statement of Ξ related to particular contexts. In the formula $\phi[\psi]_p$ we consider ϕ as a *context in negation normal form that does not end in a negation* if ϕ is in negation normal form and either $p = \lambda$ or $p = q.i$ and the formula $\phi|_q$ is not a negation.

Proposition 5.10 The transformation Ξ is also subject to the following properties:

(i) $\Xi(\neg\phi) \equiv \Xi(\phi)^{\mathrm{M}}$
(ii) Identity under Ξ is closed under contexts.
(iii) Ξ distributes over contexts in negation normal form that do not end in a negation.

Proof:

(i) We proceed by induction on the formula length. Silently we will use a number of trivial identities extending the list given in Prop. 5.8 (ii).
- $\phi \equiv \top$: $\Xi(\neg\top) \equiv \bot \equiv \Xi(\top)^{\mathrm{M}}$

- $\phi \equiv s \simeq t$: $\Xi(\neg\, s \simeq t) \equiv (s \not\approx 0 \vee t \not\approx 0) \wedge (s \not\approx 1 \vee t \not\approx 1) \equiv ((s \not\approx 0 \vee t \not\approx 1) \wedge (s \not\approx 1 \vee t \not\approx 0))^{\mathrm{M}} \equiv \Xi(s \simeq t)^{\mathrm{M}}$
- $\phi \equiv P(\vec{t})$: $\Xi(\neg P(\vec{t})) \equiv \vec{t} \notin 01^* \vee \neg P(\vec{t}) \equiv (\vec{t} \notin 01^* \vee P(\vec{t}))^{\mathrm{M}} \equiv \Xi(P(\vec{t}))^{\mathrm{M}}$
- $\phi \equiv \phi_1 \wedge \phi_2$: $\Xi(\neg(\phi_1 \wedge \phi_2)) \equiv \Xi(\neg\phi_1) \vee \Xi(\neg\phi_2) \equiv \Xi(\phi_1)^{\mathrm{M}} \vee \Xi(\phi_2)^{\mathrm{M}} \equiv (\Xi(\phi_1) \wedge \Xi(\phi_2))^{\mathrm{M}} \equiv \Xi(\phi_1 \wedge \phi_2)^{\mathrm{M}}$
- $\phi \equiv \phi_1 \to \phi_2$: $\Xi(\neg(\phi_1 \to \phi_2)) \equiv \Xi(\phi_1 \wedge \neg\phi_2) \equiv \Xi(\phi_1) \wedge \Xi(\neg\phi_2) \equiv \Xi(\neg\phi_1)^{\mathrm{M}} \wedge \Xi(\phi_2)^{\mathrm{M}} \equiv (\Xi(\neg\phi_1) \vee \Xi(\phi_2))^{\mathrm{M}} \equiv \Xi(\neg\phi_1 \vee \phi_2)^{\mathrm{M}} \equiv \Xi(\phi_1 \to \phi_2)^{\mathrm{M}}$
- $\phi \equiv \forall x.\, \phi_1$: $\Xi(\neg(\forall x.\, \phi_1)) \equiv \Xi(\exists x.\, \neg\phi_1) \equiv \exists x.\, \Xi(\neg\phi_1) \equiv \exists x.\, \Xi(\phi_1)^{\mathrm{M}} \equiv (\forall x.\, \Xi(\phi_1))^{\mathrm{M}} \equiv \Xi(\forall x.\, \phi_1)^{\mathrm{M}}$

The remaining cases are analogous.

(ii) Let $\Xi(\psi) \equiv \Xi(\psi')$, $\phi \equiv \phi[\psi]_p$ and $\phi' \equiv \phi[\psi']_p$. We show via induction on the length of p that this entails $\Xi(\phi) \equiv \Xi(\phi')$. For $p = \lambda$ the statement is trivial. Consider otherwise a position $i.p$, and let $\phi_i \equiv \phi|_i$ and $\phi'_i \equiv \phi'|_i$. Inductively we obtain that $\Xi(\phi_i) \equiv \Xi(\phi'_i)$. By a case split on the shape of ϕ we may conclude that $\Xi(\phi) \equiv \Xi(\phi')$:

- $\phi \equiv \neg\phi_1$: The identity $\Xi(\phi_1) \equiv \Xi(\phi'_1)$ entails $\Xi(\phi_1)^{\mathrm{M}} \equiv \Xi(\phi'_1)^{\mathrm{M}}$, which by Prop. 5.10 (i) is the same as $\Xi(\neg\phi_1) \equiv \Xi(\neg\phi'_1)$.
- $\phi \equiv \phi_1 \wedge \phi_2$: Because of symmetry we can restrict to the case $i = 1$. Then Prop. 5.8 (i) gives $\Xi(\phi_1) \wedge \Xi(\phi_2) \equiv \Xi(\phi'_1) \wedge \Xi(\phi_2) \equiv \Xi(\phi'_1 \wedge \phi_2)$.
- $\phi \equiv \forall x.\, \phi_1$: Similarly we apply Prop. 5.8 (i) and obtain $\Xi(\forall x.\, \phi_1) \equiv \forall x.\, \Xi(\phi_1) \equiv \forall x.\, \Xi(\phi'_1) \equiv \Xi(\forall x.\, \phi'_1)$.

The remaining cases are analogous.

(iii) Consider a formula ϕ in negation normal form and a position p such that either $p = \lambda$, or $p = q.i$ and $\phi|_q$ is not a negation. We will show that this entails $\Xi(\phi[\psi]_p) \equiv \Xi(\phi)[\Xi(\psi)]_p$, by induction on the length of p.

If $p = \lambda$ we are done. Otherwise let us look at a position $i.p$ and a formula $\Xi(\phi[\psi]_{i.p})$ such that $i.p = q.j$ and $\phi|_q$ is not a negation. We will later apply the induction hypothesis onto $\Xi(\phi|_i[\psi]_p)$. This is always possible provided p is empty. Let otherwise $p = p'.j$; then $\phi|_i|_{p'}$ must not be a negation. But since $i.p' = q$ and $\phi|_i|_{p'} \equiv \phi|_q$ is by assumption not a negation, it turns out that the hypothesis can always be applied. Let us now determine the outermost symbol of the formula ϕ. It has the strict subformula $\phi|_{i.p}$ and therefore is not an atom nor a logical constant. Since ϕ is in negation normal form, the symbol must be an element of $\{\neg, \wedge, \vee, \forall, \exists\}$. Assume ϕ were a negation. Then more precisely ϕ, as a negation normal form, were a negative literal, and the subformula position $i.p$ were equal to 1. Hence $q = \lambda$ and $\phi|_q \equiv \phi$ were a negation, in contradiction to our assumption.

Consequently ϕ starts with a connective $\wedge$ or $\vee$, or a quantifier $\forall$ or $\exists$. Note that the transformation Ξ equally distributes over all these symbols. As we shall see, this is all we need to reduce the conjecture to the induction hypothesis. Therefore we only consider the case $\phi \equiv \phi_1 \wedge \phi_2$. Because of symmetry we can restrict to $i = 1$. This leads us to the chain of identities $\Xi((\phi_1 \wedge \phi_2)[\psi]_{1.p}) \equiv \Xi(\phi_1[\psi]_p \wedge \phi_2) \equiv \Xi(\phi_1[\psi]_p) \wedge \Xi(\phi_2) \equiv \Xi(\phi_1)[\Xi(\psi)]_p \wedge \Xi(\phi_2) \equiv (\Xi(\phi_1) \wedge \Xi(\phi_2))[\Xi(\psi)]_{1.p} \equiv \Xi(\phi_1 \wedge \phi_2)[\Xi(\psi)]_{1.p}$

□

5.3.2 Soundness

We now turn to analyzing the effect of the transformation on the interpretation of formulae in the algebra of propositions. To start with, it does not alter the interpretation of literals. Furthermore, negating transformed formulae corresponds to flipping the polarity in the transformation.

Proposition 5.11 We collect some $\mathbb{B}$-equivalences related to Ξ.

(i) Each of the following formulae ϕ is $\mathbb{B}$-equivalent to $\Xi(\phi)$:
(a) $s \simeq t$, (b) $s \not\simeq t$, (c) $P(\vec{t})$, (d) $\neg P(\vec{t})$, (e) $\top$, (f) $\bot$.

(ii) $\neg\, \Xi(\phi, \pi)$ and $\Xi(\phi, -\pi)$ are $\mathbb{B}$-equivalent.

Proof:

(i)(a) $\Xi(s \simeq t)$ expands to $(s \not\simeq 0 \vee t \not\simeq 1) \wedge (s \not\simeq 1 \vee t \not\simeq 0)$. Because of $\mathbb{B} \models \forall x.\, x \simeq 0 \vee x \simeq 1$ we obtain $\mathbb{B}$-equivalence of $\Xi(s \simeq t)$ and $(s \simeq 1 \vee t \simeq 0) \wedge (s \simeq 0 \vee t \simeq 1)$. Distributing $\vee$ over $\wedge$ and removing disjuncts that contradict $\mathbb{B} \models 0 \not\simeq 1$ gives $\mathbb{B}$-equivalence with $(s \simeq 1 \wedge t \simeq 1) \vee (t \simeq 0 \wedge s \simeq 0)$, which in $\mathbb{B}$ means $s \simeq t$. The proof of (i)(b) is symmetrical.

(i)(c) Similarly $\Xi(P(\vec{t})) \equiv \vec{t} \notin 01^* \vee P(\vec{t}) \equiv \bigvee_i (t_i \not\simeq 0 \wedge t_i \not\simeq 1) \vee P(\vec{t})$ is, by $\mathbb{B} \models \forall x.\, x \simeq 0 \vee x \simeq 1$, just $\mathbb{B}$-equivalent to $\bigvee_i (t_i \simeq 1 \wedge t_i \simeq 0) \vee P(\vec{t})$, which reduces to $P(\vec{t})$ because of $\mathbb{B} \models 0 \not\simeq 1$. The argument for (i)(d) is the same.

(i)(e) This follows from $\Xi(\top) \equiv \top$. A similar argument applies to (i)(f).

(ii) From $\neg\, \Xi(\phi, 1) \leftrightarrow \Xi(\phi, -1)$ we obtain by negation of both sides that $\Xi(\phi, 1) \leftrightarrow \neg\, \Xi(\phi, -1)$, i. e. the statement with $\pi = 1$ is equivalent to that with $\pi = -1$. We now proceed by induction over the formula structure. Proposition 5.8 (iii) restricts the number of cases we have to consider.

- $\phi \equiv \top$: The left expression evaluates to $\neg\top$ and the right one to $\bot$. The case $\phi \equiv \bot$ is symmetrical.

- $\phi \equiv s \simeq t$: Plugging the two statements Prop. 5.11 (i)(a) and (b) together gives $\mathbb{B}$-equivalence of $\Xi(s \simeq t)$ and $\neg\,\Xi(s \not\simeq t)$, which expands to $\mathbb{B}$-equivalence of $\Xi(s \simeq t, 1)$ and $\neg\,\Xi(s \simeq t, -1)$.
- $\phi \equiv P(\vec{t})$: correspondingly to previous case
- $\phi \equiv \neg\psi$: $\neg\,\Xi(\neg\psi, 1)$ is identical to $\neg\,\Xi(\psi, -1)$ and inductively equivalent under $\mathcal{B}_\mu$ to $\Xi(\psi, 1) \equiv \Xi(\neg\psi, -1)$.
- $\phi \equiv \psi_1 \wedge \psi_2$: $\neg\,\Xi(\psi_1 \wedge \psi_2, 1)$ is equivalent to $\neg\,\Xi(\psi_1, 1) \vee \neg\,\Xi(\psi_2, 1)$ and inductively equivalent under $\mathcal{B}_\mu$ to $\Xi(\psi_1, -1) \vee \Xi(\psi_2, -1) \equiv \Xi(\psi_1 \wedge \psi_2, -1)$.
- $\phi \equiv \forall x.\, \psi$: $\neg\,\Xi(\forall x.\, \psi, 1)$ is equivalent to $\exists x.\, \neg\,\Xi(\psi, 1)$ and inductively equivalent under $\mathcal{B}_\mu$ to $\exists x.\, \Xi(\psi, -1) \equiv \Xi(\forall x.\, \psi, -1)$.

□

Next we show that the transformation Ξ preserves $\mathbb{B}$-equivalence.

Proposition 5.12 For every formula ϕ we have $\mathbb{B} \models \Xi(\phi) \leftrightarrow \phi$.

Proof: We show by induction over the formula length the equivalence of $\Xi(\phi)$ and ϕ in any $\mathbb{B}$-model $\mathcal{B}_\mu$. Atoms and the logical constants $\top$ and $\bot$ are covered by Prop. 5.11 (i).

- $\phi \equiv \neg\psi$: $\mathcal{B}_\mu \models \Xi(\neg\psi)$ iff $\mathcal{B}_\mu \models \Xi(\psi, -1)$ iff, by Prop. 5.11 (ii), $\mathcal{B}_\mu \not\models \Xi(\psi, 1)$ iff, inductively, $\mathcal{B}_\mu \not\models \psi$
- $\phi \equiv \psi_1 \wedge \psi_2$: [$\phi \equiv \psi_1 \vee \psi_2$:] $\mathcal{B}_\mu \models \Xi(\psi_1 \wedge \psi_2)$ [$\mathcal{B}_\mu \models \Xi(\psi_1 \vee \psi_2)$] iff, by Prop. 5.8 (i), $\mathcal{B}_\mu \models \Xi(\psi_1)$ and [or] $\mathcal{B}_\mu \models \Xi(\psi_2)$ iff, inductively, $\mathcal{B}_\mu \models \psi$ and [or] $\mathcal{B}_\mu \models \psi$
- $\phi \equiv \phi_1 \rightarrow \phi_2$: $\mathcal{B}_\mu \models \Xi(\phi_1 \rightarrow \phi_2)$ iff $\mathcal{B}_\mu \models \Xi(\neg\phi_1)$ or $\mathcal{B}_\mu \models \Xi(\phi_2)$ iff, inductively, $\mathcal{B}_\mu \models \neg\phi_1$ or $\mathcal{B}_\mu \models \phi_2$ iff $\mathcal{B}_\mu \models \phi_1 \rightarrow \phi_2$
- $\phi \equiv \phi_1 \leftrightarrow \phi_2$: $\mathcal{B}_\mu \models \Xi(\phi_1 \leftrightarrow \phi_2)$ iff $\mathcal{B}_\mu \models \Xi(\neg\phi_1)$ or $\mathcal{B}_\mu \models \Xi(\phi_2)$, and $\mathcal{B}_\mu \models \Xi(\neg\phi_2)$ or $\mathcal{B}_\mu \models \Xi(\phi_1)$, iff, inductively, $\mathcal{B}_\mu \models \neg\phi_1$ or $\mathcal{B}_\mu \models \phi_2$, and $\mathcal{B}_\mu \models \neg\phi_2$ or $\mathcal{B}_\mu \models \phi_1$, iff $\mathcal{B}_\mu \models \phi_1 \leftrightarrow \phi_1$
- $\phi \equiv \forall x.\, \psi$: [$\phi \equiv \exists x.\, \psi$:] $\mathcal{B}_\mu \models \Xi(\forall x.\, \psi)$ [$\mathcal{B}_\mu \models \Xi(\exists x.\, \psi)$] iff $\mathcal{B}_{\mu^x_\alpha} \models \Xi(\psi)$ for every α [for one α] iff, inductively, $\mathcal{B}_{\mu^x_\alpha} \models \psi$ for every α [for one α]

□

Lemma 5.13 Ξ is a sound formula transformation from $\mathbb{B}$ to $\mathbb{D}$.

Proof: Let $\mathbb{D} \models \Xi(\phi)$ for some ϕ. Then $\mathrm{Ded}(\mathbb{B}) \supseteq \mathrm{Ded}(\mathbb{D})$ (Ass. 5.6) gives $\mathbb{B} \models \Xi(\phi)$, and by Prop. 5.12 we get $\mathbb{B} \models \phi$. □

5.3.3 Completeness on Clauses

We now turn to analyzing the completeness of Ξ: For which formulae ϕ does $\mathbb{B} \models \phi$ imply $\mathbb{D} \models \Xi(\phi)$? An important link between $\mathbb{B}$ and $\mathbb{D}$ has been established in Lem. 5.5: Every model of $\mathbb{D}$, which by Ass. 5.6 is a model of $\mathbb{C}$ as well, induces a model of $\mathbb{B}$ which partially coincides.

Proposition 5.14 Ξ is a complete transformation from $\mathbb{B}$ to $\mathbb{D}$ on Σ-clauses and clause normal forms.

Proof: Let us first study a clause $C \equiv L_1 \vee \ldots \vee L_n$ where each L_i denotes a literal and A_i is the corresponding atom. Assume that $\mathbb{B} \models C$. We have to show that this entails $\mathbb{D} \models \Xi(C)$. To this end, take any $\mathbb{D}$-model $\mathcal{A}_\mu$. Because the empty clause is unsatisfiable we have $n > 0$. Now there are the following two cases:

- There exists an i such that (a) $A_i \equiv t_1 \simeq t_2$ or (b) $A_i \equiv P(\vec{t})$, and in each case $\mathcal{A}_\mu \models \vec{t} \not\in 01^*$. Regarding (a), we may by symmetry assume without loss of generality that $\mathcal{A}_\mu \models t_1 \not\simeq 0 \wedge t_1 \not\simeq 1$. The expression $\Xi(L_i)$ expands to $(t_1 \not\simeq 0 \vee t_2 \not\simeq \alpha) \wedge (t_1 \not\simeq 1 \vee t_2 \not\simeq \beta)$ where (α, β) is $(1, 0)$ or $(0, 1)$. Hence we here have $\mathcal{A}_\mu \models \Xi(L_i)$. As to (b), since $\Xi(L_i) \equiv \vec{t} \not\in 01^* \vee L_i$ we evidently have $\mathcal{A}_\mu \models \Xi(L_i)$, too. In both cases, from $\mathcal{A}_\mu \models \Xi(L_i)$ we obtain $\mathcal{A}_\mu \models \Xi(L_1) \vee \ldots \vee \Xi(L_n)$ and $\mathcal{A}_\mu \models \Xi(C)$.
- Otherwise every atom $A_i \equiv t_1 \simeq t_2$ or $A_i \equiv P(\vec{t})$ satisfies $\mathcal{A}_\mu \models t_j \simeq 0 \vee t_j \simeq 1$ for every j, such that C belongs to $\mathscr{Q}[\mathcal{A}_\mu]$. By Lem. 5.5 there exists a $\mathbb{B}$-model $\mathcal{B}$ such that $\mathcal{A}_\mu = \mathcal{B}_\nu$ on $\mathscr{Q}[\mathcal{A}_\mu]$. Now $\mathbb{B} \models C$ implies $\mathcal{B}_\nu \models C$. Since $\mathbb{B} \models \Xi(C) \leftrightarrow C$ (Prop. 5.12), we also have $\mathcal{B}_\nu \models \Xi(C)$. By construction of Ξ, the formula $\Xi(C)$ is in $\mathscr{Q}[\mathcal{A}_\mu]$; so Lem. 5.5 applies once more and gives $\mathcal{A}_\mu \models \Xi(C)$.

Hence Ξ is complete on clauses. We now turn to formulae in clause normal form. Let $\forall\vec{x}$ abbreviate a sequence of quantifications $\forall x_1. \ldots \forall x_k$ and consider a $\mathbb{B}$-valid formula $\forall\vec{x}. \bigwedge_i C_i$ where every C_i is a clause. If the conjunction is empty, i. e. $\top$, then $\mathbb{D} \models \Xi(\top)$ holds because $\Xi(\top) \equiv \top$. Otherwise we have $\mathbb{B} \models \forall\vec{x}. C_i$ and $\mathbb{B} \models C_i$ for every i. By completeness on clauses we get $\mathbb{D} \models \Xi(C_i)$, therefore $\mathbb{D} \models \Xi(\bigwedge_i C_i)$ and $\mathbb{D} \models \Xi(\forall\vec{x}. \bigwedge_i C_i)$. □

5.3.4 Tools for the Level of Formulae

The completeness proof will be extended to the level of formulae. For doing so, we will relate the formula transformation with a CNF computation. The latter will be described by a rewrite system. As a preparatory step, here we define formulae rewriting.

Definition 5.15 We sketch how to introduce a notion of rewriting on formulae.

(i) A *constrained formula pair* is a pair of formulae which are built over meta variables $\phi, \psi, s, t, x, y, \ldots$ It may be equipped with side conditions on the meta variables like in the following:

$$(\forall x.\, \phi) \vee \psi \Longrightarrow \forall y.\, \phi\{x \mapsto y\} \vee \psi \quad \text{if } y \notin \text{free}(\psi, \forall x.\, \phi)$$

(ii) We get an *instance* of a constrained formula pair whenever we replace all meta variables by object-level formulae, terms etc. such that the side conditions hold.

(iii) A set Φ of constrained formula pairs *induces* a reduction system $\Longrightarrow$ on formulae if $\Longrightarrow$ is the smallest such relation that contains every instance of the elements of Φ and is closed under formula contexts.

The next proposition describes how to check whether a reduction system on formulae preserves $\mathcal{T}$-equivalence. It essentially boils down to the fact that $\mathcal{T}$-equivalence is closed under formula contexts and is included for the sake of completeness.

Proposition 5.16 Consider a theory $\mathcal{T}$ and a set Φ of constrained formula pairs with induced relation $\Longrightarrow$. Then the following are equivalent:

(i) $\Longrightarrow^*$ preserves $\mathcal{T}$-equivalence.

(ii) For every instance (ϕ, ψ) of a constrained formula pair of Φ, the object-level formulae ϕ and ψ are $\mathcal{T}$-equivalent.

Proof: The implication (i)$\rightarrow$(ii) is obvious. For the converse direction, we will show that the condition given in (ii) entails that $\Longrightarrow$ preserves $\mathcal{T}$-equivalence. From this one obtains (i) by induction on the length of the $\Longrightarrow$-chain.

If $\phi \Longrightarrow \phi'$, then by construction of $\Longrightarrow$ this can be written as $\phi[\psi_1]_p \Longrightarrow \phi[\psi_2]_p$ where (ψ_1, ψ_2) is an instance of a constrained formula pair of Φ. By (ii) the formulae ψ_1 and ψ_2 are $\mathcal{T}$-equivalent. We proceed by induction on the length of p. If p is empty we are done. Let otherwise $p = i.q$, $\psi = \phi|_i$ and $\psi' = \phi'|_i$. Since $\psi \equiv (\phi|_i)[\psi_1]_q \Longrightarrow (\phi|_i)[\psi_2]_q \equiv \psi'$, we get inductively that ψ and ψ' are $\mathcal{T}$-equivalent. It remains to show that this implies $\mathcal{T}$-equivalence of $\phi[\psi]_i$ and $\phi[\psi']_i$, which is done by a case split on the outermost symbol of ϕ. In the case of $\forall$ the formulae at hand are $\forall x.\, \psi$ and $\forall x.\, \psi'$. For an arbitrary $\mathcal{T}$-model $\mathcal{A}_\mu$ we have $\mathcal{A}_\mu(\forall x.\, \psi)$ iff $\mathcal{A}_{\mu^x_\alpha}(\psi)$ for every α iff, by $\mathcal{T}$-equivalence of ψ and ψ', $\mathcal{A}_{\mu^x_\alpha}(\psi')$ for every α iff $\mathcal{A}_\mu(\forall x.\, \psi')$. The remaining cases are analogous. $\square$

Next we drop a technical proposition that proves implication compatible with contexts in the case of negation normal forms; it will be applied in the subsequent subsection.

Proposition 5.17 Consider a Σ-theory $\mathcal{T}$ and a formula $\phi[\psi]_p$ where ϕ is a context in negation normal form that does not end in a negation. Then $\mathcal{T} \models \psi \to \psi'$ implies $\mathcal{T} \models \phi[\psi]_p \to \phi[\psi']_p$.

Proof: Assume that ϕ is in negation normal form and either $p = \lambda$, or $p = q.i$ and the formula $\phi|_q$ is not a negation. Let furthermore $\mathcal{T} \models \psi \to \psi'$. We will show by induction on the length of p that for every $\mathcal{T}$-model $\mathcal{A}_\mu \models \phi[\psi]_p$ entails $\mathcal{A}_\mu \models \phi[\psi']_p$.

If $p = \lambda$ we are done because $\mathcal{A}_\mu$ is a model of $\mathcal{T}$. Otherwise let us look at a position $i.p$ and a context $\phi[\psi]_{i.p}$ such that $i.p = q.j$ and the formula $\phi|_q$ is not a negation. We will later apply the induction hypothesis onto $\mathcal{A}_\mu \models \phi|_i[\psi]_p$. This is always possible provided p is empty. Let otherwise $p = p'.j$; then $\phi|_i|_{p'}$ must not be a negation. But since $i.p' = q$ and $\phi|_i|_{p'} \equiv \phi|_q$ is by assumption not a negation, it turns out that the hypothesis can always be applied.

Let us now determine the outermost symbol of the formula ϕ. By assumption $\phi|_{i.p}$ is a formula. So ϕ has a strict subformula and therefore cannot be an atom or a logical constant. Since ϕ is in negation normal form, the symbol must be an element of $\{\neg, \wedge, \vee, \forall, \exists\}$. Assume ϕ were a negation. Then more precisely ϕ, as a negation normal form, were a negative literal, and the subformula position $i.p$ were equal to 1. Hence $q = \lambda$ and $\phi|_q \equiv \phi$ were a negation, in contradiction to our assumption.

We proceed by case split on the structure of ϕ.

- $\phi \equiv \phi_1 \wedge \phi_2$: By symmetry we only need to consider the case $i = 1$. Now $\mathcal{A}_\mu \models (\phi_1 \wedge \phi_2)[\psi]_{1.p}$ implies $\mathcal{A}_\mu \models \phi_1[\psi]_p$. We get inductively $\mathcal{A}_\mu \models \phi_1[\psi']_p$ and hence $\mathcal{A}_\mu \models (\phi_1 \wedge \phi_2)[\psi']_{1.p}$.
- $\phi \equiv \phi_1 \vee \phi_2$ and $i = 1$: If $\mathcal{A}_\mu \models \phi_2$ we are done. Otherwise we continue inductively like in the previous case.
- $\phi \equiv \forall x.\, \phi_2$: Here $\mathcal{A}_\mu \models (\forall x.\, \phi_2)[\psi]_{2.p}$ entails $\mathcal{A}_{\mu^x_\alpha} \models \phi_2[\psi]_p$ for every α, which inductively leads to $\mathcal{A}_{\mu^x_\alpha} \models \phi_2[\psi']_p$ for every α and hence to $\mathcal{A}_\mu \models (\forall x.\, \phi_2)[\psi']_{2.p}$.

□

5.3.5 Completeness on Formulae

We will now reduce the completeness of Ξ on arbitrarily formulae to that on clause normal forms. To this end, we introduce a CNF transformation

which preserves $\mathbb{B}$-equivalence, and show for every transformation step that completeness of Ξ on the outcome entails completeness on the input. Our presentation of the clausification adapts the one given in [NW01]. Existential quantifiers will not be eliminated via Skolemization, but by the $\mathbb{B}$-equivalence of $\exists x.\, \phi$ and $\phi\{x \mapsto 0\} \vee \phi\{x \mapsto 1\}$, in order to not extend the signature for Lem. 5.5. This may blow up formulae, but the CNF transformation is only needed to establish completeness, not as a practical reasoning device.

Definition 5.18 The relation $\Longrightarrow$ on formulae is defined as $\Longrightarrow = \Longrightarrow_1^! \circ \Longrightarrow_2^!$ where $\Longrightarrow_1$ and $\Longrightarrow_2$ are induced as follows:

(i) Negation normal form:

$$\begin{aligned}
\neg\top &\Longrightarrow_1 \bot & \qquad \phi \to \psi &\Longrightarrow_1 \neg\phi \vee \psi \\
\neg\bot &\Longrightarrow_1 \top & \qquad \phi \leftrightarrow \psi &\Longrightarrow_1 (\phi \to \psi) \wedge (\psi \to \phi) \\
\neg\neg\phi &\Longrightarrow_1 \phi & \qquad \neg\forall x.\, \phi &\Longrightarrow_1 \exists x.\, \neg\phi \\
\neg(\phi \wedge \psi) &\Longrightarrow_1 \neg\phi \vee \neg\psi & \qquad \neg\exists x.\, \phi &\Longrightarrow_1 \forall x.\, \neg\phi \\
\neg(\phi \vee \psi) &\Longrightarrow_1 \neg\phi \wedge \neg\psi
\end{aligned}$$

(ii) $\exists$-elimination, $\forall$-prenexing, $\vee$-distribution and $\top$-$\bot$-simplification:

$$\begin{aligned}
\exists x.\, \phi &\Longrightarrow_2 \phi\{x \mapsto 0\} \vee \phi\{x \mapsto 1\} \\
(\forall x.\, \phi) \otimes \psi &\Longrightarrow_2 \forall y.\, \phi\{x \mapsto y\} \otimes \psi \qquad \text{for } \otimes \in \{\wedge, \vee\} \text{ and } y \notin \mathrm{free}(\psi, \forall x.\, \phi) \\
\phi \vee (\psi_1 \wedge \psi_2) &\Longrightarrow_2 (\phi \vee \psi_1) \wedge (\phi \vee \psi_2) \\
\top \vee \phi &\Longrightarrow_2 \top \\
\top \wedge \phi &\Longrightarrow_2 \phi \\
\bot \vee \phi &\Longrightarrow_2 \phi
\end{aligned}$$

plus the rules symmetrical in $\otimes$ respectively $\vee$ or $\wedge$

Proposition 5.19 We gather some properties of $\Longrightarrow$.

(i) The relation $\Longrightarrow$ is left-total.

(ii) It preserves equivalence on quantifier-free formulae, and $\mathbb{B}$-equivalence on arbitrary formulae.

(iii) If $\phi \Longrightarrow \psi$, then ψ is a clause normal form.

Proof:

(i) If we can show that the relations $\Longrightarrow_1$ and $\Longrightarrow_2$ terminate, then by Rem. 2.19 every formula has a $\Longrightarrow_1$-normal form, and every $\Longrightarrow_1$-normal form has a $\Longrightarrow_2$-normal form.
Let us start with $\Longrightarrow_1$. We interpret formulae as terms and choose a lexicographic path ordering $>$ (cf. [KL80]) based on a precedence $\succ$ such that $\leftrightarrow \succ \to \succ \neg \succ \{\vee, \wedge, \forall, \exists, \top, \bot\}$. Then in each of the constrained pairs of $\Longrightarrow_1$ the left-hand side is greater than the right-hand side. Since

$>$ is closed under contexts and instantiations we obtain $\Longrightarrow_1^+ \subseteq >$ and hence termination.

For $\Longrightarrow_2$ we refine the interpretation approach such that we consider not the constrained pairs, but all their instances. These are interpreted as terms again, but over a sort F for formulae and a sort T for the original Σ-terms. Let $>$ denote the recursive path ordering induced by a terminating quasi-ordering $\succsim$ on $\mathscr{F}$ (see for example [Der87, p. 94, 98]) such that $\exists \succ \vee \succ \wedge \succ \forall$, and that all symbols in $\mathscr{V} \cup \{0; 1\}$ are equivalent. Thus we get $\phi \approx \phi\{x \mapsto y\} \approx \phi\{x \mapsto 1\}$. Hence, in every instance of a constrained pair of $\Longrightarrow_2$ the left-hand side is greater with respect to $>$ than the right-hand side. Since $>$ is closed under contexts, the relation $\Longrightarrow_2$ is terminating.

(ii) Because of Prop. 5.16 we only need to show that for every instance (ϕ, ψ) of a constrained formula pair of $\Longrightarrow_1$ or $\Longrightarrow_2$, the formulae ϕ and ψ are equivalent, or at least $\mathbb{B}$-equivalent if quantifiers are present. Regarding the pairs in Def. 5.18 (i), for every pair left-hand and right-hand side are evidently equivalent, in the cases without quantifiers even propositionally. The latter also holds for the last four pairs of Def. 5.18 (ii), whereas the first is justified in the theory $\mathbb{B}$ only. Let us finally have a closer look at $(\forall x.\, \phi) \otimes \psi \Longrightarrow_2 \forall y.\, \phi\{x \mapsto y\} \otimes \psi$. Since y is not free in $\forall x.\, \phi$, the formulae $\forall x.\, \phi$ and $\forall y.\, \phi\{x \mapsto y\}$ differ from each other only in the choice of quantified variable and are therefore equivalent. Because y is not free in ψ we have $\psi \models\!\!\mid \forall y.\, \psi$. Finally $(\forall y.\, \phi\{x \mapsto y\}) \otimes (\forall y.\, \psi)$ and $\forall y.\, \phi\{x \mapsto y\} \otimes \psi$ are equivalent in case $\otimes = \wedge$ because universal quantifications distribute over conjunctions, and in case $\otimes = \vee$ because y is not free in ψ.

(iii) Let $\phi \Longrightarrow_1^! \phi' \Longrightarrow_2^! \psi$. If ϕ' contained an implication, an equivalence or a negation not in front of an atom, then one of the rules in Def. 5.18 (i) were applicable, but ϕ' is in $\Longrightarrow_1$-normal form. Since the rules in Def. 5.18 (ii) introduce neither equivalences nor implications and since they do not manipulate negations it is clear that ψ is in negation normal form as well. Moreover ψ contains neither existential quantifiers nor conjunctions under disjunctions, nor universal quantifiers under Boolean connectives. The logical constant $\bot$ occurs within conjunctions only, i. e. not as a strict subclause of some clause, whereas $\top$ only occurs directly below the universal quantifier prefix, denoting the empty conjunction.

□

The following two propositions are at the heart of our completeness proof. They allow to trace $\mathbb{D}$-validity of Ξ-images backwards from clause normal

forms to arbitrary formulae.

Proposition 5.20 If $\phi \Longrightarrow_1 \psi$ and $\mathbb{D} \models \Xi(\psi)$, then $\mathbb{D} \models \Xi(\phi)$.

Proof: Let us study top-level reductions first, i. e. (ϕ, ψ) is an instance of a constrained pair of Def. 5.18 (i). Then by the shape of $\Longrightarrow_1$ and by definition of Ξ we always get that $\Xi(\phi) \equiv \Xi(\psi)$, as exercised in the proof of Prop. 5.8 (ii) for a number of formulae. Hence $\mathbb{D} \models \Xi(\psi)$ is literally the same as $\mathbb{D} \models \Xi(\phi)$.

We now turn to reductions $\phi \equiv \phi[\psi]_p \Longrightarrow_1 \phi[\psi']_p \equiv \phi'$ where the pair instance is (ψ, ψ') and therefore $\Xi(\psi) \equiv \Xi(\psi')$ as discussed. By Prop. 5.10 (ii) this entails identity of the formulae $\Xi(\phi[\psi]_p)$ and $\Xi(\phi[\psi']_p)$. □

Proposition 5.21 If $\phi \Longrightarrow_2 \psi$ and $\mathbb{D} \models \Xi(\psi)$, then $\mathbb{D} \models \Xi(\phi)$ provided ϕ is in negation normal form.

Proof: We start again with top-level reductions, namely instances (ϕ, ψ) of constrained pairs of Def. 5.18 (ii).

- $\phi \equiv \phi_1 \vee (\phi_2 \wedge \phi_3) \Longrightarrow_2 (\phi_1 \vee \phi_2) \wedge (\phi_1 \vee \phi_3) \equiv \psi$: The images of ϕ and ψ under Ξ are equivalent because Ξ distributes over $\wedge$ and $\vee$. In detail $\Xi(\phi) \equiv \Xi(\phi_1 \vee (\phi_2 \wedge \phi_3)) \equiv \Xi(\phi_1) \vee (\Xi(\phi_2) \wedge \Xi(\phi_3)) \models\!\!\models (\Xi(\phi_1) \vee \Xi(\phi_2)) \wedge (\Xi(\phi_1) \vee \Xi(\phi_3)) \equiv \Xi((\phi_1 \vee \phi_2) \wedge (\phi_1 \vee \phi_3)) \equiv \Xi(\psi)$ gives $\Xi(\phi) \models\!\!\models \Xi(\psi)$. Hence $\mathbb{D} \models \Xi(\psi)$ implies $\mathbb{D} \models \Xi(\phi)$.
- $\phi \equiv \top \vee \phi' \Longrightarrow_2 \top \equiv \psi$, $\phi \equiv \top \wedge \psi \Longrightarrow_2 \psi$, $\phi \equiv \bot \vee \psi \Longrightarrow_2 \psi$: Similar to the last item, in each case the Ξ-images of ϕ and ψ are equivalent, because Ξ distributes over $\wedge$ and $\vee$ and does not move $\top$ and $\bot$.
- $\phi \equiv (\forall x.\, \phi_1) \otimes \phi_2 \Longrightarrow_2 \forall y.\, \phi_1\{x \mapsto y\} \otimes \phi_2 \equiv \psi$: Because Ξ distributes over $\otimes$ and $\forall$ we get $\Xi(\phi) \equiv (\forall x.\, \Xi(\phi_1)) \otimes \Xi(\phi_2)$. Renaming bound variables into fresh ones preserves equivalence; so the last expression is equivalent to $(\forall y.\, \Xi(\phi_1)\{x \mapsto y\}) \otimes \Xi(\phi_2)$. As discussed in the proof of Prop. 5.19 (ii), this is equivalent to $\forall y.\, \Xi(\phi_1)\{x \mapsto y\} \otimes \Xi(\phi_2)$: The quantifier can safely be moved out because y is not free in ϕ_2 and hence not in $\Xi(\phi_2)$. By Prop. 5.8 (iv) this is identical to $\forall y.\, \Xi(\phi_1\{x \mapsto y\}) \otimes \Xi(\phi_2) \equiv \Xi(\forall y.\, \phi_1\{x \mapsto y\} \otimes \phi_2) \equiv \Xi(\psi)$. So we have $\Xi(\phi) \models\!\!\models \Xi(\psi)$ like in the previous case.
- $\phi \equiv \exists x.\, \phi' \Longrightarrow_2 \phi'\{x \mapsto 0\} \vee \phi'\{x \mapsto 1\} \equiv \psi$: Consider an arbitrary $\mathbb{D}$-model $\mathcal{A}_\mu$. Since $\mathbb{D} \models \Xi(\psi)$, let us assume without loss of generality that $\mathcal{A}_\mu \models \Xi(\phi'\{x \mapsto 0\})$. By Prop. 5.8 (iv) this can be restated as $\mathcal{A}_\mu \models \Xi(\phi')\{x \mapsto 0\}$. Moving the substitution into the assignment μ we get $\mathcal{A}_{\mu^x_{0_\mathcal{A}}} \models \Xi(\phi')$. Hence we have $\mathcal{A}_\mu \models \exists x.\, \Xi(\phi')$ as well as $\mathcal{A}_\mu \models \Xi(\exists x.\, \phi')$, which is $\mathcal{A}_\mu \models \Xi(\phi)$.

Let us now come to reductions $\phi \equiv \phi[\psi]_p \Longrightarrow_2 \phi[\psi']_p \equiv \phi'$ where the pair instance is (ψ, ψ') and therefore $\mathbb{D} \models \Xi(\psi') \to \Xi(\psi)$ as shown. The structure of $\Longrightarrow_2$ and the negation normal form requirement on ϕ imply that in $\phi[\psi]_p$ the formula ϕ is a context in negation normal that does not end in a negation. Hence by Prop. 5.10 (iii) we obtain that $\Xi(\phi[\psi]_p) \equiv \Xi(\phi)[\Xi(\psi)]_p$ and $\Xi(\phi[\psi']_p) \equiv \Xi(\phi)[\Xi(\psi')]_p$. By construction of Ξ the context condition also holds for $\Xi(\phi)$ in $\Xi(\phi)[\Xi(\psi')]_p$. Therefore we can apply Prop. 5.17 onto $\mathbb{D} \models \Xi(\psi') \to \Xi(\psi)$ and obtain $\mathbb{D} \models \Xi(\phi)[\Xi(\psi')]_p \to \Xi(\phi)[\Xi(\psi)]_p$, which is the same as $\mathbb{D} \models \Xi(\phi[\psi']_p) \to \Xi(\phi[\psi]_p)$ and $\mathbb{D} \models \Xi(\phi') \to \Xi(\phi)$. Consequently $\mathbb{D} \models \Xi(\phi)$ is true. □

Lemma 5.22 Ξ is a complete formula transformation from $\mathbb{B}$ to $\mathbb{D}$.

Proof: Consider an arbitrary formula ϕ such that $\mathbb{B} \models \phi$. Then by Prop. 5.19 there exists a formula ψ in clause normal form such that $\phi \Longrightarrow \psi$ and $\mathbb{B} \models \psi$. Because of Prop. 5.14 we have $\mathbb{D} \models \Xi(\psi)$. We can now decompose $\phi \Longrightarrow \psi$ into a chain $\phi \equiv \phi_0 \Longrightarrow_1 \ldots \Longrightarrow_1 \phi_m \equiv \psi_0 \Longrightarrow_2 \ldots \Longrightarrow_2 \psi_n \equiv \psi$. As discussed in the proof of Prop. 5.19 (iii), the formula ψ_0 is in negation normal form, and by construction of $\Longrightarrow_2$ this property carries over to the formulae ψ_{i+1}. Using Prop. 5.20 and Prop. 5.21 we conclude backwards from $\mathbb{D} \models \Xi(\psi)$ to $\mathbb{D} \models \Xi(\phi)$, by induction on the length of the derivations. □

Theorem 5.23 Ξ is a correct formula transformation from $\mathbb{B}$ to $\mathbb{D}$.

Proof: We finally combine Lem. 5.13 and Lem. 5.22. □

Note that the transformation leaves both universal and existential quantifiers as such. In particular no Skolemization is needed for completeness on arbitrary formulae. Otherwise we would have had to add an axiom $f(\vec{x}) \simeq 0 \vee f(\vec{x}) \simeq 1 \to \vec{x} \in 01^*$ for every Skolem function f.

5.4 A Clausal Approximation

5.4.1 Introducing the Approximation

We will now give a concrete instance for the stronger approximation $\mathbb{D}$. The aim is to enrich $\mathbb{C}$ with $\mathbb{B}$-valid simplifying equations as long as the outcome can still finitely be saturated up to redundancy, using simple redundancy criteria like subsumption. To start with, let us recall how we have approximated the algebra of propositions.

Definition 5.2 The following Σ-formulae, universally closed, define the theory $\mathbb{C}$:

$$
\begin{array}{ll}
0 \not\simeq 1 & \\
\overline{x} \simeq 0 \to x \simeq 1 & \overline{x} \simeq 1 \to x \simeq 0 \\
x \cdot y \simeq 0 \to x \simeq 0 \vee y \simeq 0 & x \cdot y \simeq 1 \to x \simeq 1 \wedge y \simeq 1 \\
x + y \simeq 0 \to x \simeq 0 \wedge y \simeq 0 & x + y \simeq 1 \to x \simeq 1 \vee y \simeq 1 \\
x \oplus y \simeq 0 \to (x \simeq 0 \vee y \simeq 1) \wedge (x \simeq 1 \vee y \simeq 0) & \\
x \oplus y \simeq 1 \to (x \simeq 0 \vee y \simeq 0) \wedge (x \simeq 1 \vee y \simeq 1) & \\
x \ominus y \simeq 0 \to (x \simeq 0 \vee y \simeq 0) \wedge (x \simeq 1 \vee y \simeq 1) & \\
x \ominus y \simeq 1 \to (x \simeq 0 \vee y \simeq 1) \wedge (x \simeq 1 \vee y \simeq 0) & \\
f(\vec{x}) \simeq 0 \vee f(\vec{x}) \simeq 1 \to \bigwedge_i (x_i \simeq 0 \vee x_i \simeq 1) \text{ for every free } f \in \mathscr{F} &
\end{array}
$$

In order to make this theory suitable as an input to the superposition calculus, we turn $\mathbb{C}$ into a clausal presentation, and add some simplifying equations: For every Boolean operator op we describe what $op(\vec{x})$ evaluates to in case one of the arguments is instantiated to 0 or 1.

Definition 5.24 The theory $\mathbb{D}$ consists of two parts. We consider the clauses as variable disjoint and have dropped the universal quantifiers.

$$
\begin{array}{lll}
1 \not\simeq 0 & & \left.\begin{array}{l} \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{array}\right\} \mathbb{D}_1\text{: clausification of theory } \mathbb{C} \\
\overline{x} \simeq 0 \to x \simeq 1 & \overline{x} \simeq 1 \to x \simeq 0 & \\
x \cdot y \simeq 0 \to x \simeq 0 \vee y \simeq 0 & & \\
x \cdot y \simeq 1 \to x \simeq 1 & x \cdot y \simeq 1 \to y \simeq 1 & \\
x + y \simeq 0 \to x \simeq 0 & x + y \simeq 0 \to y \simeq 0 & \\
x + y \simeq 1 \to x \simeq 1 \vee y \simeq 1 & & \\
x \oplus y \simeq 0 \to x \simeq 0 \vee y \simeq 1 & x \oplus y \simeq 0 \to x \simeq 1 \vee y \simeq 0 & \\
x \oplus y \simeq 1 \to x \simeq 0 \vee y \simeq 0 & x \oplus y \simeq 1 \to x \simeq 1 \vee y \simeq 1 & \\
x \ominus y \simeq 0 \to x \simeq 0 \vee y \simeq 0 & x \ominus y \simeq 0 \to x \simeq 1 \vee y \simeq 1 & \\
x \ominus y \simeq 1 \to x \simeq 0 \vee y \simeq 1 & x \ominus y \simeq 1 \to x \simeq 1 \vee y \simeq 0 & \\
f(\vec{x}) \simeq 0 \to x_i \simeq 0 \vee x_i \simeq 1 & & \\
f(\vec{x}) \simeq 1 \to x_i \simeq 0 \vee x_i \simeq 1 & \text{for every free } f \in \mathscr{F} \text{ and } i & \\
\overline{0} \simeq 1 \quad \overline{1} \simeq 0 & & \left.\begin{array}{l} \\ \\ \\ \\ \end{array}\right\} \mathbb{D}_2\text{: simplifiers} \\
0 \cdot x \simeq 0 \quad x \cdot 0 \simeq 0 & 1 \cdot x \simeq x \quad x \cdot 1 \simeq x & \\
0 + x \simeq x \quad x + 0 \simeq x & 1 + x \simeq 1 \quad x + 1 \simeq 1 & \\
0 \oplus x \simeq x \quad x \oplus 0 \simeq x & 1 \oplus x \simeq \overline{x} \quad x \oplus 1 \simeq \overline{x} & \\
0 \ominus x \simeq \overline{x} \quad x \ominus 0 \simeq \overline{x} & 1 \ominus x \simeq x \quad x \ominus 1 \simeq x &
\end{array}
$$

Proposition 5.25 We gather some properties of $\mathbb{D}$.

(i) $\mathbb{D}$ is a strict subtheory of $\mathbb{B}$, since for example $\mathbb{D} \not\models \forall x.\, x \simeq 0 \vee x \simeq 1$.

(ii) $\mathrm{Ded}(\mathbb{B}) \supset \mathrm{Ded}(\mathbb{D}) \supset \mathrm{Ded}(\mathbb{C})$

(iii) Ξ is a correct transformation from $\mathbb{B}$ to $\mathbb{D}$.

Proof:

(i) We construct a three-element $\mathbb{D}$-model over the domain $\{0, {}^1\!/_2, 1\}$, interpreting 0 and 1 as themselves. Regarding the interpretation of the Boolean operators, if at least one of the arguments is 0 or 1, then the result is read off $\mathbb{D}_2$. Otherwise we always evaluate to ${}^1\!/_2$, which we also do for the free functions. The interpretation of predicates is arbitrary.

(ii) Every axiom of $\mathbb{D}$ is $\mathbb{B}$-valid; and every axiom of $\mathbb{C}$ is $\mathbb{D}$-valid because $\mathbb{D}$ contains a clausification of $\mathbb{C}$. The inclusions are strict because of (i) and because of $\mathbb{C} \not\models \overline{0} \simeq 1$. For the latter, build a three-element model over $\{0, {}^1\!/_2, 1\}$ where every operator always returns ${}^1\!/_2$.

(iii) This follows from (ii) by Thm. 5.23.

□

Most of the defining laws of Boolean algebras are not entailed by $\mathbb{D}$. For example, the operators $\cdot$ and $+$ lack associativity, commutativity and idempotence, and do not distribute over each other.

As an aside, the axioms of $\mathbb{B}$ are logically independent of each other, and so are those of $\mathbb{C}$, but this property is not true of $\mathbb{D}$. For example, from $\overline{x} \simeq 1 \oplus x$ and $x \oplus 0 \simeq x$ we obtain $\overline{0} \simeq 1 \oplus 0 \simeq 1$. Similarly, $x \cdot y \simeq 1 \rightarrow x \simeq 1$ is implied by $x \cdot y \simeq 1 \rightarrow y \simeq 1$ and $x \cdot 1 \simeq x$, since $x \cdot y \simeq 1$ then entails $y \simeq 1$, such that $1 \simeq x \cdot y \simeq x \cdot 1 \simeq x$ holds.

5.4.2 Saturating the Approximation

In this subsection $\mathbb{D}$ is analyzed from a superposition point of view.

Assumption 5.26 We stipulate to work with an arbitrary simplification ordering $\succ$ total on ground terms and predicative atoms such that

(i) $op(\vec{x}) \succ 1 \succ 0$ for every operator $op \in \mathscr{F}$, free or non-free,

(ii) $\{1 \oplus x,\ x \oplus 1,\ 0 \ominus x,\ x \ominus 0\} \succcurlyeq \{\overline{x}\}$.

Examples of suitable simplification orderings are ground total

(a) Knuth-Bendix orderings [KB70] where say all the operators have the same weight,

(b) lexicographic path orderings [KL80] based on a precedence $\succ_{\mathscr{F}}$ such that $op \succ_{\mathscr{F}} 1 \succ_{\mathscr{F}} 0$ for every non-constant $op \in \mathscr{F}$ and that $\{\oplus, \ominus\} \succcurlyeq_{\mathscr{F}} \{^-\}$.

Recall that the ordering $\succ$ is extended to literals as multiset comparison of the following literal images: $s \simeq t \mapsto \{s, t\}$, $s \not\simeq t \mapsto \{s, s, t, t\}$, $P(\vec{t}) \mapsto \{P(\vec{t})\}$ and $\neg P(\vec{t}) \mapsto \{P(\vec{t}), P(\vec{t})\}$. Clauses are compared as multisets of their literals.

Proposition 5.27 In every clause of $\mathbb{D}_1$ the negative literal is the greatest one, and its left-hand side is greater than the right-hand side. The latter is also true of the unit equations of $\mathbb{D}_2$.

Proof: The first and the second statement follow from the subterm property of $\succ$ and Ass. 5.26 (i). The third additionally relies on Ass. 5.26 (ii). □

Lemma 5.28 $\mathbb{D}$ is saturated up to redundancy with arbitrary selection. This is detectable with subsumption and deletion of syntactic tautologies.

Proof: First we show that $\mathbb{D}_1$ itself is saturated. As to superposition left or right inferences, regardless of selection there is no reductive clause within $\mathbb{D}_1$, since by Prop. 5.27 no positive literal is strictly maximal, nor maximal. For the same reason there are no equality factoring inferences. Equality resolution steps fail because no negative literal has a left-hand side unifiable with the right-hand side.

Next we prove that $\mathbb{D}_2$ alone is saturated. This amounts to the computation and analysis of 16 critical pairs:

- for every $\circ \in \{\cdot, +, \oplus, \ominus\}$ and $\alpha \in \{0, 1\}$:
  $\alpha \circ x \simeq t,\ \ x \circ \alpha \simeq t \ \ \vdash \ \ t \simeq t$
- for each $\circ \in \{\cdot, +\}$ with one $\{\alpha, \beta\} = \{0, 1\}$:
  $\alpha \circ x \simeq \alpha,\ \ x \circ \beta \simeq x \ \ \vdash \ \ \alpha \simeq \alpha$
  plus the variant symmetric in $\circ$
- for each $\circ \in \{\oplus, \ominus\}$ with one $\{\alpha, \beta\} = \{0, 1\}$:
  $\alpha \circ x \simeq x,\ \ x \circ \beta \simeq \overline{x} \ \ \vdash \ \ \alpha \simeq \overline{\beta}$
  plus the variant symmetric in $\circ$

The pairs of the first and of the second category are syntactic tautologies; those of the third are subsumed by the $\mathbb{D}_2$-equations $\overline{0} \simeq 1$ and $\overline{1} \simeq 0$.

Finally we need to study superposition inferences from $\mathbb{D}_2$ into $\mathbb{D}_1$. It does not make a difference whether negative literals are selected or not. All in all there are 60 inferences:

- for every $\{\alpha, \beta\} = \{0, 1\}$:
  $\overline{x} \simeq \alpha \to x \simeq \beta,\ \ \overline{\alpha} \simeq \beta \ \ \vdash \ \ \beta \simeq \alpha \to \alpha \simeq \beta$ and
  $\overline{x} \simeq \alpha \to x \simeq \beta,\ \ \overline{\beta} \simeq \alpha \ \ \vdash \ \ \alpha \simeq \alpha \to \beta \simeq \beta$
- for each $\circ \in \{\cdot, +\}$ with one $\{\alpha, \beta\} = \{0, 1\}$ per group:
  $x \circ y \simeq \alpha \to x \simeq \alpha \vee y \simeq \alpha,\ \ \alpha \circ y \simeq \alpha \ \ \vdash \ \ \alpha \simeq \alpha \to \alpha \simeq \alpha \vee y \simeq \alpha$
  plus the symmetric variant from $x \circ \alpha \simeq \alpha$;
  $x \circ y \simeq \alpha \to x \simeq \alpha \vee y \simeq \alpha,\ \ \beta \circ y \simeq y \ \ \vdash \ \ y \simeq \alpha \to \beta \simeq \alpha \vee y \simeq \alpha$
  plus the symmetric variant from $x \circ \beta \simeq \beta$;
  $x \circ y \simeq \alpha \to x \simeq \alpha,\ \ \alpha \circ y \simeq y \ \ \vdash \ \ y \simeq \alpha \to \alpha \simeq \alpha$
  plus the symmetric variant from $x \circ \alpha \simeq \alpha$,

and all the same into $x \circ y \simeq \alpha \to y \simeq \alpha$;

$x \circ y \simeq \alpha \to x \simeq \alpha,\ \ \beta \circ y \simeq \beta \ \vdash\ \beta \simeq \alpha \to \beta \simeq \alpha$ and
$x \circ y \simeq \alpha \to x \simeq \alpha,\ \ x \circ \beta \simeq \beta \ \vdash\ \beta \simeq \alpha \to x \simeq \alpha$
plus the symmetric variants into $x \circ y \simeq \alpha \to y \simeq \alpha$

- for each $\circ \in \{\oplus, \ominus\}$ with one $\{\alpha, \beta\} = \{0, 1\}$ per group:
  $x \circ y \simeq \alpha \to x \simeq \alpha \vee y \simeq \beta,\ \ \alpha \circ y \simeq y \ \vdash\ y \simeq \alpha \to \alpha \simeq \alpha \vee y \simeq \beta$
  $x \circ y \simeq \alpha \to x \simeq \beta \vee y \simeq \alpha,\ \ \alpha \circ y \simeq y \ \vdash\ y \simeq \alpha \to \alpha \simeq \beta \vee y \simeq \alpha$
  both plus the symmetric variants from $x \circ \alpha \simeq x$;
  $x \circ y \simeq \alpha \to x \simeq \alpha \vee y \simeq \beta,\ \ \beta \circ y \simeq \overline{y} \ \vdash\ \overline{y} \simeq \alpha \to \beta \simeq \alpha \vee y \simeq \beta$
  $x \circ y \simeq \alpha \to x \simeq \beta \vee y \simeq \alpha,\ \ \beta \circ y \simeq \overline{y} \ \vdash\ \overline{y} \simeq \alpha \to \beta \simeq \beta \vee y \simeq \alpha$
  both plus the symmetric variants from $x \circ \beta \simeq \overline{x}$;
  $x \circ y \simeq \alpha \to x \simeq \beta \vee y \simeq \beta,\ \ \alpha \circ y \simeq \overline{y} \ \vdash\ \overline{y} \simeq \alpha \to \alpha \simeq \beta \vee y \simeq \beta$
  plus the symmetric variant from $x \circ \alpha \simeq \overline{x}$;
  $x \circ y \simeq \alpha \to x \simeq \beta \vee y \simeq \beta,\ \ \beta \circ y \simeq y \ \vdash\ y \simeq \alpha \to \beta \simeq \beta \vee y \simeq \beta$
  plus the symmetric variant from $x \circ \beta \simeq x$;
  $x \circ y \simeq \alpha \to x \simeq \alpha \vee y \simeq \alpha,\ \ \alpha \circ y \simeq \overline{y} \ \vdash\ \overline{y} \simeq \alpha \to \alpha \simeq \alpha \vee y \simeq \alpha$
  plus the symmetric variant from $x \circ \alpha \simeq \overline{x}$;
  $x \circ y \simeq \alpha \to x \simeq \alpha \vee y \simeq \alpha,\ \ \beta \circ y \simeq y \ \vdash\ y \simeq \alpha \to \beta \simeq \alpha \vee y \simeq \alpha$
  plus the symmetric variant from $x \circ \beta \simeq x$

All the resolvents are syntactic tautologies or are subsumed by one of the clauses $0 \not\simeq 1$, $\overline{x} \simeq 0 \to x \simeq 1$ and $\overline{x} \simeq 1 \to x \simeq 0$. □

Indeed the superposition-based theorem prover SPASS provides subsumption and tautology deletion (see [Wei01, Def. 4.18, Def. 4.16 and App. A.3]), and detects that $\mathbb{D}$ is saturated. Note that this works for arbitrary free function and predicate symbols, since no clause with such a symbol is involved in any inference.

One might like to extend $\mathbb{D}_2$ to cope with expressions $x \circ x$, or with $x \circ \overline{x}$ and its symmetric variant. Let us look at $\forall x.\, x \cdot \overline{x} \simeq 0$. Modulo $\mathbb{D}$ this implies $\forall x.\, x \simeq 0 \vee \overline{x} \simeq 0$. That is, such a simplifier would extend $\mathbb{D}$ to $\mathbb{B}$, which is not what we want. All the simplifiers induced by the above-mentioned expressions have the same effect, except idempotence of con- and disjunction. As to these, indeed SPASS manages to finitely saturate the corresponding extension of $\mathbb{D}$. But one can observe that with these axioms added SPASS usually has to derive more clauses until a proof is found, which means they are practically useless.

5.4.3 Deciding the Universal Fragment of $\mathbb{B}$

Assumption 5.29 For Skolemization purposes we stipulate that $\mathscr{F}$ contains a sufficient amount of free function symbols. Furthermore, we sharpen

Ass. 5.26 in that 0 and 1 now are the smallest and the second smallest ground term, respectively. The examples given in Ass. 5.26 carry over if the actual precedence ends in $1 \succ 0$.

Definition 5.30 A *01-clause* is a ground clause in every equational atom of which at least one side is 0 or 1. A *01-clause set* is a set of such clauses.

The essence of the following proposition is that $\mathbb{B}$-validity of universal formulae reduces to $\mathbb{D}$-unsatisfiability of 01-clause sets. To obtain these, the universal formulae are transformed via Ξ, negated and clausified in a standard way. For brevity we reuse the clausification relation $\Longrightarrow$ of Sect. 5.3.5.

Proposition 5.31 Consider a quantifier-free formula ϕ with $\mathrm{free}(\phi) = \{\vec{x}\}$. Let $\neg\Xi(\phi) \Longrightarrow \psi$ and $\psi' \equiv \psi\{\vec{x} \mapsto \vec{c}\}$ with free constants $\{\vec{c}\}$. Then ψ' is a conjunction of 01-clauses, and $\mathbb{B} \models \forall\vec{x}.\, \phi$ holds iff $\mathbb{D} \cup \{\psi'\}$ is unsatisfiable.

Proof: By Thm. 5.23 and Prop. 5.8 (i) we have equivalence of $\mathbb{B} \models \forall\vec{x}.\, \phi$ and $\mathbb{D} \models \forall\vec{x}.\, \Xi(\phi)$. The latter holds iff $\mathbb{D} \cup \{\exists\vec{x}.\, \neg\Xi(\phi)\}$ is unsatisfiable. The formula $\neg\Xi(\phi)$ inherits quantifier-freeness from ϕ because Ξ by construction does not introduce quantifiers. Therefore $\neg\Xi(\phi)$ and ψ are equivalent by Prop. 5.19 (ii). Hence the sets $\mathbb{D} \cup \{\exists\vec{x}.\, \neg\Xi(\phi)\}$ and $\mathbb{D} \cup \{\exists\vec{x}.\, \psi\}$ are equisatisfiable, which via Skolemization continues to $\mathbb{D} \cup \{\psi'\}$. The formula ψ is a clause normal form by Prop. 5.19 (iii), and so is its instance ψ'. This instance is ground because Ξ by construction does not introduce variables. Consider now an equational atom within any of the conjuncts of ψ. Since $\Longrightarrow$ does not modify atoms, the shape of this atom originates from an application of Ξ. By Def. 5.7 it is either $s \simeq 0$ or $s \simeq 1$. □

We eventually have to show that all superposition derivations that start from a satisfiable clause set are finite. The key argument will be that no consequence contains longer atoms or new symbols.

Definition 5.32 Clauses and clause sets are measured as follows:

(i) The *maximal atom length* $\|\_\|$ is $\|s \bowtie t \vee C\| = \max(|s \simeq t|, \|C\|)$ and $\|{\pm}P(\vec{t}) \vee C\| = \max(|P(\vec{t})|, \|C\|)$ for non-empty clauses, and 0 for the empty clause.

(ii) On clause sets, let $\|\Gamma\| = \max(\{\|C\|\colon C \in \Gamma\} \cup \{0\})$.

Furthermore, for every term t there exists uniquely a smallest subsignature $\Sigma' = (\mathscr{S}, \mathscr{P}', \mathscr{F}', \mathscr{V}, \tau')$ of Σ such that t is a Σ'-term. We call $\Sigma(t) = \mathscr{P}' \cup \mathscr{F}'$ the *symbols of* t, and extend the notion up to clause sets correspondingly.

Proposition 5.33 Consider a $\mathbb{D}$-satisfiable 01-clause set Γ without syntactic tautologies, and a superposition consequence C of $\mathbb{D} \cup \Gamma$. Then C is $\mathbb{D}$-subsumable, or a syntactic tautology, or a 01-clause with $\|C\| \leq \|\Gamma\|$ and $\Sigma(C) \subseteq \Sigma(\mathbb{D} \cup \Gamma)$.

Proof: We inspect the possible inferences, grouped according to the parent clauses. As short hand for $\mathcal{I}\,\dfrac{\vec{C}}{D}$, in this proof we simply write $\vec{C} \vdash D$. In the following α and β range over $\{0, 1\}$.

- Inferences within Γ: Let us start with a summary of inference steps specialized to the case of 01-clauses; detailed explanations are given below.

equality resolution	$C \vee \alpha \not\simeq \alpha$	$\vdash$	C
equality factoring	$C \vee s \simeq \alpha \vee s \simeq \beta$	$\vdash$	$C \vee \alpha \not\simeq \beta \vee s \simeq \beta$
superposition	$C \vee l \simeq \alpha,\ s[l] \bowtie \beta \vee D$	$\vdash$	$s[\alpha] \bowtie \beta \vee C \vee D$
	$C \vee l \simeq \alpha,\ \pm P(\vec{t}[l]) \vee D$	$\vdash$	$\pm P(\vec{t}[\alpha]) \vee C \vee D$
resolution	$C \vee P(\vec{t}),\ \neg P(\vec{t}) \vee D$	$\vdash$	$C \vee D$
factorization	$C \vee P(\vec{t}) \vee P(\vec{t})$	$\vdash$	$C \vee P(\vec{t})$

  As to equality factoring, its general ground version is $C \vee s \simeq t \vee s \simeq t' \vdash C \vee t \not\simeq t' \vee s \simeq t'$ where among other conditions $s \succ t$ and $s \simeq t \succeq s \simeq t'$. By the shape of 01-clauses at least one of the terms s and t is 0 or 1, which are the smallest terms. Now if it were not t, then by $s \succ t$ it could not be s as well. So we know that $t \equiv \alpha$ for some $\alpha \in \{0, 1\}$. Then from $s \simeq \alpha \succeq s \simeq t'$ we can derive that t' is not above α, hence in $\{0, 1\}$ as well because there are no smaller terms.
  Regarding superposition, we generally have $C \vee l \simeq r,\ s[l] \bowtie t \vee D \vdash s[r] \bowtie t \vee C \vee D$, the ordering restrictions containing $l \succ r$ and $s[l] \succ t$. Hence r specializes into some α, and t into some β.
  Concerning the shape of the inference conclusion, there are only three newly composed literals: $\alpha \not\simeq \beta$ within equality factoring, and $s[\alpha] \bowtie \beta$ and $\pm P(\vec{t}[\alpha])$ within superposition. They are all ground; if they are equational, then at least one side is 0 or 1; each of their symbols is present in some parent clause; and the atom lengths do not grow because $\alpha \simeq \beta$ is minimal, and because $|l| \geq |\alpha|$. The remaining literals inherit these properties from the parent clauses.
- Inferences within $\mathbb{D}$: As shown in Lem. 5.28, all the inference conclusions of $\mathbb{D}$ are $\mathbb{D}$-subsumable or syntactic tautologies.
- Inferences between Γ and $\mathbb{D}_1$: The calculus contains two binary inference rules, namely superposition and resolution. The latter is not applicable here because $\mathbb{D}$ does not contain any predicative atom. The

former in the ground case requires a main premise $C \vee l \simeq r$ where $l \simeq r \succ C$. By Prop. 5.27 this condition is not satisfied by any clause within $\mathbb{D}_1$. Hence we study superposition inferences with main premise from Γ and side premise from $\mathbb{D}_1$.

If the side premise is $1 \not\simeq 0$, then because of $l \succ r$ the main premise must have the shape $C \vee 1 \simeq 0$. Furthermore we have $1 \simeq 0 \succ C$, and the only literals smaller than $1 \simeq 0$ are $0 \simeq 0$ and $0 \not\simeq 0$. The first would turn C into a syntactic tautology and hence cannot occur. The remaining possibility for the shape of the main premise is $(\bigvee_{i=1}^{n} 0 \not\simeq 0) \vee 1 \simeq 0$, but this is $\mathbb{D}$-unsatisfiable.

The remaining side premises follow the pattern $op(\vec{x}) \not\simeq \beta \vee D$ for some operator op, $\beta \in \{0, 1\}$ and 01-constraint D for $\vec{x}$. Because of Prop. 5.27 an inference can only take place into $op(\vec{x})$, and hence is described by $C \vee op(\vec{t}) \simeq \alpha,\ op(\vec{x}) \not\simeq \beta \vee D \vdash \alpha \not\simeq \beta \vee C \vee D\{\vec{x} \mapsto \vec{t}\}$. The resolvent is ground because the main premise is ground and because $\mathrm{free}(D) \subseteq \{\vec{x}\}$. One side of each equational literal is 0 or 1 because this is also the case for the main premise, and because D is a 01-constraint. It is obvious that no new symbols are introduced. Regarding atom lengths, note that $\|D\{\vec{x} \mapsto \vec{t}\}\| = |t_i \simeq \alpha'| < |op(\vec{t}) \simeq \alpha|$ for some i and α'.

- Inferences between Γ and $\mathbb{D}_2$: Again we only need to consider superposition and resolution. Let us first consider overlaps from Γ strictly into $\mathbb{D}_2$. By Prop. 5.27 the only non-variable subterms available are 0 and 1. The former is the smallest term and cannot become maximal in a main premise. The latter needs a main premise $C \vee 1 \simeq 0$, but that does not exist as shown in the last but one paragraph.

  Next we come to superposition inferences from $\mathbb{D}_2$ into Γ. Those into equational literals follow the pattern $op(\vec{l}) \simeq r,\ s[op(\vec{l'})] \bowtie \alpha \vee C \vdash s[r\sigma] \bowtie \alpha \vee C$ where $\vec{l}\sigma \equiv \vec{l'}$. By inspection of $\mathbb{D}_2$ we have $\mathrm{free}(r) \subseteq \mathrm{free}(op(\vec{l}))$, which turns the resolvent ground, and $|op(\vec{l})| > |r|$. Since no variable occurs twice in r, this extends to $|op(\vec{l'})| > |r\sigma|$, such that the atoms in the resolvents are not longer than those in the parents. The resolvent is evidently a 01-clause, and does not introduce new symbols.

  The remaining superposition inferences into predicative literals obey the pattern $op(\vec{l}) \simeq r,\ \pm P(\vec{t}[op(\vec{l'})]) \vee C \vdash \pm P(\vec{t}[r\sigma]) \vee C$ and are treated correspondingly.

□

Theorem 5.34 Superposition with deletion of syntactic tautologies and forward subsumption decides the universal fragment of $\mathbb{B}$.

Proof: The problem to decide is, given an arbitrary quantifier-free formula ϕ, whether $\mathbb{B} \models \forall \vec{x}.\, \phi$ holds. By Prop. 5.31 this is equivalent to the unsatisfiability of $\mathbb{D} \cup \Gamma$ for some computable 01-clause set Γ. In the unsatisfiable case, by refutational completeness of superposition any fair derivation will eventually produce the empty clause. If, however, $\mathbb{D} \cup \Gamma$ is satisfiable, then we have to show that with forward subsumption and deletion of syntactic tautologies there is no infinite sequence $C_1, C_2, \ldots$ of kept inference conclusions starting from $\mathbb{D} \cup \Gamma$.

Assume the contrary. By Prop. 5.33 these conclusions are limited with respect to the maximal atom length and to the symbols occurring in them. Hence they are built over only finitely many distinct literals, say L_1 through L_N. Let us now interpret clauses as ground terms over a signature that consists of L_1 through L_N as constants plus a binary operator $\vee$, for example with parenthesizing to the right. Then we can apply Kruskal's theorem (Thm. 2.22) to the sequence $C_1, C_2, \ldots$ and obtain $C_i \leq_{\text{emb}} C_j$ for some $i < j$. In other words, if $C_i \equiv L_1 \vee (\ldots \vee (L_{m-1} \vee L_m) \ldots)$ and $C_j \equiv L'_1 \vee (\ldots \vee (L'_{n-1} \vee L'_n) \ldots)$, then C_j rewrites into C_i by repeated application of the rewrite rules $x \vee y \to x$ and $x \vee y \to y$. Hence, the multiset $\{L'_1, \ldots, L'_n\}$ is a superset of $\{L_1, \ldots, L_m\}$. But this implies that C_i subsumes C_j and contradicts the assumption on the derivation. □

If superposition derivations are augmented with arbitrary simplifications, then the clauses developed are not necessarily bounded with respect to the maximal atom length. For example, consider the clause $c \cdot 1 \simeq 0$ when using a lexicographic path ordering induced by a precedence with $\cdot \succ_{\mathscr{F}} +$. The axiomatization $\mathbb{D}$ covers the clause instances $0 \cdot 1 \simeq 0$ and $0 + \cdots + 0 \simeq 0$ with an arbitrary number of addends, say parenthesized to the right. Then $c \simeq 0 + \cdots + 0, \quad 0 + \cdots + 0 \simeq 0, \quad 0 \cdot 1 \simeq 0 \models c \cdot 1 \simeq 0$ holds. Since $c \cdot 1$ is the greatest term of all these, we may simplify the clause $c \cdot 1 \simeq 0$ to $c \simeq 0 + \cdots + 0$, which is arbitrary long. So one has to inspect the concrete reduction rules whether they are admissible.

In particular this is evident whenever a clause is discarded, or a literal within a clause. As to the superposition theorem prover SPASS, this already covers the rules trivial literal elimination, subsumption, condensation, tautology deletion, conflict, matching replacement resolution, and assignment equation deletetion, see [Wei01, Chap. 4.4]. The remaining reductions of SPASS are unit, non-unit and contextual rewriting; for the latter, see [WSH+07, Sect. 2.1]. They all follow the pattern that a clause $C[l\sigma]$ is rewritten to $C[r\sigma]$, provided there is a clause $D \equiv l \simeq r \vee D'$ such that, among other conditions, $l\sigma \succ r\sigma$ is true. If D is a 01-clause, then r is 0 or 1, and so is $r\sigma$. If D is a simplifier from $\mathbb{D}_2$, then the reduction is admissible as well.

Note that the termination argument is also compatible with explicit splitting as implemented in SPASS (cf. [Wei01, Chap. 4.5]), since in the second branch no new Skolem symbols are introduced. Implicit splitting is appropriate as well, since only finitely many new symbols are introduced.

5.5 Beyond Bits

We have presented an alternative approach for tackling entailment problems with respect to the theory of bits: The axiom $\forall x.\, x \simeq 0 \vee x \simeq 1$ contains unshielded variables, which is problematic for first-order theorem provers. We have shown that this axiom just can be dropped at the price of a simple formula transformation. The approach should carry over to other domains of small size, like small groups or fields. First-order provers are of course not the method of choice for these domains, but have their own merit in producing proof objects, as exercised in [CMSM04].

Given a domain size of $n+1$ and constants $0, \ldots, n$ in the signature, we consider a theory $\mathbb{B}$ like the following:

$$\bigwedge_{i \neq j} i \not\simeq j \qquad \forall x. \bigvee_i x \simeq i \qquad op(\vec{t}) \simeq \alpha^{op}_{\vec{t}}$$

where the last equation is a pattern that has to be instantiated for every non-free function op, arguments $\vec{t} \in (0..n)^*$ and some $\alpha^{op}_{\vec{t}} \in \{0, \ldots, n\}$. That is, without free symbols the theory must be complete.

To obtain a weak approximation $\mathbb{C}$, we only keep the distinctness of the constants, and add solutions of each non-free operator for each α as well as a scheme for free functions:

$$\bigwedge_{i \neq j} i \not\simeq j \qquad \forall \vec{x}.\, op(\vec{x}) \simeq \alpha \rightarrow S^{op}_{\alpha}(\vec{x}) \qquad \forall \vec{x}. \bigvee_i (f(\vec{x}) \simeq i) \rightarrow \bigwedge_j \bigvee_k x_j \simeq k$$

With an appropriate simplification ordering, the standard clausification of $\mathbb{C}$ will be saturated for the same reasons as $\mathbb{D}_1$ in Sect. 5.4.2: There are no reductive clauses and hence no superposition or equality factoring inferences; and equality resolution steps do not pass the unification test.

From this we get a stronger approximation $\mathbb{D}$ if we reinsert the equations $op(\vec{t}) \simeq \alpha^{op}_{\vec{t}}$, or $\mathbb{B}$-valid generalizations thereof, but such that the resulting formula set can finitely be saturated. Finally, the formula transformation now reads as follows:

$$s \simeq t \overset{\Xi}{\longmapsto} \bigwedge_i (s \simeq i \rightarrow \bigwedge_{j \neq i} t \not\simeq j) \qquad s \not\simeq t \overset{\Xi}{\longmapsto} \bigwedge_i (s \simeq i \rightarrow t \not\simeq i)$$

$$P(\vec{t}) \overset{\Xi}{\longmapsto} \vec{t} \notin (0..n)^* \vee P(\vec{t}) \qquad \neg P(\vec{t}) \overset{\Xi}{\longmapsto} \vec{t} \notin (0..n)^* \vee \neg P(\vec{t})$$

On this basis, Prop. 5.4 and Lem. 5.5 should carry over, which are the foundation of the completeness proof for clauses. Regarding soundness, one can adapt Prop. 5.11 (i), such that Prop. 5.12 remains valid. As to completeness on arbitrary formulae, we only have to extend the $\exists$-elimination rule in Def. 5.18 (ii) to deal with the larger domain.

The transformation leads to an increase in the size of the formulae which is quadratic in the cardinality of the domain. The approach is therefore likely to be useful for small domains only, which in particular bitvectors are not an example of. We will introduce a way of reducing the theory of bitvectors to the approximated theory of bits.

5.6 Axioms on Bitvectors

Given some positive integer N, let us specify bitvectors of length up to N. We use a many-sorted setting where each vector length corresponds to one sort. The vector notation will be low-endian, with bits numbered from $N-1$ down to 0. From now on m, n range from 1 to N, and i, j range for any given m from 0 to $m-1$. Furthermore, S is to be instantiated with every sort symbol, including bits and bitvectors.

Definition 5.35 We fix an elementary signature Σ_0, which essentially holds the constructors, and an extension Σ that will contain free symbols as well as defined ones.

(i) The following schematic list contains the symbols used in the signature Σ_0. Super- and subscripts are part of the identifiers.

(a) Sorts:

bits	B
bitvectors of length m	B^m

(b) Function symbols:

vector constructor	$(\_,\ldots,\_)^m :$	$\overbrace{B\ldots B}^{m} \to B^m$
projection	$P_i^m :$	$B^m \to B$
bit constants	$0, 1 :$	$\to B$

(c) Some of the variables:

bits (B)	$x, y; x_0, y_0, x_1, y_1, \ldots$
bitvectors (B^m)	x^m, y^m, z^m
arbitrary sort S	u^S, v^S

(ii) The signature Σ extends Σ_0 with:

(a) arbitrary many other sorts,

(b) the bit operators $^-$, $\cdot$, $+$, $\oplus, \ominus$ as in Sect. 5.2, working on sort B,

(c) some named operators:

selection	$P^m_{ji}:$	$B^m \to B^{j-i+1}$	$j > i$
vector concatenation	$\_ :^m_n \_:$	$B^m\, B^n \to B^{m+n}$	$m+n \le N$
bit repetition	$\_\uparrow^m:$	$B \to B^m$	
vector equality	$\mathrm{eq}^m:$	$B^m\, B^m \to B$	
if-then-else	$\_ ?\, \_ :^S \_:$	$B\, S\, S \to S$	

(d) plus arbitrary many other function and predicate symbols.

Definition 5.36 We specify a theory $\overrightarrow{\mathbb{B}}_0$ that describes the interplay of the constructors, and a supertheory $\overrightarrow{\mathbb{B}}$ that deals with the named operators of the extended signature.

(i) These schematic Σ_0-formulae, universally closed, constitute the theory $\overrightarrow{\mathbb{B}}_0$:

(a) representation axioms

$$x^m \simeq (P^m_{m-1}(x^m), \ldots, P^m_0(x^m))^m$$

(b) projection axioms

$$x^m \simeq (x_{m-1}, \ldots, x_0)^m \to P^m_i(x^m) \simeq x_i$$

(c) B cardinality axioms

$$0 \not\simeq 1 \qquad x \simeq 0 \vee x \simeq 1$$

(ii) The theory $\overrightarrow{\mathbb{B}}$ additionally contains the following schematic Σ-formulae, again universally closed:

(a) selection and concatenation axioms

$$\begin{array}{ll} x^m \simeq (x_{m-1}, \ldots, x_0)^m \wedge y^n \simeq (y_{n-1}, \ldots, y_0)^n \to & \\ \quad P^m_{ji}(x^m) \simeq (x_j, \ldots, x_i)^{j-i+1} & j > i \\ \quad \wedge\, x^m :^m_n y^n \simeq (x_{m-1}, \ldots, x_0, y_{n-1}, \ldots, y_0)^{m+n} & m+n \le N \end{array}$$

(b) repetition axioms

$$x\uparrow^m \simeq (x, \ldots, x)^m$$

(c) vector equality axioms

$$\mathrm{eq}^m(x^m, y^m) \simeq 1 \leftrightarrow x^m \simeq y^m$$

(d) if-then-else axioms

$$0 ?\, u^S :^S v^S \simeq v^S \qquad 1 ?\, u^S :^S v^S \simeq u^S$$

along with

(e) the propositional truth table for the bit operators given in Def. 5.1:

$$\overline{0} \simeq 1 \qquad \overline{1} \simeq 0$$

$0 \cdot 0 \simeq 0$	$0 \cdot 1 \simeq 0$	$1 \cdot 0 \simeq 0$	$1 \cdot 1 \simeq 1$
$0 + 0 \simeq 0$	$0 + 1 \simeq 1$	$1 + 0 \simeq 1$	$1 + 1 \simeq 1$
$0 \oplus 0 \simeq 0$	$0 \oplus 1 \simeq 1$	$1 \oplus 0 \simeq 1$	$1 \oplus 1 \simeq 0$
$0 \ominus 0 \simeq 1$	$0 \ominus 1 \simeq 0$	$1 \ominus 0 \simeq 0$	$1 \ominus 1 \simeq 1$

In principle we can even assume N to be infinite: When it comes to the question whether some formula ϕ is a consequence of $\overrightarrow{\mathbb{B}}$, it will turn out that $\overrightarrow{\mathbb{B}}$-validity of ϕ only depends on a finite subset $\overrightarrow{\mathbb{B}}[\phi]$ of $\overrightarrow{\mathbb{B}}$ which can be determined in advance.

Both signature and axiomatization are quite large: Σ contains at least $N+1$ sorts and $\frac{N^3}{6}+N^2+\frac{23}{6}N+8$ operators. If we count selection and concatenation axioms separately, then $\overrightarrow{\mathbb{B}}$ holds $\frac{N^3}{6}+N^2+\frac{23}{6}N+21$ axioms. Given $N=32$, we have 6616 operators and 6629 axioms.

As a remedy we can go from $\overrightarrow{\mathbb{B}}$ to $\overrightarrow{\mathbb{B}}[\phi]$. Besides, at the expense of introducing some polymorphism, we can later code the signature down with free constants $\underline{N},\ldots,\underline{0}$, and replace e. g. $P_i^n(x^n)$ by $P(\underline{n},\underline{i},x^n)$.

Proposition 5.37 We gather some simple properties of $\overrightarrow{\mathbb{B}}$.

(i) Vector construction is injective.
(ii) Vector concatenation is associative.
(iii) Vector equality evaluates to 0 iff its arguments are distinct.
(iv) For every $\overrightarrow{\mathbb{B}}$-model $\mathcal{A}$, the carrier $\mathcal{A}(B)$ consists of the two elements $0^{\mathcal{A}}$ and $1^{\mathcal{A}}$, and the carrier $\mathcal{A}(B^m)$ of the 2^m distinct elements $\mathcal{A}_\mu((s_{m-1},\ldots,s_0)^m)$ where each s_i is 0 or 1.

Proof:

(i) Assume that $\mathcal{A}_\mu \models (x_{m-1},\ldots,x_0)^m \simeq (y_{m-1},\ldots,y_0)^m$ for some $\overrightarrow{\mathbb{B}}$-model $\mathcal{A}_\mu$. Then we obtain by congruence $\mathcal{A}_\mu \models P_i^m((x_{m-1},\ldots,x_0)^m) \simeq P_i^m((y_{m-1},\ldots,y_0)^m)$, and by the projection axiom $\mathcal{A}_\mu \models x_i \simeq y_i$.
(ii) Taking vector lengths into account, the above statement spells out as $\overrightarrow{\mathbb{B}} \models \forall x^m y^n z^l.\, (x^m :_n^m y^n) :_l^{m+n} z^l \simeq x^m :_{n+l}^m (y^n :_l^n z^l)$, given that $m+n+l \leq N$. It does hold because both left-hand and right-hand side of the equation are $\overrightarrow{\mathbb{B}}$-equivalent to $(P_{m-1}^m(x^m),\ldots,P_0^m(x^m),P_{n-1}^n(y^n),\ldots,P_0^n(y^n),P_{l-1}^l(z^l),\ldots,P_0^l(z^l))^{m+n+l}$, by representation and concatenation axioms.
(iii) We have $\overrightarrow{\mathbb{B}} \models \forall x^m y^m.\, \mathrm{eq}^m(x^m,y^m) \simeq 0 \leftrightarrow x^m \not\simeq y^m$ by the vector equality axioms and because B only contains 0 and 1.
(iv) As to the first statement, the given list is complete because of the axiom $\forall x.\, x \simeq 0 \vee x \simeq 1$, and its members are distinct because of the axiom $0 \not\simeq 1$. We get the same for the second statement by the representation axiom and by injectivity of vector construction (Prop. 5.37 (i)).

□

Next we show that $\overrightarrow{\mathbb{B}}$-validity can be reduced to $\overrightarrow{\mathbb{B}}_0$-validity.

Lemma 5.38 $\overrightarrow{\mathbb{B}}$ can be obtained from $\overrightarrow{\mathbb{B}}_0$ by repeated definitional extension.

Proof: First note that $\overrightarrow{\mathbb{B}}_0$ is a theory in the signature Σ_0 and hence also in Σ_0 extended with those sorts and unnamed function and predicate symbols that Σ adds to Σ_0. It remains to show that each of the named operators is explicitly defined.

Now, the selection operation could equivalently have been axiomatized via $\forall x^m.\, P^m_{ji}(x^m) \simeq (P^m_j(x^m), \ldots, P^m_i(x^m))^{j-i+1}$, which amounts to an explicit definition and forms a definitional extension by Prop. 2.16. A similar argument applies to concatenation. Repetition is immediately recognized as explicit definition.

Concerning vector equality, let ϕ denote the formula $(y \simeq 1 \leftrightarrow x^m \simeq \widetilde{x}^m) \wedge (y \simeq 0 \leftrightarrow x^m \not\simeq \widetilde{x}^m)$. Then for every x^m and $\widetilde{x}^m$ there is one and only one y such that ϕ holds. Next we show, in some detail, that the vector equality axiom is $\overrightarrow{\mathbb{B}}_0$-equivalent to $\forall X.\, \phi \leftrightarrow y \simeq \mathrm{eq}(x^m, \widetilde{x}^m)$. Since ϕ is $\overrightarrow{\mathbb{B}}_0$-equivalent to $y \simeq 1 \leftrightarrow x^m \simeq \widetilde{x}^m$, we can expand the latter formula as $\forall X.\, (y \simeq 1 \leftrightarrow x^m \simeq \widetilde{x}^m) \leftrightarrow y \simeq \mathrm{eq}(x^m, \widetilde{x}^m)$. By case split this yields $\forall X.\, (x^m \simeq \widetilde{x}^m \rightarrow (y \simeq 1 \leftrightarrow y \simeq \mathrm{eq}(x^m, \widetilde{x}^m))) \wedge (x^m \not\simeq \widetilde{x}^m \rightarrow (y \simeq 0 \leftrightarrow y \simeq \mathrm{eq}(x^m, \widetilde{x}^m)))$ and can be simplified to $\forall X.\, (x^m \simeq \widetilde{x}^m \rightarrow \mathrm{eq}(x^m, \widetilde{x}^m) \simeq 1) \wedge (x^m \not\simeq \widetilde{x}^m \rightarrow \mathrm{eq}(x^m, \widetilde{x}^m) \simeq 0)$ and $\forall X.\, x^m \simeq \widetilde{x}^m \leftrightarrow \mathrm{eq}(x^m, \widetilde{x}^m) \simeq 1$, which is just the vector equality axiom.

To prove if-then-else defined we choose $\phi \equiv (y^S \simeq \widetilde{x}^S \leftrightarrow x \simeq 0 \vee x^S \simeq \widetilde{x}^S) \wedge (y^S \simeq x^S \leftrightarrow x \simeq 1 \vee x^S \simeq \widetilde{x}^S)$. Unique existence of y^S is as obvious as in the previous case, whereas $\overrightarrow{\mathbb{B}}_0$-equivalence of $\forall X.\, \phi \leftrightarrow y^S \simeq x\,?\,x^S :^S \widetilde{x}^S$ and the conjunction of the two if-then-else axioms is shown by analysis of the cases $x \simeq 0$ and $x \simeq 1$.

Regarding negation, we can use $\phi \equiv y \simeq 0 \leftrightarrow x \simeq 1$ to characterize $y \simeq \overline{x}$, and similarly $\phi \equiv y \simeq 0 \leftrightarrow x_1 \simeq 0 \vee x_2 \simeq 0$ for conjunction $y \simeq x_1 \cdot x_2$. The remaining bit-level operations can be expressed in terms of these two. □

In principle we could go even further and reduce $\overrightarrow{\mathbb{B}}_0$-validity to $\mathbb{B}$-validity. Our hope, however, is that automated reasoning procedures can achieve more by reasoning on the bitvector level.

Proposition 5.39 Assume that Σ adds no sort and no unnamed symbols to Σ_0. Then any two $\overrightarrow{\mathbb{B}}$-models are isomorphic, and hence $\overrightarrow{\mathbb{B}}$ is complete.

Proof: By Lem. 5.38 and Prop. 2.15 (iv) we only need to prove that two arbitrary $\overrightarrow{\mathbb{B}}_0$-models $\mathcal{A}$ and $\mathcal{B}$ are isomorphic. First we will construct a homomorphism φ from $\mathcal{A}$ to $\mathcal{B}$.

Let φ map as follows: $0^{\mathcal{A}} \mapsto 0^{\mathcal{B}}$, $1^{\mathcal{A}} \mapsto 1^{\mathcal{B}}$ and $(s^{\mathcal{A}}_{m-1}, \ldots, s^{\mathcal{A}}_0)^{\mathcal{A}(m)} \mapsto (s^{\mathcal{B}}_{m-1}, \ldots, s^{\mathcal{B}}_0)^{\mathcal{B}(m)}$ where the s_i run through 0 and 1. By Prop. 5.37 (iv) this definition of φ is unambiguous and complete. It remains to show that φ commutes with going from $\mathcal{A}$ to $\mathcal{B}$. By construction of φ this only needs to be

done for the projection operator. Assume that $\alpha = (s^{\mathcal{A}}_{m-1}, \ldots, s^{\mathcal{A}}_0)^{\mathcal{A}(m)}$. Then $\varphi(\mathcal{A}(P^m_i)(\alpha)) = \varphi(s^{\mathcal{A}}_i) = s^{\mathcal{B}}_i = \mathcal{B}(P^m_i)((s^{\mathcal{B}}_{m-1}, \ldots, s^{\mathcal{B}}_0)^{\mathcal{B}(m)}) = \mathcal{B}(P^m_i)(\varphi(\alpha))$.

Swapping $\mathcal{A}$ and $\mathcal{B}$ above, we obtain a homomorphism $\overline{\varphi}$ from $\mathcal{B}$ to $\mathcal{A}$, which is inverse to φ. So the algebras are isomorphic. □

Finally we formalize that validity of a formula on bitvectors only relies on the axioms for those vector lengths that are present in the formula.

Definition 5.40 For any Σ-formula ϕ, let $\Sigma[\phi] = (\mathscr{S}', \mathscr{P}', \mathscr{F}', \mathscr{V}', \tau')$ denote the subsignature of $\Sigma = (\mathscr{S}, \mathscr{P}, \mathscr{F}, \mathscr{V}, \tau)$ such that

(i) ϕ is a $\Sigma[\phi]$-formula,
(ii) $B \in \mathscr{S}'$ and $\{0, 1\} \subseteq \mathscr{F}'$,
(iii) if $B^m \in \mathscr{S}'$, then $(\_, \ldots, \_)^m \in \mathscr{F}'$ and $P^m_i \in \mathscr{F}'$ for every i,
(iv) $\Sigma[\phi]$ is minimal with respect to sorts, predicates and operators,

and let $\overrightarrow{\mathbb{B}}[\phi]$ denote the set of $\Sigma[\phi]$-formulae of $\overrightarrow{\mathbb{B}}$.

Lemma 5.41 If ϕ is a Σ-formula, then $\overrightarrow{\mathbb{B}} \models \phi$ and $\overrightarrow{\mathbb{B}}[\phi] \models \phi$ are equivalent.

Proof: Note first that by the minimality condition $\Sigma[\phi]$ is uniquely determined and well-defined, and so is $\overrightarrow{\mathbb{B}}[\phi]$. Now, regarding the implication right-to-left, if $\overrightarrow{\mathbb{B}}[\phi]$ as a $\Sigma[\phi]$-theory entails ϕ, it also does as a Σ-theory, such that $\overrightarrow{\mathbb{B}}[\phi] \subseteq \overrightarrow{\mathbb{B}}$ implies $\overrightarrow{\mathbb{B}} \models \phi$.

For the converse assume now that $\overrightarrow{\mathbb{B}} \models \phi$. Let us recall what Σ and $\overrightarrow{\mathbb{B}}$ consist of:

- $\mathscr{S}$: bit sort B, bitvector sorts B^1 through B^N, arbitrary other sorts
- $\mathscr{P}$: arbitrary free predicates
- $\mathscr{F}$: constructors $(\_, \ldots, \_)^m$, projections P^m_i, bits 0 and 1, defined operators (see Lem. 5.38), arbitrary free functions
- $\overrightarrow{\mathbb{B}}$: representation and projection axioms, bit axioms, explicit and implicit definitions (again cf. Lem. 5.38)

We obtain $\mathscr{S}'$, $\mathscr{P}'$, $\mathscr{F}'$ and $\overrightarrow{\mathbb{B}}[\phi]$ by removing from Σ and $\overrightarrow{\mathbb{B}}$ unused

(a) "other" sorts as well as free functions and predicates,
(b) defined operators plus their definitions,
(c) bitvector sorts along with corresponding constructors and projections plus representation and projection axioms.

Next we show that entailment of ϕ is invariant under each of these transformations. Stage (a) is justified because, looking at it the other way, adding symbols to the signature of a theory yields a conservative extension. Stage (b) is an application of Lem. 5.38 and Prop. 2.15 (i). Let us look at a single transformation step within stage (c), assuming we go from $\mathcal{T}$ to $\mathcal{T}'$ removing bitvectors of length m. More precisely we have $\mathcal{T} \models \phi$, $\overrightarrow{\mathbb{B}} \supseteq \mathcal{T} =$

$\mathcal{T}' \cup \{\forall X.\, x^m \simeq (P^m_{m-1}(x^m), \ldots, P^m_0(x^m))^m\} \cup \{\forall X.\, P^m_i((x_{m-1}, \ldots, x_0)^m) \simeq x_i : 0 \leq i < m\}$ and $\mathcal{T}' \supseteq \overrightarrow{\mathbb{B}}[\phi]$ with some signatures $\Sigma_{\mathcal{T}}$ and $\Sigma_{\mathcal{T}'}$. In order to show $\mathcal{T}' \models \phi$, consider a $\Sigma_{\mathcal{T}'}$-algebra and $\mathcal{T}'$-model $\mathcal{A}$. Then we can construct a $\Sigma_{\mathcal{T}}$-expansion $\mathcal{B}$ of $\mathcal{A}$ where the interpretation of B^m is the m-fold cartesian product of the one of B, i. e. $\mathcal{B}(B^m) = \mathcal{A}(B)^m$, $(\alpha_{m-1}, \ldots, \alpha_0)^{\mathcal{B}(m)} = (\alpha_{m-1}, \ldots, \alpha_0)$ and $\mathcal{B}(P^m_i)((\alpha_{m-1}, \ldots, \alpha_0)) = \alpha_i$, the α_j ranging through $\mathcal{A}(B) = \{0^{\mathcal{A}}, 1^{\mathcal{A}}\}$. The algebras $\mathcal{A}$ and $\mathcal{B}$ agree on $\Sigma_{\mathcal{T}'}$-formulae; hence $\mathcal{B}$ satisfies $\mathcal{T}'$. By construction $\mathcal{B}$ is a $\mathcal{T}$-model. Therefore $\mathcal{B}_\nu \models \phi$ for any $\mathcal{B}$-assignment ν, and $\mathcal{A}_\mu \models \phi$ for any $\mathcal{A}$-assignment μ since ϕ is a $\Sigma_{\mathcal{T}'}$-formula. □

Note that in the preceding lemma, conditions (ii) and (iii) cannot be dropped in general, since they fix the cardinality of the carrier of B^m. See for example $\phi \equiv \forall x^1 y^1 z^1.\, x^1 \simeq y^1 \vee y^1 \simeq z^1 \vee z^1 \simeq x^1$.

5.7 Extending the Transformational Approach

The bitvector axiomatization $\overrightarrow{\mathbb{B}}$ is a supertheory of the axiomatization $\mathbb{B}$ of the algebra of propositions, and therefore inherits the shortcomings of the latter. Here our aim is to lift the transformational approach up to the vector level.

One could directly map according to the generalized scheme of Ξ for arbitrary bounded domains. Assume $s, t \in \mathscr{T}_{B^m}(\Sigma)$. Then

$$s \simeq t \;\longmapsto\; \bigwedge_{\vec{\imath} \in (01)^m} \left(s \simeq (\vec{\imath})^m \rightarrow \bigwedge_{\substack{\vec{\jmath} \in (01)^m \\ \vec{\jmath} \neq \vec{\imath}}} t \not\simeq (\vec{\jmath})^m \right)$$

whereby one atom is mapped onto a formula with 4^m ones. Given $m = 32$, we observe a factor of 18,446,744,073,709,551,616, which makes the approach unfeasible in practice.

More reasonably, using the equality function eq^m for bitvectors of length m, we can rewrite bitvectors equalities $s \simeq t$ into $\mathrm{eq}^m(s, t) \not\simeq 0$, and inequalities $s \not\simeq t$ into $\mathrm{eq}^m(s, t) \not\simeq 1$. Additionally, a bit-level definition of eq^m is needed:

$$\mathrm{eq}^m(x^m, y^m) \simeq \stackrel{m-1}{\underset{i=0}{\cdot}} (P^m_i(x) \ominus P^m_i(y))$$

Conjecture	standard encoding	transformed	with solving
$x \neq 0 \rightarrow (1 < x \leftrightarrow 0 < x - 1)$	$> 100{,}000$	4.9	0.3
$x \neq 0 \rightarrow (1 \leq x \leftrightarrow 0 \leq x - 1)$	$> 100{,}000$	5.0	0.3
$x < x - z \leftrightarrow x < z$	$> 100{,}000$	656.7	2.8
$y - x < z - x \wedge x \leq z \rightarrow y < z$	$> 100{,}000$	20,663.7	3.4
$y < z \wedge x \leq y \rightarrow y - x < z - x$	$> 100{,}000$	8,303.4	3.8
$y - x \leq z - x \wedge x \leq z \rightarrow y \leq z$	$> 100{,}000$	14,267.3	3.0
$y \leq z \wedge x \leq y \rightarrow y - x \leq z - x$	$> 100{,}000$	11,088.5	2.7

Table 5.1: Experimental results in bitvector arithmetic

In Sect. 5.4.3, we have given a superposition-based decision procedure for the universal fragment of the theory of bits. For the extension to bitvectors, we can deal with supertheories of $\overrightarrow{\mathbb{B}}_0$ and $\mathbb{B}$ for which the vector operators outside Σ_0 are definitional extensions by equations as just exercised for vector equality; and in the reduction ordering, the definiendum must be greater than the definiens. Besides, quantifications over vectors shall be turned into quantifications over the individual bits. Then, running superposition, eventually all vector-level expressions can be simplified into bit-level expressions, such that the termination result carries over.

At the moment, this approach is being pursued in an ongoing Master's Thesis [Rus08]. In collaboration with NICTA Canberra, a benchmark suite of bitvector problems has been devised. It holds integer arithmetic modulo which is unsigned when it comes to comparisons. For SPASS, addition and comparison of bitvectors are defined in terms of bit-level logical operations. As a starting point, bitvectors of length 8 have been chosen; all conjectures are universal and actually valid.

First experimental results are contained in Tab. 5.1. The first column holds the theorem to be proved. The remaining ones display run-times in seconds on a Dell PowerEdge server equipped with 3GHz Xeon processors for three different proof attempts:

(i) In the *standard encoding*, the axiomatization is just the theory of bitvectors without transformation.

(ii) The *transformed* one follows the approach developed in this chapter, with SPASS as it is.

(iii) The variant *with solving* starts from the same specification, but with an extended version of SPASS with a preliminary implementation of explicit solving for theory operators: For example, the clause $a \cdot b \simeq 1$ immediately is replaced by the two clauses $a \simeq 1$ and $b \simeq 1$, which is a

simplification with respect to our axiomatization.
With the standard encoding, not a single proof is found within more than one day of computation. To check the completeness of the axiomatization, one may consider the first conjecture for bitvectors of length 2 (sic). Then a proof will be found at least, but it will take around ten minutes.

The second column indicates that the transformation approach improves upon this. Clearly, the problems are demanding for SPASS, which is orders of magnitude behind state-of-the-art solvers like YICES [DdM06] that can discharge such proof obligations in fractions of a second. However, the third column shows that orders of magnitude can be gained already with a preliminary implementation of theory-specific simplifications. The next step is to refine this implementation and explore how far one can get, in particular outside the fragment that the solvers can deal with.

6 Superposition for Bounded Domains

6.1 Introduction

Standard superposition is not a decision procedure for first-order bounded-domain problems. For example, the superposition calculus need not terminate on a given clause set even if all function symbols are constants and hence any Herbrand model has a finite domain. This is because simplifications like rewriting or subsumption do not take finiteness into account.

In the following we will study the first-order theory of domains the size of which is explicitly bounded from above. That is, any domain element equals one of some given constants. For convenience we denote them by $1, \ldots, n$ although they are not necessarily distinct, and call them *digits*. So the theory at hand is

$$\forall x.\, x \simeq 1 \vee \ldots \vee x \simeq n$$

If run without care, standard superposition need not terminate even on the theory axiomatization alone. In case $n = 2$ the cardinality-bounding clause $x \simeq 1 \vee x \simeq \underline{2}$ overlaps with itself at the underlined position and produces the resolvent $x \simeq 1 \vee x \simeq y \vee y \simeq 1$. The latter is the starting point for an infinite sequence of clauses

$$x_1 \simeq 1 \vee \left(\bigvee_{i=1}^{2^k-1} x_i \simeq x_{i+1} \right) \vee x_{2^k} \simeq 1$$

no element of which subsumes any other; and indeed this sequence can be observed for a number of popular theorem provers.

Taking more care, one can choose an ordering such that 1 and 2 are the smallest and the second smallest ground terms, respectively. Then in the clause $x \simeq 1 \vee x \simeq 2$, the constant 2 can become strictly maximal only if

$x\sigma \equiv 1$, but that instance is a tautology. Hence none of the above inferences is necessary. As this example shows, much effort can be saved if instances are inspected carefully. But does one need constraint technology to detect such redundancies?

We will exploit the observation that lifting in the case of bounded domains can be made more precise: A variable needs to stand no longer for any ground term, but just for the finitely many digits that represent the domain. Our first contribution is that, conversely, inferences involving a most general unifier σ only have to be considered if the range of σ consists of variables and digits. In other words, no complex unifiers are needed; and inferences do not increase the number of variables. This improves upon what is obtained with the basicness restriction [NR95, BGLS95] in the general case. Secondly, for any non-ground inference we can easily determine those instantiations that satisfy its ordering constraints. Thirdly, redundancy also needs to refer to digit instances only, such that stronger simplifications become possible in some situations, but compatibility with the corresponding notion of standard superposition is mostly preserved. The price for these achievements is negligible: The cardinality-bounding axiom needs to be exchanged for its functional instances in order to not lose completeness.

This lifting modification applies to the family of superposition calculi. We show that soundness and refutational completeness are preserved, using a domain-specific calculus configuration as example: Non-Horn clauses are dealt with not by equality factoring, but by aggressive splitting, which is possible because variable disjointness can be forced via instantiation. The calculus implements a positive unit literal strategy [Der91]; and models can be extracted from saturated sets simply by ordered rewriting. Combining the latter with some instantiation, we obtain a decision procedure for satisfiability modulo the cardinality bound, which decides the Bernays-Schönfinkel class as well. We thereby solve a further classical decidability problem by superposition. Finally we show that the lifting modification is also applicable to bounded sorts in combination with arbitrary other, potentially infinite sorts.

The particular calculus configuration has been chosen because the positive unit literal strategy always terminates on ground Horn clauses, provided inferences are carried out as simplifications; and splitting extends this to non-Horn clauses. In [HTW06], we have encoded Sudoku puzzles as ground satisfiability problems for the SPASS theorem prover [WSH+07], which proceeding that way succeeded within a blink of an eye. Since exactly these ground-level inferences are lifted, we think that our method is promising in practice.

From a more abstract point of view, why should superposition for boun-

ded-domain problems be studied at all? The first-order theory of such domains is clearly decidable; and sophisticated model generators have been developed. Our main motivation, however, is that bounded domains often occur in combination with infinite ones. Think for example of finite enumeration types in programming languages, or any verification problem that involves a component with finite state space. As superposition is among the most powerful calculi for infinite-domain problems, it is natural to build specific bounded-domain reasoning technology into the calculus itself. Actually, our approach constitutes a light-weight adaptation rather than a deep integration, and should therefore be easy to implement within existing theorem provers.

Our approach complements that of finite-domain model generators like MACE [McC03] or PARADOX [CS03], which search for suitable interpretations in domains of increasing order. The problem of (finite) model computation has gained renewed interest, as witnessed by various contributions to IJCAR 2006. For example, new approaches via transformation into certain fragments of logic have been presented in [BS06] and [dNM06]. A variant tailored to instantiation-based methods is given in[BFdNT06]. The fruitful interplay of superposition and decision procedures is testified, for example, by [ARR01] and [BGN+06].

Compared to instantiation-based methods for bounded-domain problems say as in [McC03, CS03], our calculus does not a priori instantiate variables, but exploits the finiteness of the domain on the level of non-ground clauses. In particular, this offers advantages if the problem has structure that can be employed by inference and reduction rules. As a first example, not a single inference is possible between the two unit clauses $P(x_1, \ldots, x_k, x_1)$ and $\neg P(a, y_1, \ldots, y_{k-1}, b)$, but instantiation-based methods will generate more than n^k clauses for domain size n. In general, a superposition inference or simplification that involves variables simulates up to exponentially many ground steps. Likewise, proving one inference redundant may save an exponential amount of work. As a second example, consider an equation $f(x) \simeq x$ and an atom $P(f(g(y)))$, which standard rewriting would simplify to $P(g(c))$ because $f(g(y))$ is matched by $f(x)$. After instantiation with digits this reduction is no longer possible, as any term $g(\ldots)$ is not a digit. For examples of this form, inferring and simplifying at the non-ground level has the potential to exponentially shorten proofs and model representations.

Transformation-based methods [MB88, BS06] translate a given clause set into a form on which standard inference mechanisms like hyperresolution search for a model in a bottom-up way. This work is orthogonal to ours since it transforms the problem whereas we exploit the finiteness of the domain truly at the calculus level.

If the size of the domain is 2, then our calculus will constitute another method for reasoning with bits, which was the topic of Chap. 5. There, we have considered the standard bit operations as part of the theory at hand; and for a syntactically restricted class of literals, we have given such axioms that complex expressions could be solved with respect to their components, see for example $x \cdot y \simeq 0 \rightarrow x \simeq 0 \vee y \simeq 0$. The calculus presented here, however, is a variant of working with the original axiomatization of bits. One might want to combine both approaches, but restricting instances of solver-like axioms like the one obove to those in terms of digits seems counterproductive.

Our calculus can be combined with general first-order theories, which is currently supported neither by the instantiation-based nor by the transformation-based approach. In fact, bounded-domain sorts are an inherent part of many verification problems that arise from software or system analysis. Therefore, this combination has a large application potential.

In the beginning, we will use a single-sorted signature Σ that just contains the digits 1 through n, besides arbitrary other function symbols. As said, the theory $\mathcal{T}$ is given by the formula

$$\forall x.\, x \simeq 1 \vee \ldots \vee x \simeq n$$

In Sect. 6.3 we will introduce a superposition-based calculus to tackle the $\mathcal{T}$-unsatisfiability of clause sets over Σ. Note that this also covers the case that the domain size is exactly n if the input clause set contains equations $i \not\simeq j$ for any distinct $i, j \in [1; n]$.

6.2 Ground Horn Superposition

Here we recapitulate a superposition calculus $\mathcal{G}$ for ground Horn clauses [BG94, NR01]. In every clause with negative literals, at least one of them shall be *selected.* This eager selection leads to a positive unit literal strategy [Der91], where the side premise of superposition inferences is always a positive unit clause. Even more, the model construction involves such unit clauses only, which later will ease the model extraction in Sect. 6.3.4. From now on, let $\succ$ denote a reduction ordering total on ground terms. It is extended to clauses and clause multisets according to Def. 2.26.

Rules of calculus $\mathcal{G}$:

Ground superposition left

$$\mathcal{I}\,\frac{l \simeq r \quad s[l] \not\simeq t \vee C}{s[r] \not\simeq t \vee C} \qquad \text{if} \begin{array}{l} \cdot\ l \text{ and } s \text{ are strictly greatest} \\ \cdot\ s \not\simeq t \text{ is selected} \end{array}$$

Ground superposition right

$$\mathcal{I}\ \frac{l \simeq r \quad s[l] \simeq t}{s[r] \simeq t} \qquad \text{if } \begin{array}{l}\cdot\ l \text{ and } s \text{ are strictly greatest} \\ \cdot\ l \simeq r \prec s \simeq t\end{array}$$

Ground equality resolution

$$\mathcal{I}\ \frac{C \vee t \not\simeq t}{C} \qquad \text{if } \cdot\ t \not\simeq t \text{ is selected}$$

An inference with maximal premise C and conclusion D is *redundant* with respect to a clause set M if $M^{\prec C} \models D$, where $M^{\prec C}$ contains all elements of M smaller than C. The calculus $\mathcal{G}$ is sound and refutationally complete in the sense that $M \models \bot$ and $\bot \in M$ coincide for every saturated set M. The completeness proof relies on a model functor that associates with M a convergent ground rewrite system R. Let R^* denote the quotient of the free ground term algebra modulo the congruence generated by R; and assume that M is saturated and does not contain the empty clause. Then R^* is a model of M. In detail, for every clause C let $\mathrm{Gen}(C) = \{l \to r\}$ if (i) $C \equiv l \simeq r \in M$, (ii) l is strictly maximal, (iii) l is R_C-irreducible; and let $\mathrm{Gen}(C) = \{\}$ otherwise. Furthermore R_C is $\bigcup_{D \prec C} \mathrm{Gen}(D)$, and finally R is $\bigcup_D \mathrm{Gen}(D)$. Notably R^* is the unique minimal Herbrand model of M [BG91]. For ground terms l and r over Σ we have $M \models l \simeq r$ iff $R^* \models l \simeq r$ iff $l \downarrow_R r$.

Notably every inference conclusion makes the corresponding main premise redundant and hence can be turned into a simplification. This way the calculus decides satisfiability of finite ground Horn clause sets, which via splitting extends to the non-Horn case. Therefore it is an attractive basis for techniques to reason modulo $\mathcal{T}$.

6.3 A Calculus for $\mathcal{T}$-Unsatisfiability

6.3.1 Calculus Rules

We now introduce a calculus $\mathcal{C}$ that shall detect unsatisfiability modulo $\mathcal{T}$. It works on finite or infinite sets of arbitrary clauses, ground or non-ground. For a substitution τ we say that it *numbers* if $\mathrm{ran}\,\tau \subseteq [1; n]$, and that it in addition *minimally numbers* with respect to a set of conditions if these are satisfied with τ, but with no other numbering τ' more general than τ. Furthermore τ *ground numbers* a clause C if τ numbers and $C\tau$ is ground. The set of all ground instances of C under such substitutions is denoted by $\Omega(C)$, and its elements are called the Ω-*instances* of C.

A distinguishing feature of the calculus $\mathcal{C}$ shows up if more than one literal is maximal in a premise under the unifier: Then we instantiate just as much as is necessary with elements of $[1;n]$ to dissolve this ambiguity. So more conclusions are generated, but altogether they have fewer Ω-instances. In this sense, lifting is more precise than without instantiation.

As a second specialty, if a most general unifier is involved in an inference rule, then its range consists only of variables and digits. Hence many of the inferences in the standard calculus are not necessary here. For example, with the lexicographic path ordering [KL80] induced by the precedence $+ \succ s$, from the two clauses $(x+y)+z \simeq x+(y+z)$ and $u+s(v) \simeq s(u+v)$ one would normally obtain every $s^i(x+y)+z \simeq x+(s^i(y)+z)$. But since y needs to be bound to $s(v)$, no inference is drawn here.

Similar to the calculus $\mathcal{G}$, in every Horn clause with negative literals at least one of them shall be selected. Non-Horn clauses are subject to mandatory splitting. Different from the usual splitting rule, if a non-Horn clause cannot be split into two variable-disjoint parts, then we will split some instances instead. In order to minimize the number of splits, we assume that for every non-Horn clause a partitioning into two subclauses is *designated* where each subclause has strictly fewer positive literals, hence at least one. Furthermore we stipulate that from now on the smallest ground terms are the digits from $[1;n]$, say such that $n \succ \ldots \succ 1$.

Rules of calculus $\mathcal{C}$:

Superposition left

$$\mathcal{I}\,\frac{l \simeq r \quad s[l'] \not\simeq t \vee C}{(s[r] \not\simeq t \vee C)\sigma\tau}$$

if
- $l' \notin \mathcal{V}$ and $\sigma = \mathrm{mgu}(l, l')$
- $\operatorname{ran}\sigma \subseteq \mathcal{V} \cup [1;n]$
- τ minimally numbers such that l and s are strictly greatest under $\sigma\tau$
- $s \not\simeq t$ is selected
- C is Horn

Superposition right

$$\mathcal{I}\,\frac{l \simeq r \quad s[l'] \simeq t}{(s[r] \simeq t)\sigma\tau}$$

if
- $l' \notin \mathcal{V}$ and $\sigma = \mathrm{mgu}(l, l')$
- $\operatorname{ran}\sigma \subseteq \mathcal{V} \cup [1;n]$
- τ minimally numbers such that l and s are strictly greatest under $\sigma\tau$ and $(l \simeq r)\sigma\tau \prec (s \simeq t)\sigma\tau$

Equality resolution

$$\mathcal{I}\,\frac{C \vee t \not\simeq t'}{C\sigma} \quad \text{if} \quad \begin{array}{l} \cdot\ \sigma = \mathrm{mgu}(t, t') \\ \cdot\ \operatorname{ran}\sigma \subseteq \mathscr{V} \cup [1; n] \\ \cdot\ t \not\simeq t' \text{ is selected} \\ \cdot\ C \text{ is Horn} \end{array}$$

Split

$$\mathcal{S}\,\frac{C \vee s \simeq t \vee l \simeq r \vee D}{(C \vee s \simeq t)\tau \mid (l \simeq r \vee D)\tau} \quad \text{if} \quad \begin{array}{l} \cdot\ \text{the partitioning is designated} \\ \cdot\ \tau \text{ minimally numbers such that} \\ \quad \text{the conclusions share no variables} \end{array}$$

Regarding the two superposition rules, if for two given premises the number of substitutions τ that satisfy the side conditions is large, one could alternatively add only a single conclusion $(s[r] \not\simeq t \vee C)\sigma$ or $(s[r] \simeq t)\sigma$, respectively, which would not affect refutational completeness. The decision procedure that will be developed later relies on the former version, however.

We consider a clause C *redundant* with respect to a set M of clauses if $\Omega(M)^{\prec C\rho} \models C\rho$ holds for every ground numbering ρ; that is, if every Ω-instance of C follows from smaller clauses in $\Omega(M)$ already. By compactness and by finiteness of $\Omega(C)$, no more than a finite subset of M is necessary. *Simplification,* in its general form, is making a clause redundant by adding (zero or more) entailed smaller clauses. Here it is already enough if these conditions hold on the Ω-instances.

Simplification

$$\mathcal{R}\,\frac{C}{\vec{D}}\,M \quad \text{if} \quad \begin{array}{l} \cdot\ C \text{ is redundant w.r.t. } \vec{D}, M \\ \cdot\ \Omega(C, M) \models \bigwedge \Omega(\vec{D}) \\ \cdot\ \Omega(C) \succ \Omega(\vec{D}) \end{array}$$

An inference with premises $\vec{C}$, most general unifier σ, minimally numbering substitution τ (identity in case of equality resolution), and conclusion D is *redundant* with respect to a set M of clauses if for every ground numbering ρ we have $\Omega(M)^{\prec \max\{\vec{C}\sigma\tau\rho\}} \models D\rho$. In the standard superposition calculus, the notions of redundancy and simplification refer to all ground instances, not to Ω-instances only. We will show in Sect. 6.3.3 that the actual difference between the notions is small.

Derivations from a $\mathcal{T}$-unsatisfiable clause set M do not necessarily produce the empty clause. For example, on a domain of size $n = 2$ there are just four unary functions: a negation-like, two constant ones, and the identity. Each of these satisfies $f^3 = f$. Hence $\{f^3(c) \not\simeq f(c),\ 1 \not\simeq 2\}$ is $\mathcal{T}$-unsatisfiable although no calculus rule is applicable to these two disequations. We will

therefore consider derivations from $M \cup \mathcal{T}'$ where $\mathcal{T}'$ consists of the following clauses:

$$f(\vec{x}) \simeq 1 \vee \ldots \vee f(\vec{x}) \simeq n \qquad \text{for any } f \in \Sigma \setminus [1;n]$$

$\mathcal{T}'$ is weaker than $\mathcal{T}$ in the sense that the upper cardinality bound is only applied to function values, but satisfies the same universal formulae. There is an increase in the initial number of clauses, but this is outshined by the fact that no inferences with complex unifiers are necessary. Interestingly, within the Bernays-Schönfinkel class the set $\mathcal{T}'$ is empty, as we will demonstrate in Sect. 6.3.6.

6.3.2 Soundness and Refutational Completeness

The first proposition relates $\mathcal{T}$-satisfiability with satisfiability of Ω-instances and justifies the exchange of $\mathcal{T}'$ for $\mathcal{T}$, since all Ω-instances of $\mathcal{T}$ are tautologies.

Proposition 6.1 A clause set M is $\mathcal{T}$-satisfiable iff $\Omega(M \cup \mathcal{T}')$ is satisfiable.

Proof: On the one hand, since $M, \mathcal{T} \models \Omega(M \cup \mathcal{T}')$, every $\mathcal{T}$-model of M is a model of $\Omega(M \cup \mathcal{T}')$ as well. On the other hand, consider any model $\mathcal{A}$ of $\Omega(M \cup \mathcal{T}')$. Its restriction to $\{1^{\mathcal{A}}, \ldots, n^{\mathcal{A}}\}$ is a Σ-algebra because of the range restriction on the functions, and it is a $\mathcal{T}$-model by construction. Finally every clause C is $\mathcal{T}$-equivalent to $\bigwedge \Omega(C)$. □

Next one has to show that within a derivation, satisfiability is inherited from each parent node to one of its immediate descendants.

Proposition 6.2 Let N denote a node in a derivation, with successors $N_1, \ldots, N_k$. If $\Omega(N)$ is satisfiable, so is some $\Omega(N_i)$.

Proof: According to the type of calculus step, we distinguish three cases.

- An inference: Here k equals 1, and N_1 is $N \cup \{C\}$ where C is N-valid. Hence N and N_1 are even equivalent.
- A simplification adhering to the form $\mathcal{R}\frac{C}{\vec{D}}N'$: Again k is 1, but N has a presentation $N = \{C\} \cup N' \cup N''$ such that $N_1 = \{\vec{D}\} \cup N' \cup N''$. The side conditions imply $\Omega(N') \models (\bigwedge \Omega(C)) \leftrightarrow (\bigwedge \Omega(\vec{D}))$, such that the clause sets $\Omega(N)$ and $\Omega(N_1)$ are equivalent.
- A split: In our concrete split rule k equals 2. Let $C' \equiv (C \vee s \simeq t)\tau$ and $D' \equiv (l \simeq r \vee D)\tau$ denote the first and the second conclusion, respectively. Then $C' \vee D'$ is N-valid, and the disjuncts share no variables. If $\mathcal{A}$ is an N-model, then $\mathcal{A}$ satisfies at least one of C' and D', and therefore at least one of $N_1 = N \cup \{C'\}$ and $N_2 = N \cup \{D'\}$.

□

Accordingly, the clause set at the root is $\mathcal{T}$-satisfiable iff the derivation has a path each element of which is satisfiable.

Proposition 6.3 For every clause set M, the following are equivalent:
(i) M is $\mathcal{T}$-satisfiable.
(ii) Every derivation from $M \cup \mathcal{T}'$ contains a complete path $N_1, N_2, \ldots$ such that every $\Omega(N_i)$ is satisfiable.

Proof: If M is $\mathcal{T}$-satisfiable, then by Prop. 6.1 the set $\Omega(N_1) = \Omega(M \cup \mathcal{T}')$ is satisfiable, from which we can recursively construct a complete path as required by Prop. 6.2. The converse implication follows from $N_1 = M \cup \mathcal{T}$ by Prop. 6.1. □

If a clause C occurs at some point in a path, then the limit N_∞ entails each of its Ω-instances from smaller or equal Ω-instances. Furthermore satisfiability of N_∞ with respect to Ω-instances is the conjunction of this property over all path elements.

Proposition 6.4 Consider a complete path $N_1, N_2, \ldots$ in some derivation.
(i) If $C \in N_i$ is ground numbered by ρ, then $\Omega(N_\infty)^{\preceq C\rho} \models C\rho$ holds, as well as $\Omega(N_j)^{\preceq C\rho} \models C\rho$ for every $j \geq i$.
(ii) Every $\Omega(N_i)$ is satisfiable iff $\Omega(N_\infty)$ is.
(iii) N_∞ is saturated in case the derivation is fair.

Proof:
(i) The proof is by induction on $C\rho$ with respect to $\succ$. Let j denote ∞ or a natural number greater than or equal to i. If $C \in N_j$ we are done. Otherwise there is an index k between i and j such that C is contained in N_i through N_k, but not in N_{k+1}. By definition of simplification we have $\Omega(\vec{D}, M)^{\prec C\rho} \models C\rho$ for appropriate $\vec{D}, M \subseteq N_{k+1}$. Either $\vec{D}, M$ is empty and $C\rho$ is a tautology, or there is a greatest clause D' in $\Omega(\vec{D}, M)^{\prec C\rho}$. Inductively all elements of $\Omega(\vec{D}, M)^{\prec C\rho}$ are valid in $\Omega(N_j)^{\preceq D'}$, and so is $C\rho$.
(ii) Assume that every $\Omega(N_i)$ is satisfiable. By compactness $\Omega(N_\infty)$ is satisfiable iff each of its finite subsets is. Given one such subset M, for every Ω-instance $C\rho$ within there is an index j such that C is contained in N_j and all successors thereof. Since M is finite, these indices have a finite maximum k. Now $\Omega(N_k)$ comprises M and is satisfiable by assumption.
As to the converse implication, consider an Ω-instance $C\rho$ of a clause $C \in N_i$. Then $\Omega(N_\infty)$ entails $C\rho$ by Prop. 6.4 (i). In other words, any model of $\Omega(N_\infty)$ is a model of $\Omega(N_i)$.

(iii) Firstly we consider an inference with premises $\vec{C}$ from N_∞ and conclusion D with ground numbering substitution ρ. Because of fairness $\Omega(N_i)^{\prec\max\{\vec{C}\rho\}} \models D\rho$ holds for some i, which can be rephrased as $C'_1\rho_1, \ldots, C'_k\rho_k \models D\rho$ for clause instances $C'_j\rho_j$ from $\Omega(N_i)$ below $\max\{\vec{C}\rho\}$. By Prop. 6.4 (i) these clause instances are valid in $\Omega(N_\infty)$ below $\max\{\vec{C}\rho\}$, and so is $D\rho$.
Secondly we study a split from a persistent clause $C \equiv C_1 \vee C_2$ with designated partitioning as indicated and minimally numbering substitution τ. Because of fairness, one split conjunct, say $C_1\tau$, is contained in some N_i or redundant with respect to it. So either $C_1\tau$ is persistent, or $C_1\tau$ is redundant with respect to some N_j where $j \geq i$. In the former case the proof is finished. In the latter we have $\Omega(N_j)^{\prec C_1\rho} \models C_1\rho$ for every ground numbering $\rho = \tau\tau'$, which extends to $\Omega(N_\infty)^{\prec C_1\rho} \models C_1\rho$ with an argument like in the preceding paragraph.

□

For any clause set M, let $\widehat{M}$ denote its Ω-instances which are Horn clauses.

Proposition 6.5 $\Omega(M)$ and $\widehat{M}$ are equivalent for $\mathcal{C}$-saturated clause sets M.

Proof: We show by induction on clause instances that every non-Horn clause $C\rho \in \Omega(M)$ is entailed by $\widehat{M}$. Now, C has a presentation $C \equiv C_1 \vee C_2$ such that the partitioning into C_1 and C_2 is designated. Then ρ numbers the clause C such that the subclauses C_1 and C_2 are variable disjoint. More general such substitutions τ have to satisfy $\tau \subseteq \rho$. There exists a $\subset$-minimal such τ because all descending $\subset$-chains are finite. Then $C \vdash C_1\tau \mid C_2\tau$ is a valid $\mathcal{C}$-split. Because M is saturated, one split conjunct, say $C_1\tau$, is contained in M or redundant with respect to M. In both cases we have $\Omega(M) \models C_1\rho$, and we obtain inductively $\widehat{M} \models C_1\rho$. Finally $C_1\rho$ entails $C\rho$. □

The crucial lifting result is the following:

Proposition 6.6 If a clause set M is $\mathcal{C}$-saturated, then $\widehat{M}$ is $\mathcal{G}$-saturated.

Proof: We adapt the usual lifting arguments to our calculus, inspecting $\mathcal{G}$-inferences with premises from $\widehat{M}$. If a clause $D \in \widehat{M}$ contains negative literals, then let the literal selection be inherited from one arbitrary $C \in M$ that instantiates into D.

- Ground superposition right: Given two clauses $l \simeq r$ and $s \simeq t$ from M with ground numbering substitution ρ, consider the $\mathcal{G}$-inference with premises $l\rho \simeq r\rho$ and $s\rho[l\rho]_p \simeq t\rho$, and conclusion $s\rho[r\rho]_p \simeq t\rho$. The position p is within s because the range of ρ consists of digits only.

This $\mathcal{G}$-inference corresponds to a variable overlap if $s|_p \equiv x \in \mathcal{V}$, and to a non-variable overlap otherwise.
In the former case we have $x\rho \equiv l\rho$, such that $l\rho$ is a digit. Because $l\rho \succ r\rho$ and the digits are the smallest ground terms, the term $r\rho$ must be a digit as well. Let ρ' denote the substitution identical to ρ except that $x\rho' \equiv r\rho$. Then $(s \simeq t)\rho'$ is contained in $\Omega(M)$ and makes the inference redundant.
Now we come to non-variable overlaps. Let $l' \equiv s|_p$, furthermore $\sigma = \mathrm{mgu}(l, l')$ with $\mathrm{dom}\,\sigma \subseteq \mathrm{var}(l, l')$, and $\rho = \sigma\sigma'$. Because ρ is ground numbering, we know that $x\rho$ is a digit for every $x \in \mathrm{dom}\,\sigma$. Given $\rho = \sigma\sigma'$, every $x\sigma$ is either a digit or a variable.
The substitution σ' numbers the clauses $s\sigma \simeq t\sigma$ and $l\sigma \simeq r\sigma$ such that the literals $l\sigma$ and $s\sigma$ are greatest under σ', respectively, and that $(l \simeq r)\sigma\sigma' \prec (s \simeq t)\sigma\sigma'$. If τ is a more general such substitution, then it satisfies $\mathrm{dom}\,\tau \subseteq \mathrm{dom}\,\sigma'$ and $x\tau \equiv x\sigma'$ for every $x \in \mathrm{dom}\,\tau$, which implies $\tau \subseteq \sigma'$. There exists a $\subset$-minimal such τ because all descending $\subset$-chains are finite. Summing it up: $l \simeq r,\ s[l'] \simeq t \vdash (s[r] \simeq t)\sigma\tau$ is a $\mathcal{C}$-inference with premises from M, and is redundant with respect to M because M is saturated. If $\sigma' = \tau\tau'$, then the inference instance under τ' is redundant with respect to $\Omega(M)$.

- Ground equality resolution: Consider a Horn clause $C \vee t \not\simeq t' \in M$ with ground numbering substitution ρ such that $C\rho \vee t\rho \not\simeq t'\rho \vdash C\rho$ is a $\mathcal{G}$-inference. We may assume that $t \not\simeq t'$ is selected in $C \vee t \not\simeq t'$. As usually, t and t' have a most general unifier σ, which specializes into ρ say via σ'. We obtain $\mathrm{cdom}\,\sigma \subseteq \mathcal{V} \cup [1; n]$ like for ground superposition right. So $C \vee t \not\simeq t' \vdash C\sigma$ is a $\mathcal{C}$-inference with premises from M; and its redundancy carries over to that of the above instance.
- Ground superposition left: similar to ground superposition right, but taking selectedness into account like for ground equality resolution.

□

Putting everything together, the calculus $\mathcal{C}$ is sound and refutationally complete:

Lemma 6.7 For every clause set M, the following are equivalent:
(i) M is $\mathcal{T}$-satisfiable.
(ii) Every fair derivation from $M \cup \mathcal{T}'$ contains a complete path $N_1, N_2, \ldots$ such that the empty clause is not in N_∞.

Proof: We successively transform the first characterization into the second. By Prop. 6.3 the clause set M is $\mathcal{T}$-satisfiable iff there exists a complete path $N_1, N_2, \ldots$ such that every $\Omega(N_i)$ is satisfiable, or such that $\Omega(N_\infty)$ is, by

Prop. 6.4 (ii). Because of Prop. 6.4 (iii) every N_∞ is saturated with respect to $\mathcal{C}$. Hence by Prop. 6.5 the sets $\Omega(N_\infty)$ and $\widehat{N_\infty}$ are equivalent, and the latter is saturated with respect to $\mathcal{G}$. Since $\mathcal{G}$ is sound and complete, the satisfiability of $\widehat{N_\infty}$ is equivalent to $\bot \notin \widehat{N_\infty}$, which is the same as $\bot \notin N_\infty$. □

Notably the minimality of the digits is indispensable for refutational completeness: Assume that $\succ$ is the lexicographic path ordering induced by the precedence $n \succ \ldots \succ 1 \succ f \succ c$. Then from the unsatisfiable clause set $\{f(x) \simeq 1,\ 1 \simeq c,\ 1 \not\simeq f(c)\}$ nothing but the clause $f(c) \not\simeq c$ is inferable. We needed this minimality in the proposition on lifting to show that variable overlaps are non-critical; and indeed the variable overlap from $1 \simeq c$ into $f(x) \simeq 1$ would produce $f(c) \simeq 1$ and eventually lead to the empty clause.

6.3.3 Redundancy in Detail

In the calculus $\mathcal{C}$, redundancy on the general level is defined via redundancy of Ω-instances on the ground level, whereas in standard superposition one goes back to redundancy of all ground instances. In the sequel we analyze the resulting difference as to redundancy of clauses. Similar considerations apply to redundancy of inferences.

Let us compare under which conditions a clause C is redundant with respect to a clause set M. In the calculus $\mathcal{C}$ we require $\Omega(M)^{\prec C\rho} \models C\rho$ for every ground numbering ρ. The condition in standard superposition is $\mathrm{gnd}(M)^{\prec C\sigma} \models C\sigma$ for every ground substitution σ, where $\mathrm{gnd}(M)$ denotes the set of all ground instances of M. So for redundancy in the sense of $\mathcal{C}$ fewer instances need to be shown redundant, but on the other hand there are fewer premises for doing so. For example, $f(g(1)) \simeq 1$ is not redundant with respect to $f(x) \simeq 1$, since it is not entailed from $f(1) \simeq 1, \ldots, f(n) \simeq 1$. Fortunately, in $\mathcal{C}$-derivations the set M with respect to which redundancy is studied always contains the clauses of $\mathcal{T}'$, possibly simplified. Therefore we additionally have $g(1) \simeq 1 \vee \ldots \vee g(1) \simeq n$ at hand, with which $f(g(1)) \simeq 1$ does become redundant.

In this subsection, we develop two results that generalize this observation. Firstly, if every digit instance $C\rho$ is entailed from smaller ground instances of M except some problematic ones, then C is redundant in the sense of $\mathcal{C}$. Secondly, if every $C\rho$ follows from arbitrary smaller ground instances, but C is not of a particular form, then C is also redundant. We employ these results to adapt one concrete simplification to our calculus, namely unit rewriting. The subsection ends with a demonstration that $\mathcal{C}$ should not be mixed with the standard notion of redundancy.

Deducing Ground Instances from Digit Instances

In the following we will prove that a ground instance $C\sigma$ of a clause C follows from $\Omega(C, \mathcal{T}')$, and give a criterion when this entailment is from smaller instances. We reserve the identifier f for non-digit function symbols, whereas i, j, k denote digits and $\vec{\imath}$ a vector thereof. For any term t, let $\mathrm{Dig}(t)$ denote the clause $t \simeq 1 \vee \ldots \vee t \simeq n$.

Proposition 6.8 For every clause C and term t, the following entailment holds: $C\{x \mapsto 1\}, \ldots, C\{x \mapsto n\}, \mathrm{Dig}(t) \models C\{x \mapsto t\}$

Proof: Consider a model $\mathcal{A}$ of the premises. Then there exists a digit i fulfilling $\mathcal{A} \models t \simeq i$. This identity inductively lifts to term contexts, and as equivalence to clause contexts. In particular $\mathcal{A} \models C\{x \mapsto i\}$ implies $\mathcal{A} \models C\{x \mapsto t\}$. □

Proposition 6.9 Let C denote a clause with ground substitution $\sigma = \{x_1 \mapsto t_1, \ldots, x_m \mapsto t_m\}$. Then $\Omega(C), \mathrm{Dig}(t_1), \ldots, \mathrm{Dig}(t_m) \models C\sigma$ holds.

Proof: The proof is by induction on m. If σ is the identity we are done. Otherwise we decompose σ according to $\sigma = \{x_1 \mapsto t_1, \ldots, x_m \mapsto t_m\} \cup \{x_{m+1} \mapsto t_{m+1}\} = \sigma_1 \cup \sigma_2$. Since the substitutions are ground we have $\sigma_1 \cup \sigma_2 = \sigma_1 \circ \sigma_2$. Inductively we obtain $\Omega(C\sigma_1), \mathrm{Dig}(t_1), \ldots, \mathrm{Dig}(t_m) \models C\sigma_1$. Proposition 6.8 gives $C\sigma_1, \mathrm{Dig}(t_{m+1}) \models C\sigma_1\sigma_2$. □

Proposition 6.10 Ground terms t obey $\Omega(\mathcal{T}') \models \mathrm{Dig}(t)$.

Proof: We induct on the structure of t. In case $t \equiv i$ the clause $\mathrm{Dig}(t)$ is a tautology. In case $t \equiv f(\vec{t})$ the proposition $\Omega(\mathcal{T}') \models \mathrm{Dig}(t_j)$ is inductively true for every j. Furthermore $\mathcal{T}'$ contains $\mathrm{Dig}(f(\vec{x}))$. Let $\sigma = \{x_1 \mapsto t_1, \ldots, x_m \mapsto t_m\}$, such that $f(\vec{t}) \equiv f(\vec{x})\sigma$. With Prop. 6.9 we obtain $\Omega(\mathrm{Dig}(f(\vec{x}))), \mathrm{Dig}(t_1), \ldots, \mathrm{Dig}(t_m) \models \mathrm{Dig}(f(\vec{x}))\sigma$. □

Proposition 6.11 $\Omega(C, \mathcal{T}') \models C\sigma$ is true for every clause C with ground substitution σ.

Proof: Assume $\sigma = \{x_1 \mapsto t_1, \ldots, x_m \mapsto t_m\}$. Then Prop. 6.10 implies $\Omega(\mathcal{T}') \models \mathrm{Dig}(t_i)$ for every i, such that from Prop. 6.9 finally we obtain $\Omega(C), \mathrm{Dig}(t_1), \ldots, \mathrm{Dig}(t_m) \models C\sigma$. □

We have seen in Prop. 6.10 that every ground term t is subject to $\Omega(\mathcal{T}') \models \mathrm{Dig}(t)$. In the following we will exploit that usually not all of $\Omega(\mathcal{T}')$ is needed for this entailment. There exist subsets $T \subseteq \Omega(\mathcal{T}')$ such that $T \models \mathrm{Dig}(t)$

holds. By compactness there are finite such T even in case the signature is infinite. Let $\Delta(t)$ denote the smallest of these finite T, with respect to the ordering on clause sets. Let furthermore $\delta(t)$ denote the greatest clause in $\Delta(t) \cup \{\bot\}$, and for ground substitutions σ let $\delta(\sigma)$ stand for the greatest clause in $\delta(\operatorname{ran}\sigma) \cup \{\bot\}$. Actually one can construct $\Delta(t)$ recursively, but this is not necessary for our purposes.

Proposition 6.12 Entailment from $\Omega(\mathcal{T}')$ can be restricted by the bounds $\delta(t)$ and $\delta(\sigma)$:

(i) Every ground term t satisfies $\Omega(\mathcal{T}')^{\preceq\delta(t)} \models \mathrm{Dig}(t)$.

(ii) If σ is a ground substitution for C, then $\Omega(C), \Omega(\mathcal{T}')^{\preceq\delta(\sigma)} \models C\sigma$ holds.

Proof:

(i) By definition we have $\Delta(t) \subseteq \Omega(\mathcal{T}')^{\preceq\delta(t)}$ and $\Delta(t) \models \mathrm{Dig}(t)$.

(ii) Let $\sigma = \{x_1 \mapsto t_1, \ldots, x_m \mapsto t_m\}$. Then we obtain $\Omega(\mathcal{T}')^{\preceq\delta(t_i)} \models \mathrm{Dig}(t_i)$ from Prop. 6.12 (i) for every i, and $\Omega(\mathcal{T}')^{\preceq\delta(\sigma)} \models \mathrm{Dig}(t_i)$ by definition of $\delta(\sigma)$. Finally we apply Prop. 6.9 to C and σ.

□

Proposition 6.13 For ground terms t we have $\delta(t) \equiv \bot$ iff t is a digit.

Proof: In case t is a digit, then $\mathrm{Dig}(t)$ is a tautology and $\Delta(t)$ is empty. Otherwise $\mathrm{Dig}(t)$ is not a tautology. □

Proposition 6.14 If t is a ground term and δ a ground substitution, then we can give estimates for $\delta(t)$ and $\delta(\sigma)$ as follows:

(i) $\delta(t) \equiv \mathrm{Dig}(u)$ implies $t \succeq u$.

(ii) $\delta(\sigma) \equiv \mathrm{Dig}(u)$ entails $\max(\operatorname{ran}\sigma) \succeq u$.

Proof:

(i) The proof is by induction on the term structure. If t is a digit, then we have $\delta(t) \equiv \bot$ by Prop. 6.13, and there is nothing to show. The case $t \equiv f(\vec{t})$ remains. Let $i_1, \ldots, i_k$ denote exactly the indices for which t_j is not a digit, and let $t' \equiv f(\vec{t})[x_1]_{i_1} \ldots [x_k]_{i_k}$. So t' is obtained from t replacing every non-digit t_j with a fresh variable. Conversely, using $\sigma = \{x_1 \mapsto t_{i_1}, \ldots, x_k \mapsto t_{i_k}\}$ one can instantiate t' back into t again.
In case $k = 0$ the argument vector $\vec{t}$ contains only digits. Choosing $T = \{\mathrm{Dig}(t)\}$ implies $T \subseteq \Omega(\mathcal{T}')$ and $T \models \mathrm{Dig}(t)$. Therefore we have $T \succeq \Delta(t)$ and $\max T \succeq \max \Delta(t) \equiv \delta(t)$, hence $\mathrm{Dig}(t) \succeq \mathrm{Dig}(u)$ and finally $t \succeq u$.
In case $k > 0$ every $\delta(t_{i_j})$ is distinct from $\bot$ by Prop. 6.13, and there exists a ground term v such that $\mathrm{Dig}(v) \equiv \max_j \delta(t_{i_j})$. By induction

hypothesis and the subterm property of t we obtain $t \succ v$. Here we choose $T = \Omega(\mathrm{Dig}(t')) \cup \Omega(\mathcal{T}')^{\preceq \mathrm{Dig}(v)}$, which satisfies $T \subseteq \Omega(\mathcal{T}')$. By construction $T \models \mathrm{Dig}(t_{i_j})$ holds for every j. Proposition 6.9 yields $\Omega(\mathrm{Dig}(t')), \mathrm{Dig}(t_{i_1}), \ldots, \mathrm{Dig}(t_{i_k}) \models \mathrm{Dig}(t'\sigma)$. Hence we may conclude that $T \succeq \Delta(t)$ and $\max T \succeq \mathrm{Dig}(u)$. Next we compare T with $\{\mathrm{Dig}(t)\}$. We have $\Omega(\mathrm{Dig}(t')) \prec \{\mathrm{Dig}(t)\}$ by minimality of the digits, and furthermore $\Omega(\mathcal{T}')^{\preceq \mathrm{Dig}(v)} \prec \{\mathrm{Dig}(t)\}$ because of $v \prec t$. Hence we may conclude that $\mathrm{Dig}(t) \succ \max T \succeq \mathrm{Dig}(u)$ holds, such that $t \succ u$ is true.

(ii) Let $\sigma = \{x_1 \mapsto t_1, \ldots, x_m \mapsto t_m\}$. Because of $\delta(\sigma) \not\equiv \bot$ we have $\delta(\sigma) \equiv t_i$ for some i. Using Prop. 6.14 (i) we may conclude that $\max_j t_j \succeq t_i \succeq u$ holds.

□

Given a clause C with ground substitution σ, we call the pair C, σ *problematic* if $x\sigma \equiv f(\vec{\imath})$ for some $x \in \mathrm{var}(C)$ and $C\sigma \preceq \mathrm{Dig}(f(\vec{\imath}))$. Otherwise the pair is called *unproblematic*. Let furthermore denote $\mathring{\mathrm{gnd}}(C)$ the set of all ground instances $C\sigma$ for which C, σ is unproblematic, and let $\mathring{\mathrm{gnd}}$ extend to clause sets in the usual way.

Here are two necessary and quite restrictive conditions for C, σ to be problematic: Firstly some variable $x \in \mathrm{var}(C)$ may occur only in literals of the form $x \simeq i$ and $x \simeq y$. Secondly the greatest literal of $C\sigma$ must have the form $f(\vec{\imath}) \simeq j$.

Proposition 6.15 Let C denote a clause with ground substitution σ such that σ is not numbering, and that C, σ is unproblematic. Then the entailment $\Omega(C, \mathcal{T}')^{\prec C\sigma} \models C\sigma$ holds.

Proof: We decompose $\sigma = \sigma_1 \cup \sigma_2$ such that the range of σ_1 contains only digits and the range of σ_2 only non-digits. Since the substitutions are ground we have $\sigma = \sigma_1 \circ \sigma_2$. Proposition 6.12 (ii) implies $\Omega(C\sigma_1), \Omega(\mathcal{T}')^{\preceq \delta(\sigma_2)} \models C\sigma_1\sigma_2$. The substitution σ_2 is not empty because σ is not numbering. Hence we have by minimality of the digits $\Omega(C\sigma_1) \prec \{C\sigma_1\sigma_2\}$. We still have to show $\delta(\sigma_2) \prec C\sigma$. Let t denote the greatest term in $\operatorname{ran}\sigma_2$. By Prop. 6.13 the clause $\delta(t)$ equals $\mathrm{Dig}(f(\vec{\imath}))$ for some term $f(\vec{\imath})$. By Prop. 6.14 (ii) we have $t \succeq f(\vec{\imath})$. If $t \succ f(\vec{\imath})$, then the greatest term of $C\sigma$ is above the greatest of $\delta(\sigma_2)$. Otherwise we obtain $C\sigma \succ \mathrm{Dig}(f(\vec{\imath}))$ from the requirement that C, σ is unproblematic. □

Using Ground Instances in Redundancy Proofs

We have seen in the preceding proposition that unproblematic ground instances of clauses follow from smaller digit instances of the same clause and

of $\mathcal{T}'$. Hence these ground instances can safely be used in redundancy proofs as if they were digit instances. Alternatively we study for which clauses it is safe to use arbitrary ground instances when showing them redundant. A clause C is called *critical* if it has an Ω-instance $C\rho$ with greatest term $f(\vec{\imath})$ such that $C\rho \preceq \mathrm{Dig}(f(\vec{\imath}))$. Otherwise C is called *noncritical.*

Lemma 6.16 Consider a path in a $\mathcal{C}$-derivation from $M \cup \mathcal{T}'$ to N and a clause C. Then C is redundant with respect to N if one of the following conditions holds, where ρ ranges over all ground numbering substitutions:

(i) $\mathring{\mathrm{gnd}}(N)^{\prec C\rho} \models C\rho$ for all ρ,

(ii) $\mathrm{gnd}(N)^{\prec C\rho} \models C\rho$ for all ρ and C is noncritical.

Proof:

(i) Given an arbitrary ground numbering substitution ρ, there exist clauses $D_1, \ldots, D_m \in N$ and ground substitutions $\sigma_1, \ldots, \sigma_m$ such that every D_i, σ_i is unproblematic and $D_i\sigma_i \prec C\rho$, and that $D_1\sigma_1, \ldots, D_m\sigma_m \models C\rho$. In order to prove $\Omega(N)^{\prec C\rho} \models C\rho$ it suffices to show that $\Omega(N)^{\prec C\rho} \models D_i\sigma_i$ holds for every i. If $D_i\sigma_i$ is a digit instance of D_i, then we have $D_i\sigma_i \in \Omega(N)^{\prec C\rho}$. Otherwise Prop. 6.15 ensures $\Omega(D_i, \mathcal{T}')^{\prec D_i\sigma_i} \models D_i\sigma_i$ because D_i, σ_i is unproblematic. With Prop. 6.4 (i) we get $\Omega(N)^{\prec D_i\sigma_i} \models D_i\sigma_i$, and therefore $\Omega(N)^{\prec C\rho} \models D_i\sigma_i$.

(ii) Similar to the proof of Lem. 6.16 (i), for every ground numbering substitution ρ there exist clauses $D_1, \ldots, D_m \in N$ and ground substitutions $\sigma_1, \ldots, \sigma_m$ such that always $D_i\sigma_i \prec C\rho$, and that $D_1\sigma_1, \ldots, D_m\sigma_m \models C\rho$. If $C\rho$ is a tautology we are done. Otherwise we decompose every $\sigma_k = \sigma'_k \cup \sigma''_k$ such that the range of σ'_k contains only digits and the range of σ''_k only non-digits. Proposition 6.12 (ii) guarantees that $\Omega(D_k\sigma'_k), \Omega(\mathcal{T}')^{\preceq \delta(\sigma''_k)} \models D_k\sigma_k$. By minimality of the digits we obtain $\Omega(D_k\sigma'_k) \preceq \{D_k\sigma_k\} \prec \{C\rho\}$.

Next we show that $\delta(\sigma''_k) \prec C\rho$. The clause C is not empty since otherwise $\models \bot$; so $C\rho$ has a greatest term s. Let t denote the greatest term of $D_k\sigma_k$, then we have $s \succeq t$. If $\delta(\sigma''_k) \equiv \bot$ then $\bot \prec C\rho$. Otherwise $\delta(\sigma''_k)$ has the shape $\mathrm{Dig}(f(\vec{\imath}))$. Because of Prop. 6.14 (ii) we have $\max(\operatorname{ran}\sigma''_k) \succeq f(\vec{\imath})$, and because of $t \succeq \max(\operatorname{ran}\sigma''_k)$ we have $s \succeq f(\vec{\imath})$ as well. Now $s \succ f(\vec{\imath})$ directly entails $C\rho \succ \delta(\sigma''_k) \equiv \mathrm{Dig}(f(\vec{\imath}))$. Otherwise s equals $f(\vec{\imath})$, and $C\rho \succ \mathrm{Dig}(f(\vec{\imath}))$ holds because C is noncritical by assumption.

Summing it up, we obtain $\Omega(D_k\sigma'_k, \mathcal{T}')^{\prec C\rho} \models D_k\sigma_k$ and therefore as well $\Omega(D_k, \mathcal{T}')^{\prec C\rho} \models D_k\sigma_k$. Via Prop. 6.4 (i) we conclude $\Omega(N)^{\prec C\rho} \models D_k\sigma_k$. □

Application: Unit Rewriting

Ordered rewriting with respect to a set of unit equations straightforwardly extends from terms to clauses. However it is a simplification in the sense of the calculus $\mathcal{C}$ only if the clause to be simplified is above the simplifying equation instances. For example, $f(3) \simeq 1 \rightarrow_{\{f(3)\simeq 2\}} 2 \simeq 1$ is a rewrite step, but not a simplification, because the clause to be rewritten is smaller than the one used for rewriting.

In order to meet the requirements of Lem. 6.16, a further condition is necessary. Rewriting $C[s\sigma]$ into $C[t\sigma]$ is called Ω-*admissible* if one of the following conditions applies:

(i) C is noncritical.

(ii) C is critical, i. e., it contains literals of the shape $f(\vec{s}) \simeq t$ where t and every s_i is a digit or a variable, such that with a suitable ground numbering ρ the term $f(\vec{s})\rho$ is the greatest of $C\rho$ and $C\rho \preceq \mathrm{Dig}(f(\vec{s})\rho)$ holds: Then rewrite steps on such $f(\vec{s})$ with equations $x \simeq i$ or $x \simeq y$ only take place below f.

Proposition 6.17 Given a path in a $\mathcal{C}$-derivation from $M \cup \mathcal{T}'$ to $N \cup \{C\}$ and an equation $s \simeq t \in N$, the following is an instance of $\mathcal{C}$-simplification:

Ordered unit rewriting

$$\mathcal{R}\,\frac{C[s\sigma]}{C[t\sigma]}\,N \qquad \text{if} \quad \begin{array}{l} \cdot\ s\sigma \succ t\sigma \\ \cdot\ C\rho \succ (s \simeq t)\sigma\rho \text{ for every} \\ \quad \text{ground numbering substitution } \rho \\ \cdot\ \text{the rewrite step is } \Omega\text{-admissible} \end{array}$$

Proof: Let $C' \equiv C[t\sigma]$. According to the definition of simplification, we have to show that three conditions are fulfilled.

(i) C is redundant with respect to C', N:
Consider an arbitrary ground numbering substitution ρ. Clearly we have $C'\rho, (s \simeq t)\sigma\rho \models C\rho$. Now $C\rho \succ C'\rho$ is valid because of the first requirement $s\sigma \succ t\sigma$. The second requirement guarantees $C\rho \succ (s \simeq t)\sigma\rho$. Hence the following holds:

$$(C'\rho, (s \simeq t)\sigma\rho)^{\prec C\rho} \models C\rho \qquad (*)$$

The crucial point is that $(s \simeq t)\sigma\rho$ is not an Ω-instance of $s \simeq t$ in general. By the third requirement, the rewrite step is Ω-admissible, such that two cases are possible:

(a) C is noncritical: Then from $(*)$ we obtain $\mathrm{gnd}(C', N)^{\prec C\rho} \models C\rho$ because of $s \simeq t \in N$. Hence C is redundant by Lem. 6.16 (ii).

(b) C is critical: The pair C', ρ is unproblematic because ρ is numbering. If $s \simeq t, \sigma\rho$ is unproblematic as well, then $(*)$ entails $\mathring{\text{gnd}}(C', N)^{\prec C\rho} \models C\rho$, such that $C\rho$ is redundant with respect to C', N by Lem. 6.16 (i).
Otherwise $s \simeq t, \sigma\rho$ is problematic. Hence the equation $s \simeq t$ has one of the shapes $x \simeq y$ or $x \simeq j$, and the instance $(s \simeq t)\sigma\rho$ can be written as $f(\vec{\imath}) \simeq j$. Furthermore $\Omega(s \simeq t)$ identifies all digits, such that we may conclude the following:

$$\Omega(s \simeq t), f(\vec{\imath}) \simeq 1 \vee \ldots \vee f(\vec{\imath}) \simeq n \models f(\vec{\imath}) \simeq j \qquad (+)$$

The clause $\text{Dig}(f(\vec{x}))$ is contained in $\mathcal{T}'$. By Prop. 6.4 (i) we get $\Omega(N)^{\preceq \text{Dig}(f(\vec{\imath}))} \models \text{Dig}(f(\vec{\imath}))$. Since the clause C is critical, it contains non-digit function symbols, such that $\Omega(s \simeq t) \prec \{C\rho\}$ is true. Because the rewrite step is Ω-admissible, either $f(\vec{\imath})$ is not the greatest term of $C\rho$, or $C\rho \succ \text{Dig}(f(\vec{\imath}))$ holds. In the former case we have $\text{Dig}(f(\vec{\imath})) \prec C\rho$ as well. Consequently $(+)$ implies $\Omega(N)^{\prec C\rho} \models (s \simeq t)\sigma\rho$, which with $(*)$ leads us to $\Omega(C', N)^{\prec C\rho} \models C\rho$. So $C\rho$ is redundant with respect to C', N.

(ii) $\Omega(C, N) \models \bigwedge \Omega(C')$:
If ρ ground numbers $C\sigma$, then $C\rho, (s \simeq t)\sigma\rho \models C'\rho$ holds. Proposition 6.11 ensures $\Omega(s \simeq t, \mathcal{T}') \models (s \simeq t)\sigma\rho$, such that $\Omega(N) \models (s \simeq t)\sigma\rho$ is true via Prop. 6.4 (i).

(iii) $\Omega(C) \succ \Omega(C')$:
Let ρ ground number C, and hence C' as well, such that every variable of the domain is mapped onto n. Then $C[s\sigma]\rho$ and $C[t\sigma]\rho$ are the greatest clauses of $\Omega(C)$ and $\Omega(C')$, respectively, and by assumption $s\sigma\rho \succ t\sigma\rho$ holds.

□

Composing Instantiation and Simplification

Unit rewriting on the non-ground level can be inapplicable although it would be possible on every Ω-instance: If $n = 2$ and $N = \{f(1) \simeq 2, f(2) \simeq 1, g(f(f(x))) \simeq g(x)\}$, then the third equation cannot be rewritten, but its Ω-instances could be turned into the tautologies $g(1) \simeq g(1)$ or $g(2) \simeq g(2)$, respectively. However, via composition of instantiation and simplification one obtains a simplification again, which we will show in this subsection. As to our calculus, instantiation always preserves finiteness, because it is with respect to digits only.

If C is a clause and Γ a set of numbering substitutions with $\operatorname{dom}\tau \subseteq \operatorname{var}(C)$ for every $\tau \in \Gamma$, then we say that Γ *covers* C if every ρ that ground numbers C can be obtained as specialization of some $\tau \in \Gamma$.

Proposition 6.18 If a clause C is covered by Γ such that $\mathcal{R}\,\dfrac{C\tau}{M_\tau}\,N_\tau$ is a simplification for every $\tau \in \Gamma$, then $\mathcal{R}\,\dfrac{C}{\bigcup_\tau M_\tau}\,\bigcup_\tau N_\tau$ is also a simplification.

Proof: Let $M = \bigcup_\tau M_\tau$ and $N = \bigcup_\tau N_\tau$. By definition of simplification, every M_τ is finite, and so is M. We need to prove that three conditions are satisfied:

(i) C is redundant with respect to M, N: If ρ ground numbers C, then $\rho = \tau\tau'$ for some $\tau \in \Gamma$. By assumption $C\tau$ is redundant with respect to $M_\tau \cup N_\tau$, hence $\Omega(M_\tau, N_\tau)^{\prec C\tau\tau'} \models C\tau\tau'$ holds. By set inclusion we obtain $\Omega(M, N)^{\prec C\rho} \models C\rho$.

(ii) $\Omega(C, N) \models \bigwedge \Omega(M)$: By assumption we have $\Omega(C\tau, N_\tau) \models \bigwedge \Omega(M_\tau)$ for every $\tau \in \Gamma$. This extends to $\bigcup_\tau \Omega(C\tau, N_\tau) \models \bigwedge_\tau \Omega(M_\tau)$, from which the condition at hand follows via $\bigcup_\tau \Omega(C\tau, N_\tau) = \Omega(C, N)$ and $\bigwedge_\tau \Omega(M_\tau) = \bigwedge \Omega(\bigcup_\tau M_\tau)$.

(iii) $\Omega(C) \succ \Omega(M)$: This follows from $\Omega(C\tau) \succ \Omega(M_\tau)$ for every $\tau \in \Gamma$.

□

We can apply this result to unit rewriting and obtain a derived simplification that will be used in the formation of a decision procedure to enforce termination.

Corollary 6.19 Consider a path in a $\mathcal{C}$-derivation from $M \cup \mathcal{T}'$ to $N \cup \{C\}$. Then the following is an instance of $\mathcal{C}$-simplification:

Instance rewriting

$$\mathcal{R}\,\frac{C}{\{D_\tau : \tau \in \Gamma\}}\,N \qquad \text{if} \begin{array}{l} \cdot\ \Gamma \text{ covers } C \\ \cdot\ \text{for every } \tau \in \Gamma\text{: } \mathcal{R}\,\dfrac{C\tau}{D_\tau}\,N \text{ is an} \\ \quad \text{ordered unit rewriting step} \end{array}$$

Incompatibility of $\mathcal{C}$ with Standard Redundancy

The difference of our redundancy notion to the one of standard superposition may show up in practice: Assume $n = 2$ and some input M which via $\mathcal{C}$ eventually leads to the clause set $N = \{x \simeq 1,\ f(1) \simeq 2,\ f(2) \simeq 2,\ f(1) \not\simeq 1\}$. Now the clause $x \simeq 1$ has the ground instances $2 \simeq 1$ and $f(1) \simeq 1$ which make the second and the third clause redundant in the standard sense. Since

$f(1) \simeq 1$ is not an Ω-instance of $x \simeq 1$, these clauses are not redundant in the sense of $\mathcal{C}$.

Going further, the example shows that combining $\mathcal{C}$ with standard redundancy is problematic: If $f(1) \simeq 2$ and $f(2) \simeq 2$ were deleted from N, then the rest $\{x \simeq 1,\ f(1) \not\simeq 1\}$ would be $\mathcal{C}$-saturated, despite the apparent unsatisfiability. Summing it up, refutational completeness would be lost. However, because of Lem. 6.16 only in rare cases standard redundancy is stronger than redundancy in the sense of $\mathcal{C}$.

Notably the opposite relation can be observed as well: Let $n = 2$, $C \equiv x \simeq y \vee f(1) \simeq y$ and $N = \{f(1) \simeq 1 \vee f(1) \simeq 2,\ f(2) \simeq 1 \vee f(2) \simeq 2,\ 1 \simeq 2,\ C\} \supseteq \mathcal{T}'$. The clause C is redundant in the sense of $\mathcal{C}$ because $C\rho$ is a tautology if $x\rho \equiv y\rho$, and because otherwise $C\rho$ is subsumed by $1 \simeq 2$. However C is not redundant in the standard sense: Consider the ground instance $C\sigma \equiv f(1) \simeq 1 \vee f(1) \simeq 1$. We obtain $\mathrm{gnd}(N)^{\prec C\sigma} = \{1 \simeq 2,\ 1 \simeq 1 \vee f(1) \simeq 1,\ 2 \simeq 1 \vee f(1) \simeq 1\}$, but $1 \simeq 2 \not\models f(1) \simeq 1$. One cannot hold the exchange of $\mathcal{T}'$ for $\mathcal{T}$ responsible for this phenomenon, since it also occurs in case of $N' = \{x \simeq 1 \vee x \simeq 2,\ 1 \simeq 2,\ C\}$.

6.3.4 Model Extraction

Here we study the case that a fair derivation from $M \cup \mathcal{T}'$ contains a complete path $N_1, N_2, \ldots$ without the empty clause. Let R denote the rewrite system that the superposition model functor of Sect. 6.2 produces from $\widehat{N_\infty}$. Then R^* is a witness that M is $\mathcal{T}$-satisfiable:

Proposition 6.20 R^* is subject to the following properties:

(i) $C \in N_i$ implies $R^* \models \bigwedge \Omega(C)$.
(ii) For every ground term t there exists some digit j such that $R^* \models t \simeq j$.
(iii) R^* is a $\mathcal{T}$-model of M.

Proof:

(i) From Prop. 6.4 (i) we obtain $\Omega(N_\infty) \models \bigwedge \Omega(C)$. Due to Prop. 6.4 (iii) the limit N_∞ is $\mathcal{C}$-saturated. So by Prop. 6.5 the clause sets $\Omega(N_\infty)$ and $\widehat{N_\infty}$ are equivalent. Because of Prop. 6.6, the set $\widehat{N_\infty}$ is $\mathcal{G}$-saturated. Finally R^* is a model of $\widehat{N_\infty}$.

(ii) If t is a digit itself, then we are done. Otherwise $t \equiv f(\vec{t})$; and inductively $R^* \models \bigwedge_k t_k \simeq i_k$ for some digit vector $\vec{\imath}$. Because of $\mathcal{T}' \subseteq N_1$ there is a clause $\bigvee_j f(\vec{x}) \simeq j$ in N_1, which has an Ω-instance under the substitution $\rho = \{\vec{x} \mapsto \vec{\imath}\}$. This Ω-instance $f(\vec{\imath}) \simeq 1 \vee \ldots \vee f(\vec{\imath}) \simeq n$ holds in R^* by Prop. 6.20 (i); and necessarily R^* satisfies one disjunct.

(iii) Firstly we show that R^* satisfies $\forall x. \bigvee_i x \simeq i$. Consider an R^*-assignment μ such that $\mu(x) = [t]_R$. Then t is a ground term by construction of R^*, such that $[t]_R = [j]_R$ by Prop. 6.20 (ii). Secondly, given $C \in M \subseteq N_1$, we get $R^* \models \bigwedge \Omega(C)$ from Prop. 6.20 (i). Now, C and $\bigwedge \Omega(C)$ are $\mathcal{T}$-equivalent, and R^* is a $\mathcal{T}$-model.

□

Next we will demonstrate that standard ordered rewriting is sufficient to extract the $\mathcal{T}$-model R^*. By Prop. 6.20 (ii) the carrier of R^* is given by $\{[1]_R, \ldots, [n]_R\}$, where some of the classes may coincide. Since the digits from $[1; n]$ are the smallest ground terms, every non-digit ground term $f(\vec{t})$ is R-reducible.

In order to rephrase this reducibility in terms of N_∞, let $E_\infty \subseteq N_\infty$ denote the set of all persistent unit equations, and E_k the corresponding subset of every N_k. For an arbitrary set E of such equations, the ordered rewrite relation $\to_E$ is the smallest relation on terms such that $u[s\sigma] \to_E u[t\sigma]$ whenever $s \simeq t \in E$, $s\sigma \succ t\sigma$ and $(\mathrm{var}(t) \setminus \mathrm{var}(s))\sigma \equiv \{1\}$. The third condition ensures that given say $f(x) \simeq f(y)$, the term $f(n)$ can only be rewritten to $f(1)$, thus eliminating the need to search decreasing y-instances. This restricted version of ordered rewriting with respect to E_∞ reduces every ground term to its R-normal form:

Proposition 6.21 On ground terms, E_∞-normal forms are unique and coincide with R-normal forms, and $\to_R \subseteq \to_E$ holds.

Proof: Let $E = E_\infty$. To start with, we prove $\to_R \subseteq \to_E$: Assume t is R-reducible say with $l \to r$ generated from $u \simeq v \in E$ instantiated via some ground numbering ρ. If every variable of v occurs in u, then we have directly $t \equiv t[l] \equiv t[u\rho] \to_E t[v\rho] \equiv t[r]$. Otherwise we still have to show that $x\rho \equiv 1$ for any x exclusive to v. Imagine $x\rho \succ 1$, and let ρ' coincide with ρ except that $x\rho' \equiv 1$. Since $l \to r$ has been generated, we know that $u\rho$ is $R_{(u \simeq v)\rho}$-irreducible. Because of $v\rho \succ v\rho'$, the term $u\rho$ is $R_{(u \simeq v)\rho'}$-irreducible as well. But then $u\rho \to v\rho'$ would have been generated and not $u\rho \to v\rho$.

Consider now a ground rewrite step with $u \simeq v \in E$ instantiated via σ. Let ρ denote the substitution that maps every x to $x\sigma{\downarrow_R}$. Then $R^* \models u\rho \simeq v\rho$ holds because R^* is a model of $\Omega(E) \subseteq \widehat{N_\infty}$. Because of $u\rho \downarrow_R u\sigma$ we obtain $R^* \models u\sigma \simeq v\sigma$. So we have $\to_E \subseteq \downarrow_R$ on ground terms.

This extends to ${}_E\!\leftarrow \circ \to_E \subseteq \downarrow_R \circ \downarrow_R \subseteq \leftrightarrow_R^* \subseteq \downarrow_R \subseteq \downarrow_E$, by the Church-Rosser property of R. Since the relation $\to_E$ is terminating by construction, it is also ground confluent, such that ground normal forms are unique.

Finally, E-irreducibility implies R-irreducibility because of $\to_R \subseteq \to_E$; and the converse holds because of $\to_E \subseteq \downarrow_R$. □

6.3.5 Termination

The calculus $\mathcal{C}$ is refutationally complete. If M is $\mathcal{T}$-unsatisfiable, then in every path of any fair derivation eventually the empty clause will show up, even for infinite M. Since the derivation tree is finitely branching, any such derivation is finite. If M is $\mathcal{T}$-satisfiable, however, then derivations without suitable simplification steps may become infinite. In order to overcome this, we will characterize a family of derivations which are guaranteed to terminate. In order to make this effective, naturally the input clause set M must be finite, and so is the signature Σ; and the ordering $\succ$ must be decidable.

We have seen in the preceding section that, unless the empty clause has been derived, every function application to digits, i.e., every $f(\vec{\imath})$, is reducible with respect to the limit E_∞. The key observation now is that E_∞ can sufficiently be approximated finitely: Only finitely many of the persistent equations can actually reduce the terms $f(\vec{\imath})$; and these are all present from some E_κ on. Given a non-ground function occurrence, each of its finitely many Ω-instances can then be simplified into a digit, which simplifies the non-ground expression provided some ordering restriction is met. In the end, non-digit function symbols only occur on top-level.

Formally, given a clause set N with unit equations $E \subseteq N$, we say that N *reduces to digits* if every term $f(\vec{\imath})$ is E-reducible without considering equations $x \simeq y$ or $x \simeq k$ on top-level. Inductively every ground term can then be rewritten to a digit as well. Furthermore a clause is called $[1;n]$-*shallow* if non-digit function symbols occur only at the top-level of positive literals.

Proposition 6.22 Consider a complete path $N_1, N_2, \ldots$ in a fair derivation from $M \cup \mathcal{T}'$, where M is finite.

(i) For some index κ, all $N_{\kappa+i}$ contain $\bot$; or they all reduce to digits.

(ii) If $C \in N_{\kappa+i}$ is not $[1;n]$-shallow, then C can effectively be simplified into a finite set of $[1;n]$-shallow clauses.

Proof:

(i) If M is $\mathcal{T}$-unsatisfiable, then $\bot$ is continuously present from some N_κ on, by Lem. 6.7. Otherwise we consider the rewrite system R that the superposition model functor of Sect. 6.2 produces from $\widehat{N_\infty}$. Proposition 6.20 (ii) guarantees that for every term $f(\vec{\imath})$ there is a term t such that $f(\vec{\imath}) \to_R t$ holds. Because of Prop. 6.21 the above rewrite step is

identically possible with respect to E_∞. Since the signature is finite and $\to_{E_\infty}$ is terminating, only a finite portion E of E_∞ is needed for reducing all the $f(\vec{\imath})$. For every element $s \simeq t$ of E, there is an index $k_{s\simeq t}$ such that $s \simeq t \in E_i$ for every $i \geq k_{s\simeq t}$. By finiteness of E, the maximum κ of all $k_{s\simeq t}$ is finite.

(ii) Let $E = E_{\kappa+i}$. If N_κ contains $\bot$, which is $[1;n]$-shallow itself, then any other clause is redundant. Otherwise we first show that every ground clause D which is not $[1;n]$-shallow can be simplified via ordered unit rewriting as in Prop. 6.17: By definition there is a presentation $D \equiv D[f(\vec{s})]$ such that $f(\vec{s})$ occurs (a) within a negative literal or (b) under some function symbol g. Since $N_{\kappa+i}$ reduces to digits, there is a reduction $f(\vec{s}) \to_E t$ without using equations $x \simeq y$ or $x \simeq j$ on top-level, such that Ω-admissibility is given. Furthermore $f(\vec{s}) \succ t$ holds by construction of $\to_E$. Additionally the clause to be rewritten is above the equation instance used for simplification: In case (a) we have $D[f(\vec{s})] \succ f(\vec{s}) \simeq f(\vec{s}) \succ f(\vec{s}) \simeq t$, and in case (b) there is an estimate $D[f(\vec{s})] \equiv D[g(u_1, \ldots, u_{k-1}, f(\vec{s}), u_{k+1}, \ldots, u_m)] \succeq g(u_1, \ldots, u_{k-1}, f(\vec{s}), u_{k+1}, \ldots, u_m) \simeq 1 \succ f(\vec{s}) \simeq t$.
As an inductive consequence, every ground clause which is not $[1;n]$-shallow can be simplified into a $[1;n]$-shallow clause via a sequence of unit rewriting steps. Let now Γ denote the finite set of all substitutions ρ that ground number C and satisfy $\operatorname{dom}\rho = \operatorname{var}(C)$. Then for every instance $C\rho$ there is a $[1;n]$-shallow clause D_ρ into which $C\rho$ can be rewritten. Summing it up, we obtain an instance rewriting step $\mathcal{R}\,\dfrac{C}{\{D_\rho\colon \rho \in \Gamma\}}\,N_{\kappa+i}$ in accordance with Cor. 6.19.

□

One may want to test explicitly whether a given N_k reduces to digits already (and if so, perhaps test immediately whether E_k describes a $\mathcal{T}$-model of M). Notably the property is not always inherited from N_k to N_{k+1}. Consider for example the following simplification steps in the sense of the calculus $\mathcal{C}$:

$$\mathcal{R}\,\frac{f(3) \simeq f(1)}{1 \simeq f(1)}\,1 \not\simeq 1 \vee f(3) \simeq 1 \qquad\qquad \mathcal{R}\,\frac{f(3) \simeq 1 \quad f(1) \simeq 3}{f(2) \simeq 1 \quad f(1) \simeq 2}$$

The term $f(3)$ is E_k-reducible, but not necessarily E_{k+1}-reducible. As the second example shows, this may even occur if unit equations are simplified with respect to E_k only. In case this is not desired, one has to restrict the simplification of unit equations. For example, ordered unit rewriting, instance rewriting, subsumption and tautology elimination are compatible.

Now we come to the second ingredient of our argumentation towards

termination, which will allow us to establish a bound on the number of variables in clauses.

Proposition 6.23 Inference and split conclusions do not have more variables than one of the premises:

(i) If $\sigma = \mathrm{mgu}(u, v)$ with $\operatorname{ran}\sigma \subseteq \mathscr{V} \cup [1; n]$ and $\operatorname{dom}\sigma \cup \operatorname{cdom}\sigma \subseteq \operatorname{var}(u, v)$, then there is a variant σ' that additionally satisfies $\operatorname{var}(v\sigma') \subseteq \operatorname{var}(v)$.

(ii) If $C \vdash D$ is a unary inference or a split, then $\operatorname{var}(D) \subseteq \operatorname{var}(C)$ holds.

(iii) If $l \simeq r,\ C \vdash D$ is a binary inference, then $|\operatorname{var}(D)| \leq |\operatorname{var}(C)|$ is true.

Proof:

(i) Let $\mathcal{P}(m)$ hold iff there exists an mgu σ of u and v with $\operatorname{ran}\sigma \subseteq \mathscr{V} \cup [1; n]$, $\operatorname{dom}\sigma \cup \operatorname{cdom}\sigma \subseteq \operatorname{var}(u, v)$, and $|\operatorname{var}(v\sigma) \setminus \operatorname{var}(v)| = m$. By assumption $\mathcal{P}$ holds for some $m \geq 0$. We will now show that $\mathcal{P}(j+1)$ implies $\mathcal{P}(j)$.
Assume σ is a witness for $\mathcal{P}(j+1)$. Because of $j + 1 > 0$ there exists a variable y in $\operatorname{var}(v\sigma) \setminus \operatorname{var}(v)$. By the shape of σ, this variable is the σ-image of another variable $x \in var(v)$. Consider now the substitutions $\tau = \{x \mapsto y, y \mapsto x\}$ and $\sigma' = \sigma \circ \tau$. The latter is a unifier of u and v. Because of $\sigma'\tau = \sigma\tau^2 = \sigma$, it is even a most general one. The image of a variable z under σ' is x if $z\sigma \equiv y$, and $z\sigma$ otherwise; in particular $x\sigma' \equiv x$ and $y\sigma' \equiv x$. That is, going from σ to σ', the variable x moves from the dom-part to the cdom-part, and y in the opposite direction, which are all effects in terms of dom and cdom. The identity $\operatorname{var}(v\sigma') = (\operatorname{var}(v\sigma) \cup \{x\}) \setminus \{y\}$ concludes the proof of $\mathcal{P}(j)$.

(ii) In case of an equality resolution step $C \vee t \simeq t' \vdash C\sigma$ we have $\operatorname{cdom}\sigma \subseteq \operatorname{var}(t, t')$. Given a split $C \vee s \simeq t \vee l \simeq r \vee D \vdash (C \vee s \simeq t)\tau \mid (l \simeq r \vee D)\tau$, the substitution τ is numbering, such that $\operatorname{cdom}\tau \subseteq [1; n]$.

(iii) We will prove that $\operatorname{var}(D) \subseteq \operatorname{var}(C)$ holds in case the most general unifier is chosen according to Prop. 6.23 (i). All mgu's are equal up to variable renaming; and the number of variables in a clause is invariant under such renamings. This yields the estimate stated above.
We jointly treat superposition left and right inferences via the pattern $l \simeq r,\ C[l'] \vdash C[r]\sigma\tau \equiv D$. Because of $l'\sigma\tau \equiv l\sigma\tau \succ r\sigma\tau$ we know that $\operatorname{var}(l'\sigma\tau) \supseteq \operatorname{var}(r\sigma\tau)$ is true, and hence $\operatorname{var}(D) \subseteq \operatorname{var}(C\sigma\tau) \subseteq \operatorname{var}(C\sigma)$. Applying Prop. 6.23 (i), without loss of generality σ can be chosen such that $\operatorname{var}(l'\sigma) \subseteq \operatorname{var}(l')$. Let σ' denote the restriction of σ to $\operatorname{var}(l')$. By this definition we have $\operatorname{cdom}\sigma' \subseteq \operatorname{var}(l'\sigma) \subseteq \operatorname{var}(l') \subseteq \operatorname{var}(C)$. Since the premises are variable disjoint, we obtain $\operatorname{var}(C\sigma) = \operatorname{var}(C\sigma') \subseteq \operatorname{var}(C) \cup \operatorname{cdom}\sigma' = \operatorname{var}(C)$, which completes the proof of $\operatorname{var}(D) \subseteq \operatorname{var}(C)$.

□

Simplifying a $[1;n]$-shallow clause with respect to other such clauses can arbitrarily increase the number of variables and need not preserve $[1;n]$-shallowness, as witnessed by

$$\mathcal{R}\,\frac{f(x) \simeq 2}{g(1) \not\simeq g(1) \vee y_1 \not\simeq y_1 \vee \ldots \vee y_m \not\simeq y_m \vee f(x) \simeq 1}\, 2 \simeq 1$$

if $f(1) \succ g(1)$. Clearly this counteracts our efforts towards termination; so a strategy is needed that guides the execution of calculus steps. We say that a $\mathcal{C}$-derivation is a $\mathcal{C}_\kappa$-*derivation* from a clause set M if (i) it is fair, (ii) the root node is $M \cup \mathcal{T}'$, and in every path eventually the following conditions hold: (iii) simplifications do not increase the number of variables, (iv) $[1;n]$-shallowness is preserved under simplifications, (v) inferences and splits are not repeated, (vi) every fresh inference conclusion which is not $[1;n]$-shallow is immediately simplified into a set of $[1;n]$-shallow clauses, (vii) no duplicate literals occur in $[1;n]$-shallow clauses, and (viii) $[1;n]$-shallow clauses equal up to variable renaming are identified. Indeed such derivations exist for every finite M: The crucial item (vi) can be satisfied because of Prop. 6.22. Condition (v) does not conflict with (i) because repeated inferences and splits are redundant.

Theorem 6.24 $\mathcal{C}_\kappa$-derivations decide $\mathcal{T}$-satisfiability of finite clause sets.

Proof: Consider a $\mathcal{C}_\kappa$-derivation from a finite clause set M. Then M by Lem. 6.7 is $\mathcal{T}$-satisfiable if and only if the derivation contains a complete path without the empty clause in the limit. The derivation tree is finitely branching. It remains to show that every path $N_1, N_2, \ldots$ is finite. Let $\|N_i\| = \max\{|\mathrm{var}(C)| : C \in N_i\}$.

There exists an index κ such that from N_κ on, the conditions (iii) through (vi) of the definition of $\mathcal{C}_\kappa$-derivation are satisfied. We form a subsequence of $N_1, N_2, \ldots$ that starts from $N'_1 = N_\kappa$. If in N_i a new clause C is inferred and, according to condition (vi), immediately simplified into $[1;n]$-shallow clauses $\vec{D}$ until N_{i+k}, then for $(N'_j)_j$ all sequence elements but N_{i+k} are dropped, and the latter shows up only if $\vec{D}$ is not empty. Assume now $(N_i)_i$ is infinite. Inferences with empty $\vec{D}$ are not repeated because of condition (v), as well as splits; so there must be infinitely many simplifications or inferences with non-empty $\vec{D}$. Since simplifications are decreasing with respect to Ω-instances, the latter occur infinitely many times; so $(N'_j)_j$ is then infinite as well.

Inductively $\|N'_j\| \leq \|N'_1\|$ holds for all j: If a clause $C \in N'_j$ is simplified to some non-empty $\vec{D}$, then we know that $|\mathrm{var}(D_i)| \leq |\mathrm{var}(C)|$ by condition

(iii). In case of a split or an inference, we additionally apply Prop. 6.23 (ii) and Prop. 6.23 (iii).

Assume now $(N'_j)_j$ were infinite; then we can argue like above for $(N_i)_i$ and obtain that infinitely many inferences are drawn. The inference conclusions are simplified according to condition (vi), such that they become $[1; n]$-shallow and have no more than $\|N'_1\|$ variables. Because of conditions (vii) and (viii), only finitely many such clauses exist. Moreover the number of clauses that are produced from simplification and splitting alone is finite. Therefore, eventually an inference has to be repeated, but this contradicts condition (v). Hence $(N'_j)_j$ is finite, and so is $(N_i)_i$. □

6.3.6 Extensions

Let us have a short look at a many-sorted setting where $\mathcal{T}$ consists of size restrictions for every sort, each built over an individual set of digits. One has to employ the usual typing constraints for equations, terms and substitutions. Then the calculus $\mathcal{C}$ and the results obtained for it so far straightforwardly extend to this situation.

Up to now, our calculus did not deal with predicates. Of course one could extend $\mathcal{C}$ with an ordered resolution rule, and consider predicate atoms in the superposition and split rules. Alternatively, we can introduce a two-element sort Bool, say over the digits I and II, and provide a clause $\mathrm{I} \not\simeq \mathrm{II}$. As usually we can now encode predicate atoms $P(\vec{t})$ of any other sort as equations $P(\vec{t}) \simeq \mathrm{I}$. Notably $\mathcal{T}'$ need not contain an axiom $P(\vec{x}) \simeq \mathrm{I} \vee P(\vec{x}) \simeq \mathrm{II}$: Given an algebra $\mathcal{A}$ such that at some point $P^{\mathcal{A}}$ does not map into $\{\mathrm{I}^{\mathcal{A}}, \mathrm{II}^{\mathcal{A}}\}$, let the algebra $\mathcal{B}$ coincide with $\mathcal{A}$ except that $P^{\mathcal{B}}$ maps all such points onto $\mathrm{II}^{\mathcal{B}}$. Then $\mathcal{A}$ and $\mathcal{B}$ satisfy the same encoded atoms $P(\vec{t}) \simeq \mathrm{I}$.

As an application, consider the validity problem for a formula $\phi \equiv \forall x_1 \ldots \forall x_n \exists y_1 \ldots \exists y_m \phi'$ where ϕ' is quantifier-free and contains no function symbols. This problem was proven decidable by Bernays and Schönfinkel [BS28]. Now, ϕ is valid iff $\psi \equiv \forall y_1 \ldots \forall y_m \neg\phi'\{x_1 \mapsto 1, \ldots, x_n \mapsto n\}$ is unsatisfiable iff ψ is $\mathcal{T}$-unsatisfiable. Since no function symbols are present, the set $\mathcal{T}'$ is empty. That is, derivations start from unaugmented clause sets.

Corollary 6.25 $\mathcal{C}_\kappa$-derivations decide the Bernays-Schönfinkel class.

Notably no instance rewriting steps are needed in such derivations because all literals are shallow.

6.4 Combinations with First-Order Theories

So far, we have only considered the case where actually the overall Herbrand domain of a formula is finite. The interesting question is whether the techniques developed in the previous sections can be generalized to a setting where the overall Herbrand domain may be infinite, but finite subsets of the domain are specified. The answer we give in the section is affirmative; the combination can exploit the advanced technology: In every inference, variables over any finite subset only need to be instantiated to variables and to the finitely many domain representatives. Furthermore no inferences with the cardinality-bounding axiom are needed.

The overall approach is to code bounded subsets via monadic predicates, which we also call *sorts* [Wei01].[1] Provided a clause C contains a negative literal $\neg S(x)$, we say that x *is of sort* S *in* C. To give an example, any model $\mathcal{A}$ of the clauses $S(1)$, $S(2)$, $\neg S(x) \vee x \simeq 1 \vee x \simeq 2$ must satisfy $1 \leq |S^{\mathcal{A}}| \leq 2$. If we add the clause $1 \not\simeq 2$, then any model $\mathcal{A}$ fulfills $|S^{\mathcal{A}}| = 2$, whereas the alternative extension with $1 \simeq 2$ leads to $|S^{\mathcal{A}}| = 1$. Concerning functions, the clause $\neg S(x) \vee S(f(x))$ declares f to map elements from S into S.

So please recall that in contrast to many-sorted or order-sorted logics and reasoning, in our context the semantics of a sort is the semantics of its monadic predicate. Therefore, sorts may be empty, there are no restrictions to the language, sorts are not a priori disjoint, elements of sorts are not necessarily different and sorts may of course be also defined via general clauses. For example, the clause $\neg R(x, f(x)) \vee S(x)$ defines x to be contained in the sort S if the relation $R(x, f(x))$ holds, and the clause $\neg S(x) \vee \neg T(x)$ states that the sorts S and T are disjoint.

In this section, we study the bounded-domain theory $\mathcal{T}$ for a sort S of cardinality up to n defined by

$$S(1), \quad S(2), \quad \ldots, \quad S(n), \quad \neg S(x) \vee x \simeq 1 \vee \ldots \vee x \simeq n$$

which is a clausal presentation of the formula

$$\forall x.\, S(x) \leftrightarrow x \simeq 1 \vee \ldots \vee x \simeq n$$

Similar to the restricted case of Sect. 6.3, we would like to instantiate variables of sort S with digits only. All such instances of the clause $\neg S(x) \vee x \simeq 1 \vee \ldots \vee x \simeq n \in \mathcal{T}$ are tautologies. To compensate for this, we introduce an operator $\_^{\circ}$ to be applied on input clauses that replaces every positive literal $S(t)$ by the disjunction $t \simeq 1 \vee \ldots \vee t \simeq n$. Furthermore let in this

[1] This denomination collides with sorts in the many-sorted framework as in Def. 2.2 (i). Formally, sorts in the sense of this section reside within sorts in the sense of Chap. 2.

section $\mathcal{T}'$ stand for the theory $\{S(1), \ldots, S(n)\}$. Finally $\Omega_S(C)$ shall denote the set of all clauses obtained from C via instantiation of all variables of sort S in C with digits. In this sense Ω_S is the restriction of Ω to variables of sort S. The following lemma is the analogue of Prop. 6.1:

Lemma 6.26 A clause set N is $\mathcal{T}$-satisfiable iff $\Omega_S(N^\circ)$ is $\mathcal{T}'$-satisfiable.

Proof: First of all, N and N° are $\mathcal{T}$-equivalent by definition of $\_^\circ$ and $\mathcal{T}$. Secondly we show that for N° $\mathcal{T}$-satisfiability is the same as $\mathcal{T}'$-satisfibility. In the positive case this follows from $\mathcal{T} \supseteq \mathcal{T}'$. Assume now that N° is $\mathcal{T}$-unsatisfiable. Then from $N^\circ \cup \mathcal{T}$ one can derive $\bot$ via standard superposition, in particular if in the clause $\neg S(x) \vee x \simeq 1 \vee \ldots \vee x \simeq n$ the literal $\neg S(x)$ is selected. Hence this clause contributes to the derivation only via resolution steps through $\neg S(x)$. But in N° there is no positive literal $S(t)$, and the resolvents with the literals $S(i) \in \mathcal{T}'$ are tautologies. Therefore we can derive $\bot$ from $N^\circ \cup \mathcal{T}'$ alone, such that N° is $\mathcal{T}'$-unsatisfiable.

Thirdly, if N° is $\mathcal{T}'$-satisfiable, so is $\Omega_S(N^\circ)$. For the converse, consider a $\mathcal{T}'$-model $\mathcal{A}$ of $\Omega_S(N^\circ)$, and let $\mathcal{B}$ coincide with $\mathcal{A}$ except that $S^\mathcal{B}$ holds on $1^\mathcal{B}, \ldots, n^\mathcal{B}$ only. Hence $\mathcal{B}$ is a $\mathcal{T}'$-model and fulfills $S^\mathcal{B} \subseteq S^\mathcal{A}$. Therefore $\mathcal{B}$ with any assignment satisfies all the negative literals $\neg S(t)$ that $\mathcal{A}$ does. Since N° is free of positive literals $S(t)$, we know that $\mathcal{B}$ is a model of N°. □

Note that the four clauses $S(1)$, $S(2)$, $\neg S(x) \vee x \simeq 1 \vee x \simeq 2$, $f^3(x) \not\simeq f(x)$ are satisfiable as neither the input, nor the output of f is of sort S. Adding the declaration $\neg S(x) \vee S(f(x))$ lets f map from S into S and hence causes unsatisfiability. For the latter clause, the transformation of Lem. 6.26 applies.

Now by the lifting theorem for standard superposition, we know because of Lem. 6.26 that $N \cup \mathcal{T}$ has a superposition refutation iff $N^\circ \cup \mathcal{T}'$ has one. The open question is how we can exploit the fact that we considered solely numbering substitutions for variables of sort S. Note that although S has a finite domain, the overall domain of N may be infinite. Therefore, we cannot take the approach of Sect. 6.3 where we used the numbering substitution available for all variables to require that inferences are only performed on strictly greatest terms and literals. Furthermore, the abstract superposition redundancy notion is no longer effective and satisfiability is of course not decidable anymore. Therefore, the idea is to restrict the range of substitutions for variables of sort S to $\mathcal{V} \cup [1; n]$, and to require that (strict) maximality is preserved under any numbering substitution for the bounded sort S. We formulate the following refinement of the standard superposition calculus for bounded sorts:

Superposition left

$$\mathcal{I}\,\frac{C \vee l \simeq r \quad s[l'] \not\simeq t \vee D}{(C \vee s[r] \simeq t \vee D)\sigma}$$

if
- $l' \notin \mathscr{V}$ and $\sigma = \mathrm{mgu}(l, l')$
- $\operatorname{ran}\sigma|_S \subseteq \mathscr{V} \cup [1; n]$
- there exists a minimally numbering τ of sort S such that $s \not\simeq t$ is maximal under $\sigma\tau$ and l, $l \simeq r$, s are strictly so

Superposition right

$$\mathcal{I}\,\frac{C \vee l \simeq r \quad s[l'] \simeq t \vee D}{(C \vee s[r] \simeq t \vee D)\sigma}$$

if
- $l' \notin \mathscr{V}$ and $\sigma = \mathrm{mgu}(l, l')$
- $\operatorname{ran}\sigma|_S \subseteq \mathscr{V} \cup [1; n]$
- there exists a minimally numbering τ of sort S such that l, $l \simeq r$, s, $s \simeq t$ are strictly maximal under $\sigma\tau$ and $(C \vee l \simeq r)\sigma\tau \not\succeq (s \simeq t \vee D)\sigma\tau$

Equality resolution

$$\mathcal{I}\,\frac{C \vee t \not\simeq t'}{C\sigma}$$

if
- $\sigma = \mathrm{mgu}(t, t')$
- $\operatorname{ran}\sigma|_S \subseteq \mathscr{V} \cup [1; n]$
- there exists a minimally numbering τ of sort S such that $t \not\simeq t'$ is maximal under $\sigma\tau$

Equality factoring

$$\mathcal{I}\,\frac{C \vee s \simeq t \vee s' \simeq u}{C \vee t \not\simeq u \vee s \simeq u}$$

if
- $\sigma = \mathrm{mgu}(s, s')$
- $\operatorname{ran}\sigma|_S \subseteq \mathscr{V} \cup [1; n]$
- there exists a minimally numbering τ of sort S such that $s \simeq t$ is maximal under $\sigma\tau$ and s is strictly so

For the general combination of a bounded sort with arbitrary formulae over potentially infinite domains, we cannot aim at a decision procedure, but only at refutational completeness. Therefore, the above rule delays instantiations of bounded-sort variables as long as possible, but applies the underlying restrictions. This is different from the inference rules presented in Sect. 6.3 where variables are instantiated by digits in order to meet the ordering restrictions of superposition.

We stipulate that the digits $1, \ldots, n$ are minimal in the ordering $\succ$. Furthermore, in every clause $\neg S(t) \vee C$ with a negative sort literal, the argument of S shall always be a digit or a variable. If t is neither of these, then one can apply variable abstraction and obtains $\neg S(x) \vee x \not\simeq t \vee C$. Notably the

new variable x needs to be instantiated with digits only, and hence cannot become maximal. Our considerations give rise to the following completeness result:

Theorem 6.27 A clause set N is $\mathcal{T}$-unsatisfiable iff there is a derivation of the empty clause from $N^\circ \cup \mathcal{T}'$ by the superposition calculus defined above.

Proof: By Lem. 6.26 the sets $N \cup \mathcal{T}$ and $\Omega_S(N^\circ) \cup \mathcal{T}'$ are equisatisfiable. The latter is unsatisfiable iff there is a derivation of the empty clause by the standard superposition calculus. As the digits are minimal in the ordering, they might only be replaced by each other. For any clause $\neg S(x) \vee C \in N^\circ$, all instances of $\neg S(x)$ in the proof are generated by substitutions from x into $[1; n]$. Hence, all steps can be lifted to steps of the above refined superposition calculus on N°. □

Here is an example for the refined maximality condition. Let $\succ$ denote the lexicographic path ordering induced by the precedence $f \succ g \succ n \succ \ldots \succ 1$. Then in the clause $\neg S(x) \vee g(x, y) \simeq y \vee f(y) \simeq y$ the literal $g(x, y) \simeq y$ is not maximal because $f(y) \succ g(i, y)$ holds for every $i \in [1; n]$.

6.5 Summary

We have presented a light-weight adaptation of superposition calculi to the first-order theory of bounded domains. The achievement is a superposition calculus for bounded domains that

(i) facilitates the precise calculation of ordering restrictions,
(ii) restricts the range of inference unifiers to digits and variables,
(iii) turns the general semantic redundancy notion effective,
(iv) incorporates a particular splitting rule for non-Horn clauses,
(v) constitutes a decision procedure for any bounded-domain problem,
(vi) is mostly compatible with the redundancy notion of standard superposition,

and can in particular

(vii) be embedded via monadic predicates (sorts) in general first-order settings with potentially infinite domains.

We have already done some promising experiments on the basis of the superposition calculus for bounded domains [HTW06], and a full-fledged integration into SPASS is on the way. To this end, ordering computation, inference computation and redundancy notions will be refined for the case of variables with a bounded domain.

7 On the Proliferation of an AC Deletion Rule

The superposition calculus, which has been studied in the preceding chapters from various perspectives, combines techniques of ordered resolution on the clause level with such of Knuth-Bendix completion on the term level.

In this chapter, we will go back to completion and recall a simple, but effective deletion rule for theories with operators that are associative and commutative (AC). This rule was once developed by me for the WALDMEISTER system [LH02], which is a completion-based theorem prover that originates from the diploma theses of Arnim Buch and myself [HBF96]. Since 1997, the system has dominated, in its category, the annual CADE ATP System Competition [SS06], which so to say is the world championship for first-order automated theorem proving. In writing "A New Kind of Science", S. Wolfram employed the prover for deriving minimal axiomatizations of the Sheffer stroke [Wol02, pp. 772–821, p. 1158]. These successful experiments initiated the integration of WALDMEISTER into the MATHEMATICA computer algebra system of Wolfram Research, Inc. during the time when this thesis was carried out.

Interestingly, the aforementioned deletion rule has spread, since its publication in [Hil00, Chap. 6.1] and [AHL03], to the first-order provers E [Sch01, p. 371] and PROVER9 [McC08, Sect. *Process Inferred*], which are based on variants of superposition. However, a correctness proof is missing up to now [Sch07].

We begin this chapter with a recapitulation of ordered completion. Next, we demonstrate that ground confluence of rewrite systems can be rephrased as a property of proofs, namely that rewrite proofs of ground instances are essentially ground instances of first-order level proofs. Then we recall a ground convergent system for AC. We continue with a motivation why AC reasoning is difficult and mention some approaches to cope with it. As a simple remedy, we present our AC deletion rule, and prove its correctness in the completion

framework on the basis of the above characterization of ground confluence. Thanks to this more abstract proceeding, our proof immediately applies to any ground convergent rewrite system; and admittedly, it closes a gap that has remained unnoticed so far in all previous versions.

Moving over to the superposition framework, we demonstrate that the deletion rule is not correct with respect to the standard notion of redundancy. We show that this can be fixed with a refined literal ordering. Interestingly, the latter can be extended such that superposition redundancy subsumes completion redundancy.

7.1 Background

As said, Knuth-Bendix completion is an ancestor of superposition. While the realm of the latter is clausal logic, the former is restricted to equational logic, where a theory is a (countable) set E of universally quantified equations, not of arbitrary clauses. The *word problem for* E is to decide whether or not $E \models \forall X.\, s \simeq t$ holds. Birkhoff's theorem [Bir35] states that $E \models \forall X.\, s \simeq t$ is true if and only if s can be converted into t by repeated application of equations from E. Thus, it constitutes already a semi-decision procedure for the word problem, which is the limit by Turing completeness of equational logic.

Many theories, however, enjoy a finite presentation that facilitates a full decision procedure via rewriting: Guided by a reduction ordering, every equation is applied in one direction only. Furthermore, the Church-Rosser property is required, or equivalently confluence, and ensures that the non-determinism due to competing redexes is don't-care. In sum, normal forms exist and are unique. The word problem hence reduces to the computation and comparison of normal forms. Quite a number of theories can be transformed into such a shape: By Newman's Diamond Lemma [New42], local confluence amounts to confluence in the presence of termination. Local confluence can effectively be tested according to the Critical Pair Lemma [KB70]. In the negative case, the presentation is enriched with particular equational consequences called critical pairs. Adding simplifications and iterating the procedure, we have the essence of Knuth-Bendix completion [KB70].

This original variant of completion fails if equations are encountered that cannot be oriented. Fortunately, for the word problem confluence restricted to the level of ground terms is sufficient. Such theory presentations, though infinite in general, can always be obtained via ordered completion [Lan75], where orientable instances of non-orientable equations are taken into account as well. Now, if two ground terms reduce to the same normal form in the

limit, then also in a finite approximation thereof. We obtain another semi-decision procedure for the word problem, with a drastically reduced search space [HR87].

A proof-theoretic framework for reasoning about completion was introduced in [BDH86]. The description as rule-based calculus allows to abstract from details of a particular strategy. Completion is understood as a process that eventually transforms any arbitrary proof into a particular normal form: The transformation is embedded into a well-founded proof ordering, by construction of which minimal proofs are just proofs by rewriting. The counterpart of this proof-theoretic construction in the superposition world is the model-generation method. Our outline of completion in the next section follows the presentation in [Bac91, Chap. 4].

As an aside, E-validity both of universal and of positive formulae (cf. Def. 2.2 (v)) can be reduced to the word problem in simple ways. In the former case, the clausification is not only equisatisfiable, but even equivalent, since there is no existential Skolemization. Hence, the problem boils down to a sequence of clausal E-validity problems $E \models \forall X.\, \vec{s} \simeq \vec{t} \to \bigvee_j u_j \simeq v_j$, where $\vec{s} \simeq \vec{t}$ stands for a conjunction $\bigwedge_i s_i \simeq t_i$. By convexity of equational logic, these problems reduce to uniform word problems $E \models \forall X.\, \vec{s} \simeq \vec{t} \to u_j \simeq v_j$. Via universal Skolemization and the deduction theorem, we arrive at word problems $E, \overline{\vec{s}} \simeq \overline{\vec{t}} \models \overline{u_j} \simeq \overline{v_j}$. Concerning positive formulae, these are without loss of generality presented in prenex normal form. Then one applies universal Skolemization, which preserves E-validity. After transformation into disjunctive normal form, a sequence of problems $E \models \exists X.\, \vec{s} \simeq \vec{t}$ remains. Each of these is equivalent to $E, eq(\vec{s}, \vec{t}) \simeq F, eq(\vec{x}, \vec{x}) \simeq T \models T \simeq F$ where the symbols eq, T and F are fresh.

7.2 Ordered Completion

We mentioned in the previous section that, when dealing with the word problem, ordered rewrite systems need not be confluent, but only ground confluent. More precisely, this property must be stable under every extension Σ^e of the given signature Σ with zero or finitely many new function symbols, which arise in the universal Skolemization of conjectures. Throughout this chapter we require that a ground total reduction ordering $\succ$ is provided. Whenever a signature extension Σ^e is considered, we silently assume that $\succ$ is arbitrarily extended to a ground total reduction ordering over Σ^e, the latter always being possible [Rub95, Sect. 7]. Actually, if $\succ$ is a lexicographic path ordering, only one particular extension Σ^e must be taken into account [Nie93, Sect. 2].

In order to facilitate finer distinctions between equational proofs, we annotate theory equations with *labels* like in $s \simeq_m t$.[1] The labels are taken from some well-founded, totally ordered domain, say the natural numbers or length-bounded sequences thereof, and are disregarded in the semantics of equations. As before, $s \simeq_m t$ is identified with $t \simeq_m s$. However, $s \simeq_m t$ and $s \simeq_n t$ are distinct provided $m \neq n$. We occasionally drop labels where they do not matter. For example, an equation $s \simeq t$ is called *orientable* if $s \succ t$ or $t \succ s$ holds.

Given a set of equations E and a reduction ordering $\succ$, then the set of *orientable instances* is the rewrite system $E^{\succ} = \{u\sigma \to v\sigma\colon u \simeq_m v \in E \wedge u\sigma \succ v\sigma\}$. Instances are built with respect to the given signature Σ, but also with respect to some extension Σ^e if indicated in the context. The orientable instances generate the *ordered rewrite relation* $\longrightarrow_{E^{\succ}}$, which is terminating by definition. A *peak* is a rewrite sequence $s \;{}_{E^{\succ}}\!\longleftarrow u \longrightarrow_{E^{\succ}} t$, and is a *ground peak* if all terms are ground. In order to express unrestricted replacement of equals by equals according to E, we identify E with the rewrite system $\{u \to v,\ v \to u\colon u \simeq_m v \in E\}$.

A *proof step in* E is an expression $s \longleftrightarrow^{p}_{u \Rightarrow_m v} t$ such that $u \simeq_m v \in E$, $s|_p \equiv u\sigma$ and $t \equiv s[v\sigma]_p$ for some substitution σ. We write $s \longrightarrow^{p}_{u \Rightarrow_m v} t$ if $s|_p \succ t|_p$ holds, and $s \longleftarrow^{p}_{u \Rightarrow_m v} t$ if $s|_p \prec t|_p$ does. A *proof* P *of* $t_0 \longleftrightarrow^*_E t_n$ is a sequence

$$t_0 \longleftrightarrow^{p_1}_{u_1 \Rightarrow_{m_1} v_1} t_1 \longleftrightarrow^{p_2}_{u_2 \Rightarrow_{m_2} v_2} \ldots \longleftrightarrow^{p_n}_{u_n \Rightarrow_{m_n} v_n} t_n$$

of proof steps in E. Annotations that are not relevant may be omitted. The empty sequence serves as proof of $t_0 \longleftrightarrow^*_E t_0$. A *subproof* Q *of* P is a subsequence of P from some t_i up to some t_{i+j}, and is a proof of $t_i \longleftrightarrow^*_E t_{i+j}$. We write $P[Q]$ to indicate that P contains Q as a subproof, and (ambiguously) denote by $P[Q']$ the proof obtained from P by replacing Q by Q', where hence Q' is also a proof of $t_i \longleftrightarrow^*_E t_{i+j}$. We write P^{-1} for the proof

$$t_n \longleftrightarrow^{p_n}_{v_n \Rightarrow_{m_n} u_n} t_{n-1} \longleftrightarrow^{p_{n-1}}_{v_{n-1} \Rightarrow_{m_{n-1}} u_{n-1}} \ldots \longleftrightarrow^{p_1}_{v_1 \Rightarrow_{m_1} u_1} t_0$$

of $t_n \longleftrightarrow^*_E t_0$. Furthermore, $P\sigma$ denotes the proof

$$t_0\sigma \longleftrightarrow^{p_1}_{u_1 \Rightarrow_{m_1} v_1} t_1\sigma \longleftrightarrow^{p_2}_{u_2 \Rightarrow_{m_2} v_2} \ldots \longleftrightarrow^{p_n}_{u_n \Rightarrow_{m_n} v_n} t_n\sigma$$

[1] In order to give an example for these finder distinctions, we shall see that if $\succ$ is a lexicographic path ordering or a Knuth-Bendix ordering, then the following system is ground confluent:

$$\begin{array}{ll} (x+y)+z \simeq_m x+(y+z) & x+y \simeq_o y+x \\ (x+y)+z \simeq_n y+(x+z) & x+(y+z) \simeq_p y+(x+z) \end{array}$$

This property is preserved, as well as equivalence, if one drops one of the equations in the left column. However, it will turn out that deleting say the first one can be justified as simplification only if labels are taken into account and m is greater than n.

of $t_0\sigma \longleftrightarrow_E^* t_n\sigma$. If s is a term with position q, then $s[P]_q$ stands for the proof

$$s[t_0]_q \longleftrightarrow^{qp_1}_{u_1 \Rightarrow_{m_1} v_1} s[t_1]_q \longleftrightarrow^{qp_2}_{u_2 \Rightarrow_{m_2} v_2} \ldots \longleftrightarrow^{qp_n}_{u_n \Rightarrow_{m_n} v_n} s[t_n]_q$$

of $s[t_0]_q \longleftrightarrow_E^* s[t_n]_q$. Given a proof P' of $t_n \longleftrightarrow_E^* s$, the *composition* of P and P' is denoted by juxtaposition PP', and is a proof of $t_0 \longleftrightarrow_E^* s$. Finally, a *rewrite proof of* $t_0 \longleftrightarrow_E^* t_n$ is any proof of the shape

$$t_0 \longrightarrow^{p_1}_{u_1 \Rightarrow_{m_1} v_1} \ldots \longrightarrow^{p_i}_{u_i \Rightarrow_{m_i} v_i} t_i \longleftarrow^{p_{i+1}}_{u_{i+1} \Rightarrow_{m_{i+1}} v_{i+1}} \ldots \longleftarrow^{p_n}_{u_n \Rightarrow_{m_n} v_n} t_n$$

We assign a *complexity* to individual proof steps according to

$$s \longleftrightarrow^{p}_{u \Rightarrow_m v} t \;\longmapsto\; \begin{cases} (\{s\}, & u,\ m, t\ ) & \text{if } s \succ t \\ (\{t\}, & v,\ m, s\ ) & \text{if } t \succ s \\ (\{s,t\}, & \bot, \bot, \bot) & \text{otherwise} \end{cases}$$

where in every domain, the symbol $\bot$ shall stand for some fixed element. Complexities are compared in the lexicographic combination of the following orderings: (i) the multiset extension of the reduction ordering $\succ$, (ii) the encompassment ordering $\gtrsim_{\text{enc}}$, (iii) the ordering on labels, (iv) the reduction ordering $\succ$.[2] Going further, the *complexity of a proof* is the multiset of the complexities of its individual proof steps. The multiset extension of the ordering on proof step complexities is an ordering on proof complexities. We define a relation $\circledcirc$ on proofs via $P \circledcirc Q$ if the complexity of P is greater than the one of Q. It is a well-founded ordering on proofs, and compatible with substitutions, the term structure and the proof structure: $Q \circledcirc Q'$ implies $Q\sigma \circledcirc Q'\sigma$ as well as $u[Q] \circledcirc u[Q']$ and $P[Q] \circledcirc P[Q']$, provided $P[Q]$ and $P[Q']$ are proofs. A relation with these properties is dubbed a *proof ordering.* By construction, every proof is $\circledcirc$-greater than each of its strict subproofs. When comparing proofs, we write the equation $s \simeq_m t$ as shorthand for the one-step proof $s \longleftrightarrow^{\lambda}_{s \Rightarrow_m t} t$, which is unambiguous because $t \longleftrightarrow^{\lambda}_{t \Rightarrow_m s} s$ has the same complexity. Additionally, in this context $(s \simeq_m t)\sigma$ stands for $s\sigma \longleftrightarrow^{\lambda}_{s \Rightarrow_m t} t\sigma$.

[2]Alternatively, the redex, which is $s|_p$, $t|_p$ or $\bot$, may serve as another component between (i) and (ii), and be compared in the subterm ordering $\geq_{\text{subt}}$. Then the subsumption ordering $\gtrsim_{\text{subs}}$ is sufficient for component (ii). This variant is preceding [BDH86] and still shows up in [Bac87] and [Bac91, Chap. 2], whereas the above proof ordering appears in [BDP89] and [Bac91, Chap. 4]. With both versions, ordered completion specializes into unfailing completion. Furthermore, our AC deletion rule is correct with respect to both orderings. But when comparing individual proofs, opposite results can be obtained, as for $f(f(c)) \longrightarrow^{\lambda}_{f(x) \Rightarrow_m c} c$ and $f(f(c)) \longrightarrow^{1}_{f(x) \Rightarrow_m c} f(c) \longrightarrow^{\lambda}_{f(x) \Rightarrow_m c} c$: In the variant with redex, the left proof is greater, in the variant without, the right one is.

We continue to use the notions of inference, reduction, derivation, limit etc. of Sect. 2.5. Doing so, the *ordered completion* calculus consists of the following two rules:

Deduction

$$\mathcal{I}\,\frac{E}{s \simeq_m t} \qquad \text{if } \cdot\ s \longleftrightarrow^*_E t$$

Deletion

$$\mathcal{R}\,\frac{s \simeq_m t}{}\,E \quad \text{if} \quad \begin{array}{l} \cdot \text{ for every } \Sigma^e \text{ and ground substitution } \sigma \\ \text{there is a proof } P \text{ of } s\sigma \longleftrightarrow^*_E t\sigma \\ \text{such that } (s \simeq_m t)\sigma \circledcirc P \end{array}$$

Note that in the *Deduction* rule, the set E of premises may be assumed finite, hence meeting the requirements on inference rules of Sect. 2.5. Furthermore, the *Deletion* rule has no effect in case $s \simeq_m t \in E$.

To this calculus, we can add simplification as a derived rule, capturing that an equation is deduced which triggers a subsequent deletion of another equation. In order to exclude infinite simplification chains, the deduced equation should be smaller than the deleted one with respect to some fixed well-founded ordering. One formulation is this:

Simplification

$$\mathcal{R}\,\frac{s \simeq_m t}{u \simeq_n v}\,E \quad \text{if} \quad \begin{array}{l} \cdot\ u \longleftrightarrow^*_{E \cup \{s \simeq_m t\}} v \\ \cdot \text{ there is a proof } P \text{ of } s \longleftrightarrow^*_{E \cup \{u \simeq_n v\}} t \\ \quad \text{such that } s \simeq_m t \circledcirc P \\ \cdot\ s \simeq_m t \circledcirc u \simeq_n v \end{array}$$

Since there is no splitting rule, the notion of derivation of Sect. 2.5 specializes into a sequence $E_1, E_2, \ldots$ of sets of equations. Here, we will call such a derivation *fair* if for every Σ^e and ground peak $s \;{}_{E^{\succ}_{\infty}}\!\longleftarrow u \longrightarrow_{E^{\succ}_{\infty}} t$ there exists a $\circledcirc$-smaller proof of $s \longleftrightarrow^*_{\bigcup_i E_i} t$. Next we show that the limit of a fair derivation is ground confluent, and equivalent to the original theory.

Theorem 7.1 Consider a fair derivation $E_1, E_2, \ldots$ by ordered completion.

(i) E_i and E_{i+1} are equivalent.

(ii) If $s \simeq_m t \in E_i \setminus E_{i+1}$, then for every Σ^e and ground substitution σ there is a proof of $s\sigma \longleftrightarrow^*_{E_\infty} t\sigma$ which is $\circledcirc$-smaller than $(s \simeq_m t)\sigma$.

(iii) E_1 and E_∞ are equivalent.

(iv) For every Σ^e and ground peak $s \;{}_{E^{\succ}_{\infty}}\!\longleftarrow u \longrightarrow_{E^{\succ}_{\infty}} t$, there exists a $\circledcirc$-smaller proof of $s \longleftrightarrow^*_{E_\infty} t$.

(v) $\longrightarrow_{E^{\succ}_{\infty}}$ is ground confluent in every Σ^e.

Proof:

(i) The transition from E_i to E_{i+1} is either by a deduction step or by a deletion step. In the former case, the new equation is a consequence of E_i. In the latter case, if $s \simeq_m t$ is deleted, consider a signature extension Σ^e by enough constants to Skolemize $s \simeq_m t$, and let σ denote such a Skolemizing substitution. Then $s\sigma \longleftrightarrow^*_{E_{i+1}} t\sigma$ holds, and therefore $E_{i+1} \models \forall X.\, s \simeq t$ as well.

(ii) By definition of *Deletion,* there exists a proof P of $s\sigma \longleftrightarrow^*_{E_{i+1}} t\sigma$ such that $(s \simeq_m t)\sigma \circledcirc P$. Stripping off contexts, this proof is by steps $e_1\sigma_1, \ldots, e_k\sigma_k$ where each e_j stands for an equation $u_j \simeq_{n_j} v_j \in E_{i+1}$, and $(s \simeq_m t)\sigma \circledcirc e_j\sigma_j$ holds by construction of the proof ordering. For every non-persistent equation e_j there exists inductively a proof Q_j of $u_j\sigma_j \longleftrightarrow^*_{E_\infty} v_j\sigma_j$ which is $\circledcirc$-smaller than $e_j\sigma_j$. Next we replace all such proof steps $w[e_j\sigma_j]_p$ in P by $w[Q_j]_p$, and finally obtain a proof P' of $s\sigma \longleftrightarrow^*_{E_\infty} t\sigma$ that moreover satisfies $P' \underline{\circledcirc} P \circledcirc (s \simeq_m t)\sigma$.

(iii) Every persistent equation shows up in some E_j and hence is E_1-valid by (i). Every equation of E_1 which is not persistent has after Skolemization a proof in E_∞, according to (ii), and hence is E_∞-valid, also without Skolemization.

(iv) By fairness there exists a proof P of $s \longleftrightarrow^*_{\bigcup_i E_i} t$ which is $\circledcirc$-smaller than the peak. Thanks to (ii), this proof can non-increasingly be transformed into a proof of $s \longleftrightarrow^*_{E_\infty} t$.

(v) For every ground peak $s \;{}_{E^\succ_\infty}\!\longleftarrow u \longrightarrow_{E^\succ_\infty} t$, there is a smaller proof according to (iv). By well-foundedness of $\circledcirc$, there exists a minimal proof of $s \longleftrightarrow^*_{E_\infty} t$, which has no peak. By ground totality of $\succ$, this proof follows the pattern $s \longrightarrow^*_{E^\succ_\infty} \circ\, {}_{E^\succ_\infty}\!{}^*\longleftarrow t$, or $s \downarrow_{E^\succ_\infty} t$.

□

For actually constructing fair derivations, we introduce a notion which captures the essence of local divergencies: In the following instance of the deduction rule, the conclusion is called a *critical pair* between the premises.

Critical pairing

$$\mathcal{I}\ \frac{s \simeq_m t \quad u[s'] \simeq_n v}{(u[t] \simeq_o v)\sigma}\ \text{if}\ \begin{array}{l}\cdot\ s' \notin \mathcal{V} \text{ and } \sigma = \mathrm{mgu}(s, s') \\ \cdot\ s,\ u \text{ and } u \simeq_n v \text{ are strictly maximal under} \\ \quad \sigma\tau \text{ for one } \Sigma^e \text{ and ground substitution } \tau\end{array}$$

The second side condition is decidable for example if $\succ$ is a lexicographic path ordering [Com90], but not in the general case, and hence often over-approximated as maximality under σ. Disregarding labels, a finite set of equations gives rise to only finitely many critical pairs. Interestingly, crit-

ical pairing literally corresponds to the inference rule *Superposition right* restricted to equational logic.

Lemma 7.2 Critical pairs defuse peaks, and ensure fairness:

(i) Consider a ground peak $s \; {}_{E^\succ}\!\longleftarrow u \longrightarrow_{E^\succ} t$ in some Σ^e.
Then $s \downarrow_{E^\succ} t$ holds, or $s \longleftrightarrow_{\{e\}^\succ} t$ by some critical pair e of E.
This is known as *Ordered Critical Pair Lemma* [Lan75].

(ii) Assume $E_1, E_2, \ldots$ is a derivation such that every critical pair, up to different labels, between persistent equations appears in some E_i. Then the derivation is fair.

Proof:

(i) More concretely, let the ground peak be given by

$$s \equiv u[w\sigma]_p \longleftarrow^p_{w \Rightarrow v} u \longrightarrow^q_{v' \Rightarrow w'} u[w'\sigma']_q \equiv t$$

If none of the positions is a prefix of the other, then the two redexes within $u \equiv u[v\sigma]_p[v'\sigma']_q$ do not interfere. Because of $v'\sigma \succ w'\sigma$ and $v\sigma \succ w\sigma$, there is a rewrite proof

$$s \equiv u[w\sigma]_p[v'\sigma']_q \longrightarrow_{\{v' \simeq w'\}^\succ} u[w\sigma]_p[w'\sigma']_{q\ \{v \simeq w\}^\succ}\!\longleftarrow u[v\sigma]_p[w'\sigma']_q \equiv t$$

Otherwise, we assume without loss of generality that p is a prefix of q, say because $q = pp'$. In other words, q is below p, and $u \equiv u[v\sigma]_p[v'\sigma']_{pp'} \equiv u[v\sigma[v'\sigma']_{p'}]_p$.

(a) p' might have a presentation $q'q''$ such that $v|_{q'}$ is some variable x. Hence we have $u \equiv u[v\sigma[x\sigma[v'\sigma']_{q''}]_{q'}]_p$. Let now τ denote the substitution identical to σ except that $x\tau \equiv x\sigma[w'\sigma]_{q''}$. So we obtain $x\sigma \longrightarrow_{\{v' \simeq w'\}^\succ} x\tau$. If l is an arbitrary term where x occurs at positions $p_1, \ldots, p_n$, then this extends to

$$l\sigma \equiv l\sigma[x\sigma]_{p_1} \ldots [x\sigma]_{p_n} \longrightarrow^*_{\{v' \simeq w'\}^\succ} l\sigma[x\tau]_{p_1} \ldots [x\tau]_{p_n} \equiv l\tau$$

Applying this to $t \equiv u[v\sigma[x\sigma[w'\sigma']_{q''}]_{q'}]_p$, there is a rewrite sequence

$$t \equiv u[v\sigma[x\tau]_{q'}]_p \longrightarrow^*_{\{v' \simeq w'\}^\succ} u[v\tau[x\tau]_{q'}]_p \equiv u[v\tau]_p$$

Correspondingly, the term $s \equiv u[w\sigma]_p$ can be rewritten into $u[w\tau]_p$. Finally, by ground totality of $\succ$, the terms $v\tau$ and $w\tau$, unless already equal, can be joined by one rewrite step with $\longrightarrow_{\{v \simeq w\}^\succ}$.

(b) Otherwise p' is a position within v such that $v|_{p'}$ is not a variable. Without loss of generality, the equations $v \simeq w$ and $v' \simeq w'$ share no variables. Let $\rho = \sigma \cup \sigma'$. This gives $u \equiv u[v\rho[v'\rho]_{p'}]_p \equiv u[v[v']_{p'}\rho]_p$. In particular, the terms $v|_{p'}$ and v' are unifiable, say with most general unifier τ. By definition of ground peak, the terms v and

v' are strictly maximal under ρ in $v \simeq w$ and $v' \simeq w'$, respectively. Besides, these two equations are not identical under ρ because $s \not\equiv t$. Summing it up, there is a critical pair $v[w']_{p'}\tau \simeq w\tau$. Finally, by ground totality of $\succ$, the terms $s \equiv u[w\rho]_p$ and $t \equiv u[v[w']_{p'}\rho]_p$, unless already equal, can be joined by one rewrite step with $\longrightarrow_{\{(v[w']\simeq w)\tau\}^{\succ}}$.

(ii) Given a ground peak $s \;{}_{E^{\succ}_{\infty}}\!\longleftarrow\; u \longrightarrow_{E^{\succ}_{\infty}} t$ in some Σ^e, because of Lem. 7.2 (i) one of the following applies:

(a) $s \downarrow_{E^{\succ}_{\infty}} t$: Then there is a rewrite proof of $s \longleftrightarrow^*_{E\infty} t$ and therefore also of $s \longleftrightarrow^*_{\bigcup_i E_i} t$. This rewrite proof is smaller than the peak.

(b) $s \longleftrightarrow_{\{e\}^{\succ}} t$ by some critical pair e of E_∞: Then e is already a critical pair of some E_i, and by assumption contained in some E_j. Hence, in E_j there exists a proof smaller than the peak.

□

The requirements of Lem. 7.2 (ii) can be weakened as follows, the proof easily being adapted: For every critical pair between persistent equations, it is actually sufficient that for every ground instance in some Σ^e, there exists a proof in some E_i which is smaller than the associated peak. This gives rise to *critical pair criteria* like connectedness [WB86] or compositeness [KMN85].

Next we apply ordered completion to the word problem. If $^-$ indicates Skolemization and $\succ$ is ground total on the extension of Σ with Skolem symbols, then the following statements are equivalent for arbitrary fair derivations $E, E_1, E_2, \ldots$:

(i) $E \models \forall X.\, s \simeq t$
(ii) $E \models \overline{s} \simeq \overline{t}$
(iii) $\overline{s} \downarrow_{E^{\succ}_{\infty}} \overline{t}$
(iv) $\overline{s} \downarrow_{E^{\succ}_{i}} \overline{t}$ for some i
(v) $\overline{s} \downarrow_{E^{\succ}_{i}} \overline{t}$ for almost every i

Testing for $\downarrow_{E^{\succ}_{i}}$ involves search. A simpler method is to repeatedly replace $\overline{s}$ and $\overline{t}$ by some current $\longrightarrow_{E^{\succ}_{i}}$-normal form thereof, because by well-foundedness eventually a $\longrightarrow_{E^{\succ}_{\infty}}$-normal form is reached. A subtle search problem remains: Given an equation with extra variables like in $f(x) \simeq g(y)$ and a term matched by the left-hand side say under $\{x \mapsto s\}$, then one must look for an instance t of y such that $f(s) \succ g(t)$ holds. Clearly, it is sufficient to check the smallest ground term, which is the minimal constant in the extension of Σ with Skolem symbols.

An alternative more successful in practice is to always restart the normalization from s and t. For reducing ground terms to normal forms, we can deal with extra variables as just described. So if E_1 is finite, then by König's Lemma every ground term has only finitely many normal forms in E_1. By definition of *Deletion,* if a ground term is reducible with respect to

$\longrightarrow_{E_i^\succ}$, then it is also with respect to $\longrightarrow_{E_{i+1}^\succ}$. Conversely, if S_i denotes the set of $\longrightarrow_{E_i^\succ}$-normal forms of s, then $S_1 \supseteq S_2 \supseteq \ldots$ holds. By fairness and finiteness, almost all sets S_j contain only one element, namely $s{\downarrow_{E_\infty^\succ}}$.

Unfailing completion [BDP89] is a constructive instance of ordered completion. Emphasis is put on equations $s \simeq_m t$ where $s \succ t$, since these can be applied, if at all, in one direction only. An equation for which this has been detected is called a rule; and it is understood that all labels of rules are smaller than labels of equations. There is an explicit calculus rule for orienting an equation into a rule, which is a simplification step in the sense of ordered completion. The deduction rule is instantiated into critical pairing, overapproximating the maximality conditions on the left-hand sides and on the main premise. Fairness is via Lem. 7.2 (ii). The calculus comes with a number of concrete simplifications that allow to rewrite one equation or rule with another. Applied exhaustively, they guarantee that the limit of the derivation is even confluent, respectively confluent and finite, in case the initial theory has, under the given reduction ordering, any presentation with these properties. Rewriting the left-hand side of a rule, for example, is as follows:

Collapse of rule

$$\mathcal{R}\,\frac{l[s\sigma] \simeq_m r}{l[t\sigma] \simeq_o r}\; s \simeq_n t \quad \text{if} \quad \begin{array}{l} \cdot\ l \succ r,\ s\sigma \succ t\sigma \text{ and } l >_{\text{enc}} s \\ \cdot\ l[s\sigma] \simeq_m r \text{ is a rule, } l[t\sigma] \simeq_o r \text{ is not} \end{array}$$

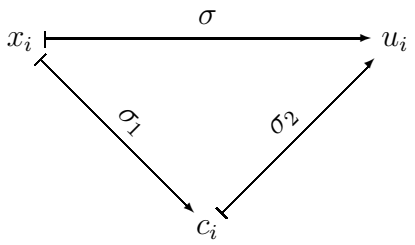
7.3 Proofs in Ground Confluent Systems

The *Deletion* rule requires that in order to get rid of an equation $s \simeq_m t$, there must be a smaller proof of $s\sigma \longleftrightarrow_E^* t\sigma$ for every ground substitution σ in every signature extension Σ^e. If E is confluent, then s and t are joinable, hence $s \longleftrightarrow_E^* t$ has a rewrite proof. If E is only ground confluent, then there is just a rewrite proof of $s\sigma \longleftrightarrow_E^* t\sigma$ for every σ as above. In an arbitrary such proof, rewriting may take place within terms introduced by σ, or at the fringe of this substitution part and the skeleton of s or t. Here we will show that such rewrite steps can be avoided. In other words, for any such σ there is a proof P of $s \longleftrightarrow_E^* t$ such that $P\sigma$ is a rewrite proof of $s\sigma \longleftrightarrow_E^* t\sigma$. In the next section, we will use this particular proof as witness in a deletion rule based on a ground confluent subsystem, because $P\sigma$ is easier to compare with $(s \simeq_m t)\sigma$ than an arbitrary rewrite proof.

For this section, we fix a set E of equations over a given signature Σ, an arbitrary signature extension Σ^e, a number of variables $x_1, \ldots, x_k$ not

showing up in E, and a ground substitution $\sigma = \{\vec{x} \mapsto \vec{u}\}$ over the extended signature. By $\mathscr{T}(\Sigma)[\vec{x}] = \{t \in \mathscr{T}(\Sigma)\colon \mathrm{var}(t) \subseteq \{\vec{x}\}\}$ we denote the subset of terms built over no other variables than $x_1, \ldots, x_k$. The idea for the construction of a proof P with only skeleton steps is to consider an alternative signature extension Σ^c by fresh constants c_i which will be used in place of the substitutes u_i. For convenience we assume that these constants are employed as variables in Σ^e, such that $\mathscr{T}(\Sigma^e)$ becomes a superset of $\mathscr{T}(\Sigma^c)$.

We want to decompose σ with an intermediate step via Σ^c, as indicated on the right, and therefore consider substitutions $\sigma_1 = \{\vec{x} \mapsto \vec{c}\}$ and $\sigma_2 = \{\vec{c} \mapsto \vec{u}\}$, the latter in Σ^e. There, σ_1 is just a variable renaming, hence has an inverse $\sigma_1^{-1} = \{\vec{c} \mapsto \vec{x}\}$ such that $\sigma_2 = (\sigma_1^{-1}\sigma_1)\sigma_2 = \sigma_1^{-1}\sigma$ is true. The identity $t\sigma_1\sigma_2 \equiv t\sigma$ holds on terms over Σ, as proven inductively by:



$$t\sigma_1\sigma_2 \equiv \left\{\begin{array}{ll} t \equiv x_i\colon & x_i\sigma_1\sigma_2 \equiv c_i\sigma_2 \equiv u_i \equiv x_i\sigma \\ t \equiv y\colon & y\sigma_1\sigma_2 \equiv y\sigma_2 \equiv y \equiv y\sigma \\ t \equiv f(\vec{t})\colon & f(\vec{t})\sigma_1\sigma_2 \equiv f(\vec{t}\sigma_1\sigma_2) \equiv f(\vec{t}\sigma) \end{array}\right\} \equiv t\sigma$$

On terms over the signature Σ^c, we need an extension of the reduction ordering such that the constants c_i essentially play the rôle of the substitutes u_i in $\mathscr{T}(\Sigma^e)$. To this end, let $s \gtrsim_1 t$ hold whenever $s\sigma_2 \succeq t\sigma_2$ does, fix an arbitrary ground total reduction ordering $>_2$ on $\mathscr{T}(\Sigma^c)$, and let $\gtrsim$ denote the lexicographic combination of $\gtrsim_1$ and $\geq_2$.

Proposition 7.3 The relation $>$ is a ground total reduction ordering and extends $\succ$ from $\mathscr{T}(\Sigma)$ to $\mathscr{T}(\Sigma^c)$.

Proof: In order to reduce the number of indices, let $\tau = \sigma_2$ in this proof. First of all, we will establish that $\gtrsim_1$ is a terminating quasi-ordering: It is transitive because $s \gtrsim_1 t \gtrsim_1 u$ implies $s\tau \succeq t\tau \succeq u\tau$, hence $s\tau \succeq u\tau$ and $s \gtrsim_1 u$. Reflexivity holds because of $s\tau \succeq s\tau$. From $s >_1 t$ we may conclude $s\tau \succ t\tau$ because $s \gtrsim_1 t \not\gtrsim_1 s$ implies $s\tau \succeq_1 t\tau \not\geq_1 s\tau$. Therefore, any infinite descending chain in $>_1$ would induce another one in $\succ$, such that $\gtrsim_1$ must be terminating. Because of Prop. 2.18, the relation $\gtrsim$ now is a terminating quasi-ordering with equivalence part $\equiv$, which means that $\gtrsim$ is a partial ordering. In the following, we just speak of its strict part $> = >_1 \cup (\approx_1 \cap >_2)$.

Next we deal with context stability. Without loss of generality, we restrict to a context $f(\_, \vec{v})$. The ordering $>_1$ inherits the property from $\succ$: $s >_1 t$ spells out as $s\tau \succ t\tau$, hence $f(s, \vec{v})\tau \equiv f(s\tau, \vec{v}\tau) \succ f(t\tau, \vec{v}\tau) \equiv f(t, \vec{v})\tau$ and

$f(s, \vec{v}) >_1 f(t, \vec{v})$. Similarly, context stability carries over from $\equiv$ to $\approx_1$; and it applies to the reduction ordering $>_2$ by assumption. Since now all components of $>$ are stable under contexts, so is $>$ itself.

For proving stability under substitutions, we proceed in the same fashion. Consider an arbitrary substitution ρ over Σ^c. Stability does not directly carry over from $\succ$ to $>_1$ because $\succ$ deals with terms over the signature Σ^e. To overcome this problem, let ρ^τ denote the substitution that maps every $x \in \operatorname{dom}\rho$ to $x\rho\tau$, where the constants c_i are replaced by the substitutes u_i. Notably, the range of ρ^τ is within $\mathscr{T}(\Sigma^e)$. The domains of ρ and ρ^τ coincide because σ is a ground substitution, such that $x\rho\tau \not\equiv x$. The substitution ρ and the mapping τ can be permuted in the sense that the identity $t\rho\tau \equiv t\tau\rho^\tau$ holds on terms over Σ^e, which can be proven inductively as follows:

$$t\rho\tau \equiv \left\{\begin{array}{ll} t \equiv x \in \operatorname{dom}\rho: & x\rho\tau \equiv x\tau\rho\tau \overset{(1)}{\equiv} x\tau\rho^\tau \\ t \equiv y \notin \operatorname{dom}\rho: & y\rho\tau \overset{(1)}{\equiv} y\rho^\tau \equiv y\tau\rho^\tau \\ t \equiv c_i: & c_i\rho\tau \equiv c_i\tau \equiv u_i \overset{(2)}{\equiv} u_i\rho^\tau \equiv c_i\tau\rho^\tau \\ t \equiv f(\vec{t}): & f(\vec{t})\rho\tau \equiv f(\vec{t}\rho\tau) \overset{(3)}{\equiv} f(\vec{t}\tau\rho^\tau) \equiv f(\vec{t})\tau\rho^\tau \end{array}\right\} \equiv t\tau\rho^\tau$$

The missing arguments are that (1) there is no difference between $s\rho\tau$ and $s\rho^\tau$ if s contains no constant c_i, (2) u_i is ground and (3) the induction hypothesis applies. We apply this permutation property as follows: Assume $s >_1 t$ holds in $\mathscr{T}(\Sigma^c)$; then we have $s\tau \succ t\tau$ in $\mathscr{T}(\Sigma^e)$ and $s\tau\rho^\tau \succ t\tau\rho^\tau$ because $\succ$ is stable under substitutions with range in $\mathscr{T}(\Sigma^e)$. This can just be rewritten into $s\rho\tau \succ t\rho\tau$, which is nothing but $s\rho >_1 t\rho$. In the same way, we get from $s \approx_1 t$ to $s\tau \equiv t\tau$, then to $s\tau\rho^\tau \equiv t\tau\rho^\tau$, next to $s\rho\tau \equiv t\rho\tau$, and finally to $s\rho \approx_1 t\rho$. Summing it up, all components of $>$ are stable under substitutions, and so is $>$, which by now is proven a reduction ordering.

Going further, since $\succ$ is ground total, for all ground terms s and t over Σ^c we either have $s\tau \succeq t\tau$ or $s\tau \preceq t\tau$, hence $s \gtrsim_1 t$ or $s \lesssim_1 t$. Assume the former, which implies $s >_1 t$ or $s\ (\approx_1 \cap \geq_2)\ t$, therefore $s > t$ or $s \equiv t$.

Finally, assume that $s \succ t$ holds for two terms over Σ. Terms without constants c_i are fix under τ, such that the above relation can be rewritten into $s\tau \succ t\tau$, which implies $s >_1 t$ and $s > t$. Hence, the ordering $>$ is an extension of $\succ$. □

Rewrites of σ_1-instances induce rewrites of σ-instances:

Proposition 7.4 If $s\sigma_1 \longrightarrow^p_{l\Rightarrow r} t$ is a proof step in E with $s \in \mathscr{T}(\Sigma)[\vec{x}]$ and $t \in \mathscr{T}(\Sigma^c)$, then $s\sigma \longrightarrow^p_{l\Rightarrow r} t\sigma_1^{-1}\sigma$ is a proof step in E unless $s\sigma \equiv t\sigma_1^{-1}\sigma$.

Proof: The given rewrite step can be written as replacement

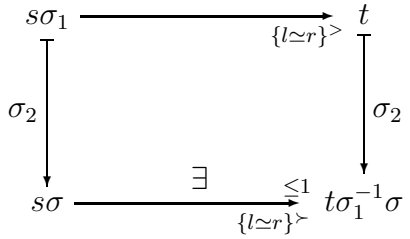
$$s\sigma_1 \equiv s\sigma_1[l\theta]_p \longleftrightarrow_{\{l\simeq r\}} s\sigma_1[r\theta]_p \equiv t \qquad (*)$$

where θ is a ground substitution over Σ^c with $\operatorname{dom}\theta \subseteq \operatorname{var}(l \simeq r)$ and $l\theta > r\theta$. Since $s\sigma_1$ and t are ground, we even have $\operatorname{dom}\theta = \operatorname{var}(l \simeq r)$. Because of $\mathscr{T}(\Sigma^c) \subseteq \mathscr{T}(\Sigma^e)$ we can read $(*)$ also in the signature Σ^e, where σ_2 is a substitution and $\longleftrightarrow_{\{l\simeq r\}}$ is closed under σ_2. We just obtain

$$s\sigma_1\sigma_2 \equiv s\sigma_1\sigma_2[l\theta\sigma_2]_p \longleftrightarrow_{\{l\simeq r\}} s\sigma_1\sigma_2[r\theta\sigma_2]_p \equiv t\sigma_2$$

From $l\theta > r\theta$ we conclude that $l\theta \gtrsim_1 r\theta$ and $l\theta\sigma_2 \succeq r\theta\sigma_2$ are true. The above identities $\sigma_1\sigma_2 = \sigma$ on Σ and $\sigma_2 = \sigma_1^{-1}\sigma$ on Σ^e yield $s\sigma \longrightarrow^{\leq 1}_{\{l\simeq r\}^\succ} t\sigma_1^{-1}\sigma$. $\square$

In the preceding proposition, we have shown that the diagram to the right commutes. The extension to rewrite sequences is straightforward:



Proposition 7.5 For every proof $s\sigma_1 \equiv s_0 \longrightarrow s_1 \longrightarrow \ldots \longrightarrow s_m$ of $s\sigma_1 \longleftrightarrow_E^* s_m$ in Σ^c where $s \in \mathscr{T}(\Sigma)[\vec{x}]$, there is a subsequence $\vec{\imath}$ of $(j)_{j=1}^m$ such that the following is a proof of $s\sigma \longleftrightarrow_E^* s_m\sigma_1^{-1}\sigma$ in Σ^e:

$$s\sigma \equiv s_0\sigma_1^{-1}\sigma \longrightarrow s_{i_1}\sigma_1^{-1}\sigma \longrightarrow \ldots \longrightarrow s_{i_n}\sigma_1^{-1}\sigma \equiv s_m\sigma_1^{-1}\sigma$$

Furthermore, every $s_i\sigma_1^{-1}$ is in $\mathscr{T}(\Sigma)[\vec{x}]$.

Proof: If $m = 0$, then $\vec{\imath}$ is empty. Otherwise, we are given a proof $s_0 \longrightarrow \ldots \longrightarrow s_m \longrightarrow s_{m+1}$. Since $s\sigma_1$ is ground and all steps are decreasing, every s_i must be ground. Inductively exist $\vec{\imath}$ and a proof $s_0\sigma_1^{-1}\sigma \longrightarrow s_{i_1}\sigma_1^{-1}\sigma \longrightarrow \ldots \longrightarrow s_{i_n}\sigma_1^{-1}\sigma \equiv s_m\sigma_1^{-1}\sigma$. If s_m and s_{m+1} coincide under $\sigma_1^{-1}\sigma$, then we are already done. Otherwise, since s_m is a ground term in Σ^c, we know that $s_m\sigma_1^{-1}$ belongs to $\mathscr{T}(\Sigma)[\vec{x}]$. Hence, we can apply Prop. 7.4 to the given proof step $(s_m\sigma_1^{-1})\sigma_1 \longrightarrow s_{m+1}$, and obtain a proof step $(s_m\sigma_1^{-1})\sigma \longrightarrow s_{m+1}\sigma_1^{-1}\sigma$. $\square$

With two applications of Prop. 7.5, the picture becomes complete:

Proposition 7.6 Assume E is ground confluent in Σ^c. If $s \simeq t$ is E-valid and $s, t \in \mathscr{T}(\Sigma)[\vec{x}]$ holds, then there is a proof P of $s \longleftrightarrow_E^* t$ such that $P\sigma$ is a rewrite proof.

Proof: By ground confluence, there is a term $v \in \mathscr{T}(\Sigma^c)$ such that $s\sigma_1 \longrightarrow_{E^>}^* v \;{}_{E^>}^*\!\longleftarrow t\sigma_1$ is true. By Prop. 7.5, the following proofs exist in Σ^e such that

every s_i and t_j is in $\mathscr{T}(\Sigma)$:

$$\begin{array}{ll} P: & s\sigma \longrightarrow s_1\sigma \longrightarrow \ldots \longrightarrow s_m\ \sigma \equiv v\sigma_1^{-1}\sigma \\ Q: & t\sigma \longrightarrow t_1\sigma \longrightarrow \ldots \longrightarrow t_n\ \sigma \equiv v\sigma_1^{-1}\sigma \end{array}$$

Factoring out the substitution σ, we obtain two proofs in Σ:

$$\begin{array}{ll} P': & s \longleftrightarrow s_1 \longleftrightarrow \ldots \longleftrightarrow s_m\ \equiv v\sigma_1^{-1} \\ Q': & t \longleftrightarrow t_1 \longleftrightarrow \ldots \longleftrightarrow t_n\ \equiv v\sigma_1^{-1} \end{array}$$

Finally, the proof $P'(Q')^{-1}$ has the desired properties. □

We have just concluded from ground confluence in Σ^c to a structural property on proofs in Σ with ground instances in Σ^e. Notably, from the latter we get to ground confluence in Σ^e: For every ground instance $(s \simeq t)\sigma$ of every critical pair, since critical pairs are E-valid, the property implies $s\sigma \downarrow_{E^\succ} t\sigma$, and thus ground confluence by Lem. 7.2 (i). Moving on from the fixed signature extension Σ^e to arbitrary ones, we obtain:

Lemma 7.7 The following are equivalent for every set E of equations:
(i) E is ground confluent in every Σ^e.
(ii) For every E-valid Σ-equation $s \simeq t$ and ground substitution σ in every Σ^e, there is a proof P of $s \longleftrightarrow_E^* t$ such that $P\sigma$ is a rewrite proof.

Another straightforward consequence of Prop. 7.6 is that ground confluence in arbitrary signature extensions reduces to ground confluence in extensions by constants.

Lemma 7.8 The following are equivalent for every set E of equations:
(i) E is ground confluent in every Σ^e.
(ii) E is ground confluent in every Σ^e that extends Σ only by constants.

7.4 A Ground Confluent System for AC

Assumption 7.9 For the rest of this chapter, we fix the following data:
(i) $\mathscr{F}_{\mathrm{AC}} \subseteq \mathscr{F}$ is a set of binary function symbols.
(ii) AC consists of the following equations for every operator $+ \in \mathscr{F}_{\mathrm{AC}}$:

$$(x+y)+z \simeq x+(y+z) \qquad x+y \simeq y+x \qquad x+(y+z) \simeq y+(x+z)$$

(iii) $\succ$ is a reduction ordering satisfying the following for all ground terms in every Σ^e, provided $+ \in \Sigma_{\mathrm{AC}}$ and $s \succ t$:

$$(t+u)+v \succ t+(u+v) \qquad s+t \succ t+s \qquad s+(t+u) \succ t+(s+u)$$

The elements of $\mathscr{F}_{\mathrm{AC}}$ are called *AC operators* and usually written in infix notation. The equations within the set AC are called named *associativity, commutativity* and *extended commutativity*, respectively; the first and second imply the third. Reduction orderings as required in (iii) indeed exist: For example, every Knuth-Bendix ordering is adequate (Def. 2.25), or every recursive path ordering with status (Def. 2.24) where AC operators have lexicographic status.

The rewrite relation $\longrightarrow_{\mathrm{AC}\succ}$ can be used for sorting sums:

Proposition 7.10 For all terms $s_1 \preceq s_2 \preceq \ldots \preceq s_n$ and every AC operator $+$ and permutation π, the following holds:

$$s_{\pi 1} + (\ldots + (s_{\pi(n-1)} + s_{\pi n}) \ldots) \longrightarrow^*_{\mathrm{AC}\succ} s_1 + (\ldots + (s_{n-1} + s_n) \ldots)$$

Proof: If $n = 1$, then no rewrite step is necessary. Otherwise, we conclude from n to $n+1$. Dropping parentheses, we assume here that all sums are implicitly parenthesized to the right. We consider $s \equiv s_{\pi 1} + \ldots + s_{\pi(n+1)}$ where $\pi j = 1$ for some index j. The summands s_i are not necessarily distinct. But without loss of generality, j is the smallest position at which the term s_1 occurs: If s_k were a duplicate of s_1 at a position $k < j$, then k could be assumed as minimal, and we would employ the permutation $\pi \circ (j\ k)$ in place of π. Now, if j equals 1, then by induction hypothesis there is a rewrite sequence

$$s \equiv s_1 + s_{\pi 2} + \ldots + s_{\pi(n+1)} \longrightarrow^*_{\mathrm{AC}\succ} s_1 + s_2 + \ldots + s_{n+1}$$

Otherwise, s_1 must be permuted to the front before, according to

$$\begin{aligned} s \quad &\equiv \quad s_{\pi 1} + \ldots + s_{\pi(j-2)} + s_{\pi(j-1)} + s_1 + s_{\pi(j+1)} + \ldots + s_{\pi(n+1)} \\ &\longrightarrow_{\mathrm{AC}\succ} s_{\pi 1} + \ldots + s_{\pi(j-2)} + s_1 + s_{\pi(j-1)} + s_{\pi(j+1)} + \ldots + s_{\pi(n+1)} \\ &\longrightarrow^*_{\mathrm{AC}\succ} s_1 + s_{\pi 1} + \ldots + s_{\pi(j-1)} + s_{\pi(j+1)} + \ldots + s_{\pi(n+1)} \end{aligned}$$

All rewrite steps are by extended commutativity, except the first one in case $j = n$. The steps are decreasing because by construction, all terms $s_{\pi 1}, \ldots, s_{\pi(j-1)}$ are greater than s_1. □

A sufficient criterion for ground confluence was presented in [MN90]: In the analysis of critical pairs, one can perform case splits according to the different ordering relations that are possible between variable instances; and joinability in all cases implies ground confluence. An implementation of this method confirms that $\longrightarrow_{\mathrm{AC}\succ}$ is ground confluent. Here, we give a compact hand-made proof of this property, working with patterns of critical pairs and exploiting the above sorting property.

Proposition 7.11 $\longrightarrow_{\mathrm{AC}\succ}$ is ground confluent in every Σ^e.

Proof: We apply Lem. 7.2 (i) and show that for every critical pair, each of its ground instances in some Σ^e can be joined. In every pair, only one AC operator occurs. Each of the axioms associativity, commutativity and extended commutativity of Ass. 7.9 shows up as side premise, but only in left-to-right direction, either by symmetry of the equation or by the reduction ordering. Abstracting over the concrete main premises, we obtain the following schema:

$$(1)\frac{(x+y)+z \simeq x+(y+z) \quad e[x'+t]}{e[x+(y+t)]\{x' \mapsto x+y\}} \quad (2)\frac{(x+y)+z \simeq x+(y+z) \quad e[(s+t)+u]}{e[s+(t+u)]}$$

$$(3)\frac{x+y \simeq y+x \quad e[s+t]}{e[t+s]}$$

$$(4)\frac{x+(y+z) \simeq y+(x+z) \quad e[t+x']}{e[y+(t+z)]\{x' \mapsto y+z\}} \quad (5)\frac{x+(y+z) \simeq y+(x+z) \quad e[s+(t+u)]}{e[t+(s+u)]}$$

Every main premise e is built exclusively over the operator $+$ and up to three variables, each of which occurs exactly once on each side. The inference conclusion in (2), (3) and (5) are AC-variants of their respective main premise and hence fall under this description as well. Concerning the conclusions in (1) and (4), the number of variables increases to four, besides which the description still fits. With some applications of associativity and renaming of variables, every critical pair can be rewritten into an identity

$$x_1 + (\ldots + (x_{n-1} + x_n)\ldots) \simeq x_{\pi 1} + (\ldots + (x_{\pi(n-1)} + x_{\pi n})\ldots)$$

Consider now a signature extension Σ^e and a ground substitution $\sigma = \{\vec{x} \mapsto \vec{t}\}$. Since $\succ$ is total in Σ^e, there exists a permutation ρ such that $t_{\rho 1} \preceq t_{\rho 2} \preceq \ldots \preceq t_{\rho n}$ holds. Given the instance of the above equation under σ, by Prop. 7.10 both sides can be rewritten into $t_{\rho 1} + (\ldots + (t_{\rho(n-1)} + t_{\rho n})\ldots)$. □

7.5 A Deletion Rule for AC Theories

Theories with AC operators are difficult to reason with for unfailing completion procedures: The commutativity axiom cannot be oriented. With the Knuth-Bendix ordering and the lexicographic path ordering as defined in Sect. 2.4, which are the orderings most frequently used, associativity is oriented such that sums are eventually parenthesized to the right. Consider now a rule that contains a sum in the left-hand side, like $s[u_1 + (u_2 + u_3)] \simeq t$ where each u_i is headed by a symbol different from $+$. Now, of all permutative variants $s[u_{\pi 1} + (u_{\pi 2} + u_{\pi 3})] \simeq t$, unfailing completion is doomed to generate all those with $\succ$-minimal permuted sums.

Even worse, feed unfailing completion with the AC axioms $(x+y)+z \simeq x+(y+z)$ and $x+y \simeq y+z$ alone. Most completion implementations

feature subsumption deletion, which allows to discard $u[s\sigma] \simeq u[t\sigma]$ if $s \simeq t$ is present. Notwithstanding this simplification, one can witness an infinite band of permutative equations over variables

$$x_1 + (\ldots + (x_{n-1} + x_n)\ldots) \;\simeq\; x_{\pi 1} + (\ldots + (x_{\pi(n-1)} + x_{\pi n})\ldots)$$

where π ranges over all permutations such that (i) thanks to subsumption, π moves 1, and (ii) for each π and π^{-1}, there is only one equation, because it is applied in both directions.

The number of band elements per n develops as 1, 3, 11, 53, 313, ... More precisely, let $I(n)$ denote the number of involution permutations[3] of order n. For every n there exist $n! - (n-1)!$ permutations that move 1. Of these, $I(n-1)$ are involutions and correspond to one equation. In the remainder, two permutations give rise to one equation. The number of equations for each n sums up to

$$\frac{1}{2}\left(I(n-1) + (n-1)(n-1)!\right)$$

An elegant, but intricate way to overcome this nuisance is to perform identity tests, rewriting and completion modulo the AC congruence, thereby considering only one representative per class. This has led to a rich strand of research, starting with completion modulo AC [LB77, PS81]. The approach has been extended to ordered paramodulation [RV91, Pau92] and superposition [Wer92, BG95]. The number of solutions to an AC unification problem is finite, but doubly exponential in the problem size [KN92]. Instead of explicitly computing unifiers, the AC unification problems can just be kept as constraints attached to the clauses, and lazily be checked for satisfiability [Vig94, NR94]; but inspection of constraints may become necessary for concrete simplifications.

Alas, these techniques are not easy to integrate into an already existing implementation of unfailing completion: Matching and unification must take AC into account, a requirement that also severely affects the underlying indexing mechanism. Besides, AC-compatible reduction orderings must be implemented. This has motivated the search for alternatives that require less implementation effort, starting in [Hil00, Chap. 6.1] and substantially extended in [AHL03] and [Löc04]. The overall approach is to leave the framework of unfailing and resort to the more fine-grained notions of ordered

[3]Involution permutations coincide with products of disjoint transpositions. There are $\binom{n}{2j}\frac{(2j)!}{2^j j!}$ involutions with exactly j transpositions, hence $I(n) = \Sigma_{j=0}^{\lfloor n/2 \rfloor} \frac{n!}{2^j j!(n-2j)!}$. The sequence I cannot be exhibited as a sum of a fixed number of hypergeometric terms [PWZ96, Thm. 8.8.1], but is asymptotically equal to $\frac{e^{\sqrt{n}}}{\sqrt{2}e^{1/4}}\left(\frac{n}{e}\right)^{n/2}$ [CHM51].

completion. We recall the starting-point of this development, the following definition of a simple, but effective deletion rule:

AC deletion

$$\mathcal{R} \frac{s \simeq_m t}{} \mathrm{AC} \;\text{ if }\; \begin{array}{l} \cdot\ \mathrm{AC} \models s \simeq t \\ \cdot\ s \simeq_m t \notin \mathrm{AC} \end{array}$$

In Sect. 7.3 we have derived a structural property of ground confluent rewrite systems: Rewrite proofs of ground instances can always be thought of as instances of first-order level proofs. Using this property, we will demonstrate that *AC deletion* is correct in the sense that it is an instance of the general *Deletion* rule of ordered completion.

But as a prerequisite, we have to fix a policy on how labels are given, in order to give preference to proof steps with AC. Without that, smaller proofs do not always exist: Composing associativity and extended commutativity, we obtain the equation $(x+y)+z \simeq_m y+(x+z)$, a permuting variant of associativity which nevertheless is orientable with every Knuth-Bendix and lexicographic path order.[4] We consider the ground instance under the substitution $\sigma = \{x \mapsto a, y \mapsto a, z \mapsto b\}$ and assume that $\succ$ is the lexicographic path ordering induced by the precedence $b \succ + \succ a$. The complexity of $(s \simeq_m t)\sigma$ is $C_1 = \{(\{(a+a)+b\}, (x+y)+z, m, a+(a+b))\}$. The only rewrite proof of $s\sigma \longleftrightarrow^*_{\mathrm{AC}} t\sigma$ is by one application of associativity, which say carries the label n. This step has complexity $C_2 = \{(\{(a+a)+b\}, (x+y)+z, n, a+(a+b))\}$ which is smaller than C_1 only if $n < m$.

Abstracting over the concrete equations, this is exactly what we will demand of the labelling policy. For every given term s, let $\mathrm{AC}|_{s,\sigma}$ denote those equations of AC which reduce $s\sigma$ within the skeleton, namely $\mathrm{AC}|_{s,\sigma} = \{u \Rightarrow_n v \colon u \simeq_n v \in \mathrm{AC}, s \longleftrightarrow^p_{u \Rightarrow_n v} s', s\sigma \longrightarrow^p_{u \Rightarrow_n v} s'\sigma\}$. The requirement on labelling now reads a bit long-winded, but it constrains the policy as few as possible.

Assumption 7.12 If the following applies to an equation $s \simeq_m t$ with some signature extension Σ^e and ground substitution σ:
(i) $\mathrm{AC} \models s \simeq t$ (ii) $s \simeq_m t \notin \mathrm{AC}$ (iii) $s\sigma \succ t\sigma$
(iv) $s \approx_{\mathrm{enc}} u$ for every $u \Rightarrow_n v \in \mathrm{AC}|_{s,\sigma}$
then $m > n$ shall hold for some element $u \Rightarrow_n v$ of $\mathrm{AC}|_{s,\sigma}$.

The simplest realization of this requirement is to have equations from AC always bear smaller labels than the rest. However, if one sticks to the convention of unfailing completion that labels of oriented equations be smaller

[4] Replacing associativity by its permuting variant in the set AC would not affect ground confluence of $\longrightarrow_{\mathrm{AC}^\succ}$, since Prop. 7.10 is unaffected and Prop. 7.11 only needs an update of critical pairs (1) and (2).

than those of unoriented ones, then this is too coarse-grained: For example, there is a conflict between commutativity and the permuting variant of associativity from above. As a refinement, we call a labelling policy *AC conform* if in the class of equations the sides of which modulo AC and variable renaming are equal to $x + y + z$, the label of associativity in the set AC shall be smallest, and that of extended commutativity in AC smaller than those of any unorientable element.

Proposition 7.13 AC conform labelling is compatible with unfailing completion and meets the requirement of Ass. 7.12.

Proof: Up to variable renaming, any crucial term s of Ass. 7.12 because of condition (iv) is one of $x + y$, $(x + y) + z$, and $x + (y + z)$. The first and the second are covered by commutativity and associativity, respectively. In the third case, at least one of $x\sigma \succ y\sigma$ and $y\sigma \succ z\sigma$ is true, because $x\sigma \preceq y\sigma \preceq z\sigma$ would imply $t\sigma \longrightarrow^*_{\mathrm{AC}\succ} s\sigma$ by Prop. 7.10. Given $y\sigma \succ z\sigma$, then commutativity is applicable strictly within s, which therefore is smaller. Given $x\sigma \succ y\sigma$, the first proof step is via extended commutativity, and it remains to show that $s \simeq_m t$ is unorientable. We are done if t is parenthesized to the right, because then s and t unify. Otherwise, t equals $(x_1 + x_2) + x_3$ where $\{x_1, x_2, x_3\} = \{x, y, z\}$. Because of Ass. 7.9, t is greater than $x_1 + (x_2 + x_3) \equiv t'$, which unifies with s say under θ. If $s \succ t$ were true, then $s\theta \succ t\theta \succ t'\theta$ as well, contradicting $s\theta \equiv t'\theta$. Besides, $s \preceq t$ cannot hold because of $s\sigma \succ t\sigma$. □

We are now ready to prove the correctness of *AC deletion*. But please recall that both the requirements of Ass. 7.12 on labelling and of Ass. 7.9 on the reduction ordering must be met, the latter guaranteeing ground confluence of the rewrite relation $\longrightarrow_{\mathrm{AC}\succ}$.

Lemma 7.14 *AC deletion* is an instance of *Deletion*.

Proof: We consider the instance of $s \simeq_m t$ under a ground substitution σ in some Σ^e, and have to provide a proof P of $s\sigma \longleftrightarrow^*_{\mathrm{AC}} t\sigma$ such that $(s \simeq_m t)\sigma \circledcirc P$ holds. If $s\sigma \equiv t\sigma$, then the empty proof is a choice for P. Otherwise, assume that $s\sigma \succ t\sigma$. The complexity of the given proof $s\sigma \longleftrightarrow^{\lambda}_{s \Rightarrow_m t} t\sigma$ is $C_1 = \{(\{s\sigma\}, s, m, t\sigma)\}$. Since $\longrightarrow_{\mathrm{AC}\succ}$ is ground confluent (Prop. 7.11), we may apply Lem. 7.7 and obtain a proof Q of $s \longleftrightarrow^*_{\mathrm{AC}} t$ such that $Q\sigma$ is a rewrite proof. By $s\sigma \not\equiv t\sigma$ the proof Q is not empty. Hence

$$s[u\rho]_p \longleftrightarrow_{\mathrm{AC}} s[v\rho]_p \longleftrightarrow^*_{\mathrm{AC}} t$$

where $u \simeq_n v \in \mathrm{AC}$. From $s\sigma \succ t\sigma$ we obtain

$$s[u\rho]_p\sigma \longrightarrow_{\mathrm{AC}\succ} s[v\rho]_p\sigma \downarrow_{\mathrm{AC}\succ} t\sigma$$

In the new proof $Q\sigma$, the greatest term is $s\sigma$. Hence, the complexities of the proof steps within $s[v\rho]_p\sigma \downarrow_{\mathrm{AC}\succ} t\sigma$ are dominated by that of the first proof step. It is therefore sufficient to show that $C_1 = \{(\{s\sigma\}, s, m, t\sigma)\}$ is undercut by $C_2 = \{(\{s\sigma\}, u, n, s[v\rho]_p\sigma)\}$.

Because of $s \equiv s[u\rho]_p$ we know that $s \gtrsim_{\mathrm{enc}} u$ holds. If even $s >_{\mathrm{enc}} u$ is true, then $C_1 \circledcirc C_2$ is established. Otherwise we have $s \approx_{\mathrm{enc}} u$. By definition of *AC deletion*, the equation $s \simeq_m t$ does not belong to AC. By Ass. 7.12 we may assume without loss of generality that $m > n$ holds. Hence, $C_1 \circledcirc C_2$ is also true in this case. □

As a remark, we would like to illustrate that not every rewrite proof resulting from $s\sigma \downarrow_{\mathrm{AC}\succ} t\sigma$ is $\circledcirc$-smaller than $(s \simeq t)\sigma$. Consider the substitution $\sigma = \{x \mapsto c + c\}$, and let $s_i \simeq t_i$ denote the equation

$$x + (x + (\ldots + (x + c)\ldots)) \simeq c + (x + (\ldots + (x + x)\ldots))$$

where x occurs i-times on each side. Note that AC $\models s_i \simeq t_i$ and $s_i\sigma \succ t_i\sigma$ hold. Proofs of $s_i\sigma \downarrow_{\mathrm{AC}\succ} t_i\sigma$ may start with a top-level application of associativity like in

$$s_1\sigma \equiv (c + c) + c \longrightarrow_{\mathrm{AC}\succ} c + (c + c) \longrightarrow^*_{\mathrm{AC}\succ} t_1\sigma$$

Disregarding components (iii) and (iv), the complexity of this proof is $\{((c + c) + c, (x + y) + z)\}$, whereas $(s_1 \simeq t_1)\sigma$ amounts to $\{((c + c) + c, x + c)\}$. Because of $(x + y) + z >_{\mathrm{enc}} x + c$, the above proof is $\circledcirc$-greater than the original one, and therefore useless in showing $s_1 \simeq t_1$ redundant.

Looking closer at this example, the malicious proof begins with a fringe step, whereas our result in Sect. 7.3 guarantees that rewrite proofs exist without rewriting in the fringe or in the substitutes. In this case, commutativity would do the job.

The property that AC-joinability is just by skeleton steps has not been made explicit, neither been shown, in previous correctness proofs of *AC deletion*, but was employed nevertheless. Taking the above example $(s_i \simeq t_i)\sigma$, this gap shows up in [Löc05, p. 115] when $i = 1$, in [Hil00, p. 133f] when $i = 2$, and in [AHL03, p. 226f] when $i = 3$.

Lemma 7.14 straightforwardly extends to arbitrary ground confluent systems E in place of AC: Reinspecting the proof of Lem. 7.14, no other property of AC is exploited but ground confluence. Furthermore, Assumption 7.12 can be imposed on any set of equations. It can always be met by the simple realization sketched above, or refinements thereof. This gives rise to the following generalization:

Corollary 7.15 Assume that E is ground confluent in every Σ^e, and that Ass. 7.12 is met with E in place of AC. Then the following reduction rule is an instance of *Deletion*:
E deletion:

$$\mathcal{R} \frac{s \simeq_m t}{} E \quad \text{if} \begin{array}{l} \cdot\ E \models s \simeq t \\ \cdot\ s \simeq_m t \notin E \end{array}$$

This deletion rule strictly improves upon rewrite-based simplification techniques whenever E is not confluent, but only ground confluent. To give some examples, such systems have been reported for AC plus idempotence, groups of exponent 2, AC plus distributivity, Boolean rings [MN90], and Abelian groups [AHL03].

Upon introduction of the *AC deletion* rule into the WALDMEISTER system, experiments were conducted on proof problems from the TPTP library [SS98], such that the effects of the rule could be assessed in a quantitative manner. The following table, reprinted from [Hil00, p. 134], holds data for a system version WM-AC which employs the rule, and for another version WM-STD which does not. When completing the input axiomatization, WALDMEISTER distinguishes active facts, which induce a rewrite relation, and passive facts, which are the one-step conclusions of the active ones up to redundancy. The measured quantities are the overall numbers $|A|$ and $|P|$ of active and passive facts until a proof was found, and the run-time $|t|$. The experiments were carried out on a SPARCStation Ultra-10/333 MHz.

	$\|A\|$		$\|P\|$		t	
	WM-STD	WM-AC	WM-STD	WM-AC	WM-STD	WM-AC
ROB005-1	312	212	169 000	33 000	25.5	2.1
RNG027-5	1 081	293	416 000	48 000	271.2	16.2
LAT023-1	327	248	123 000	66 000	12.0	5.7
RNG035-7	554	495	237 000	161 000	29.8	21.5
GRP180-1	426	422	83 000	88 000	4.7	4.9

In this test series, the figures $|A|$ and $|P|$ were reduced by factors of up to 3.7 and 8.7, whereas run-time was cut down by a factor of up to 16.7. All in all, *AC deletion* has proven advantageous for many proof problems that contain AC operators.

7.6 AC Deletion in a Clausal Setting

The first-order provers E and PROVER9 are based on variants of superposition. According to [Sch01, p. 371] and [McC08, Sect. *Process Inferred*], they

feature the following simplification rules:

AC clause deletion

$$\mathcal{R}\,\frac{C \vee s \simeq t}{}\,\mathrm{AC} \;\text{ if }\; \begin{array}{l} \cdot\ \mathrm{AC} \models s \simeq t \\ \cdot\ C \vee s \simeq t \notin \mathrm{AC} \end{array}$$

AC literal deletion

$$\mathcal{R}\,\frac{C \vee s \not\simeq t}{C}\,\mathrm{AC} \text{ if } \cdot\ \mathrm{AC} \models s \simeq t$$

Using the standard redundancy notion of superposition, say as in [BG94, Sect. 5] or in [NR01, Sect. 4], it is not possible to justify these rules. Consider the clause $C \equiv a+(c+b) \simeq c+(b+a)$, and assume that $\succ$ is a lexicographic path ordering induced by the precedence $+ \succ a \succ b \succ c$. Then $\mathrm{AC} \models C$ holds, but $\mathrm{AC}^{\prec C} \models C$ does not: The only non-increasing clause instance applicable to the left-hand side of C is $D \equiv a + (c + b) \simeq c + (a + b)$, via extended commutativity. However, D is bigger in the clause ordering than C, by comparison of right-hand sides. Therefore, the clause C is *not* redundant with respect to AC.

Going further, if the clause C is used in redundancy proofs and then erroneously discarded, then this may even lead to incompleteness. Given the clauses $D_1 \equiv c+(b+a) \not\simeq c$ and $D_2 \equiv a+(c+b) \simeq c$, the set $\mathrm{AC} \cup \{C, D_1, D_2\}$ is unsatisfiable. Now we produce a spurious derivation as follows:

(i) As suggested by *AC clause deletion*, the set AC is indeed saturated: According to Prop. 7.11, for every ground instance $l \simeq r$, $s[l] \simeq t \vdash s[r] \simeq t$ of an inference with premises from AC there exists a rewrite proof of $s[r] \longleftrightarrow^*_{\mathrm{AC}} t$. The terms of the participating clause instances are bounded by the maximum of $s[r]$ and t. Because of $l \succ r$ and $s[l] \succ t$, all these clause instances are smaller than $s[l] \simeq t$. So the inference conclusion follows from clauses below the main premise, turning the inference redundant.

(ii) There is no inference between D_1 and D_2, and none between D_1 and AC. But there is one between D_2 and AC, namely

$$D_2,\ x + (y + z) \simeq y + (x + z) \vdash\ c + (a + b) \simeq c$$

Its only ground instance is

$$a + (c + b) \simeq c,\ a + (c + b) \simeq c + (a + b) \vdash\ c + (a + b) \simeq c$$

However, this conclusion follows already from smaller instances of $\{C,\ x + y \simeq y + x,\ D_2\}$, because

$$\begin{array}{ll} a + (c + b) \simeq c + (b + a) & \\ \quad b + a \simeq a + b & \models\ c + (a + b) \simeq c \\ a + (c + b) \simeq c & \end{array}$$

Hence, the inference between D_2 and AC is redundant.

(iii) If discarding the clause C via *AC clause deletion* were correct, then the set $\mathrm{AC} \cup \{D_1, D_2\}$ would be saturated. That is, the derivation could stop without producing the empty clause.

One might conjecture that any satisfying solution of this problem would require to transfer the more fine-grained redundancy notion of completion into the framework of superposition. In particular, completeness of the former is shown with a proof-theoretic method, whereas refutational completeness of paramodulation or superposition usually is deduced with semantic methods. An early exception is a paper by L. Bachmair ([Bac89]), revealing that "proof-theoretic methods are difficult to apply to paramodulation-like calculi, as the corresponding derivations reveal some intricate non-local structure" [BG98a, p. 373]. Ongoing research in this direction is to employ the notion of proof ordering for a unifying presentation of reasoning procedures like completion, unification, Gröbner bases and resolution [DK06, BD07].

Fortunately, the problem can be already solved by simpler means, without proof-theoretic machinery. For the sake of simplicity, we restrict to equational clauses; the extension to arbitrary predicate symbols is straightforward. The key point is to employ a definition of literal ordering more general than the one Def. 2.26. There, equational literals are first assigned a complexity according to $s \simeq t \mapsto \{s, t\}$ and $s \not\simeq t \mapsto \{s, s, t, t\}$; and then the literal ordering is defined by comparing the respective complexities in the multiset extension of the given reduction ordering. More generally [BG98a, Sect. 6], one can work with any well-founded ordering $\succ$ on literals and terms which is *admissible* in the sense that (i) $\succ$ is total on ground literals, (ii) $\succ$ on terms is a ground total reduction ordering, (iii) if L and L' are ground literals, then $L \succ L'$ must hold whenever (iii.1) $\max(L) \succ \max(L')$, or (iii.2) $\max(L) = \max(L')$, L is negative, L' is positive. Here $\max(L)$ denotes the maximal term of the literal L. The multiset extension of an ordering on literals yields an ordering on clauses with which refutational completeness of superposition can be established.

Before we give a concrete instance of an admissible ordering, let us stipulate that from now on, literals shall be annotated with labels from some well-founded, totally ordered domain, as introduced for equations in the context of ordered completion in Sect. 7.2. For reasons of simplicity, labels used within AC shall always be smaller than those used without. Of course, this labelling policy can be liberalized similar to the AC conform one in Sect. 7.5.

Definition 7.16 Given a ground total reduction ordering $\succ$ on terms, we redefine its extension to literals and clauses as follows:

(i) The complexity of a literal $s \bowtie_m t$, where $\bowtie \in \{\simeq, \not\simeq\}$, now is defined

according to $s \bowtie_m t \longmapsto \begin{cases} (\{s\}, & \bowtie, m, t\ ) & \text{if } s \succ t \\ (\{t\}, & \bowtie, m, s\ ) & \text{if } t \succ s \\ (\{s,t\}, & \bowtie, m, \bot) & \text{otherwise} \end{cases}$
where the symbol $\bot$ shall stand for some fixed element.

(ii) Complexities are compared in the lexicographic combination of the following orderings:
   (a) the multiset extension of the reduction ordering $\succ$,
   (b) an ordering in which $\not\simeq$ is greater than $\simeq$,
   (c) the ordering on labels,
   (d) the reduction ordering $\succ$.
   Literals are compared in terms of their complexities.

(iii) As before, the clause ordering takes the multisets of the literal complexities and compares them in the multiset extension of the literal ordering.

This new ordering is admissible just by construction. On literals bearing the same label, it should essentially coincide with the old ordering. Fortunately, when it comes to AC deletion in a clausal setting, then it is an improvement:

Lemma 7.17 The simplification rules *AC clause deletion* and *AC literal deletion* are correct with respect to the clause ordering of Def. 7.16 and the stipulated labelling policy.

Proof:

(i) Concerning *AC clause deletion*, assume that both $\mathrm{AC} \models s \simeq_m t$ and $C \vee s \simeq_m t \notin \mathrm{AC}$ hold, and consider an arbitrary ground substitution σ. We will show that $\mathrm{gnd}(\mathrm{AC})^{\prec(s \simeq_m t)\sigma} \models (s \simeq_m t)\sigma$ holds; so $C \vee s \simeq_m t$ will be redundant with respect to AC. We assume without loss of generality that $s\sigma \succ t\sigma$.
   According to Prop. 7.11, the rewrite relation $\longrightarrow_{\mathrm{AC}^\succ}$ is ground confluent. Hence, there exists a ground rewrite proof of $s\sigma \longleftrightarrow^*_{\mathrm{AC}} t\sigma$, say

$$\begin{array}{ccccccccc} s\sigma & & & & & & & & t\sigma \\ \| & & & & & & & & \| \\ s_0 & \longrightarrow^{p_1}_{u_1 \Rightarrow_{m_1} v_1} & \cdots & \longrightarrow^{p_i}_{u_i \Rightarrow_{m_i} v_i} & s_i & \longleftarrow^{p_{i+1}}_{u_{i+1} \Rightarrow_{m_{i+1}} v_{i+1}} & \cdots & \longleftarrow^{p_n}_{u_n \Rightarrow_{m_n} v_n} & s_n \end{array}$$

   Let ρ_j denote, for every j, the minimal substitution such that $s_{j-1}|_{p_j} \equiv u_j\rho_j$ and $s_j|_{p_j} \equiv v_j\rho_j$ hold; and let $e_j \equiv u_j \simeq_{m_j} v_j$. The above rewrite proof semantically means that $\bigwedge_j e_j\rho_j \models (s \simeq_m t)\sigma$ is true. It remains to prove $e_j\rho_j \prec (s \simeq_m t)\sigma$. Because of $s_0 \equiv s\sigma \succ t\sigma \equiv s_n$, the term s_0 is the strictly greatest of all s_j. Therefore, for $j > 1$ we obtain $(s \simeq_m t)\sigma \equiv s_0 \simeq_m s_n \succ s_{j-1} \simeq_{m_j} s_j \succeq e_j\rho_j$. In case $j = 1$, we need

$(\{s\sigma\}, \simeq, m, t\sigma) \succ (\{u_1\rho_1\}, \simeq, m_1, v_1\rho_1)$. If $p_1 \neq \lambda$, then this is settled by the subterm property of $\succ$. Otherwise, the labelling policy applies.

(ii) As to *AC literal deletion*, the simplified clause C is clearly smaller than the original one $C \vee s \not\simeq t$. In turn, $C \vee s \not\simeq t$ is redundant already with respect to C. Finally, $C \vee s \not\simeq t$ and AC imply C because AC $\models s \simeq t$ by assumption.

□

Justifying *AC clause deletion* in this clausal setting here is much simpler than the correctness proof of *AC deletion*. In particular, there is no reference to the particular property of ground confluent systems developed in Sect. 7.3, namely that rewrite proofs of ground instances require no rewriting at the fringe or within the substitution part.

However, it must be noted that the complexity notion of Def. 7.16 does not record information about the substitution by which an instance of a literal is obtained. This excludes many simplifications that are allowed from the completion point of view. For example, if $\succ$ is the lexicographic path ordering induced by the precedence $f \succ a \succ b \succ c$, then $f(a) \simeq_m c$ can be simplified with $f(x) \simeq_n b$ into $b \simeq c$ only in case that $m > n$ holds.

This deficiency cannot be overcome without refining the notion of literal complexity again. The tuple from above can for example be extended with the uninstantiated term, say right in front of the label, and perform comparison on this component with respect to the encompassment ordering. So if $(s \simeq_m t)\sigma$ is a ground instance of $s \simeq_m t$ and $s\sigma$ is strictly maximal, then we now obtain $(\{s\sigma\}, \simeq, s, m, t\sigma)$. Notably, in order to adapt the above correctness proof of *AC clause deletion* to this refinement, the mentioned result of Sect. 7.3 now is needed.

Even more, if $(s \simeq_m t)\sigma$ is used to prove a ground equation $v \simeq_n w$ redundant where $v \succ w$, then there is not much difference any more between the superposition view and the completion view:

(i) In the former, the complexity of $(s \simeq_m t)\sigma$, which is $(\{s\sigma\}, \simeq, s, m, t\sigma)$, must be below $(\{v\}, \simeq, v, n, w)$.

(ii) In the latter, the complexity is derived from the concrete application, say $u[s\sigma]_p \longrightarrow^p_{s \Rightarrow_m t} u[t\sigma]_p$, namely as $(\{u[s\sigma]\}, s, m, u[t\sigma])$. It must be smaller than $(\{v\}, v, n, w)$.

The only essential difference is whether or not the term context $u[\,]$ occurs in the complexity of $(s \simeq_m t)\sigma$ in the first and in the last component. Because reduction orderings have the subterm property, condition (ii) implies condition (i). In other words, completion redundancy implies superposition redundancy. The converse, however, does not hold: Let $\succ$ denote the lexicographic path ordering induced by the precedence $f \succ a \succ b$, and consider

the entailment

$$f(a) \simeq a,\ a \simeq b \models f(f(b)) \simeq f(a)$$

The conclusion is greater than the premises with respect to the clause ordering, and hence redundant in the superposition view. But the smallest proof of the conclusion is

$$f(f(b)) \longleftarrow^{1.1}_{b \Rightarrow a} f(f(a)) \longrightarrow^{1}_{f(a) \Rightarrow a} f(a)$$

The first proof step is increasing and produces a term greater than any of the above conclusion. Therefore, the latter is *not* redundant in the completion view. However, if one drops the above-mentioned term context $u[\ ]$ in the complexity definition for completion, then both redundancy notions will coincide.

Summing it up, in this section we have demonstrated that *AC clause deletion* cannot be justified with the standard redundancy notion of superposition. Furthermore, we have shown that this nuisance can be overcome with a refined construction of the ordering on the level of literals. Then we have added another refinement on this level such that subsumption between left-hand sides leads to simplification. We have argued that the resulting notion of superposition redundancy finally subsumes completion redundancy. Therefore, our construction bridges a gap between these two closely related reasoning procedures.

8 Future Directions

In this thesis, we have studied the interplay of superposition and decision procedures in various instances. For each of them, we now give a short summary and directions for future work.

An integration of a Shostak theory into the superposition calculus was presented in Chapter 3. The Shostak-style components for deciding the clausal validity problem, namely canonizer and solver, have been employed as simplification devices, so that no coherence pairs between theory axioms and other axioms have to be considered. Under the assumption that the solver meets some ordering restriction, as is the case for linear arithmetic, our calculus is refutationally complete on mixed ground clauses.

When lifting this calculus to non-ground clauses, variables that occur directly below theory contexts turn out to be unmanageable: Ground instances thereof may arbitrarily enlarge the theory context, and the set of all canonizations or solved forms of these ground instances is in general not finitely representable. This case can only be handled if more information on the internals of the canonizer and the solver is available. After all, the list of relevant Shostak theories is easy to survey, and linear arithmetic is by far the most important one. White-box approaches for integrating this theory into superposition have been pursued in [BGW94], [Wal01] and [KV07]. At least, our ground-level calculus improves over Shostak's method in that it can detect the unsatisfiability of infinite sets of ground clauses, and might therefore also be of interest if combined with instantiation.

The Nelson-Oppen procedure for the combination of decision procedures has been revisited in Chapter 4. In particular, we have used the model generation method in order to exhibit a joint model of the combined theories and the query in case that the procedure reports satisfiability: Treating the combination aspect was eased by the fact that saturatedness of clause sets is closed under signature-disjoint unions, provided there are no isolated variable occurrences. The resulting new correctness proof is quite concise, because it is based on superposition turned into a powerful proof-theoretic tool.

To simplify the presentation, we have restricted ourselves to the important special case that the component theories enjoy the stronger property of convexity, such that the Nelson-Oppen procedure comes without branching. Dropping the convexity requirement is the next step of generalization, after which extensions to non-disjoint signatures (see [GNZ05] for a survey) might be attempted, which is particularly promising with refinements of superposition like the chaining calculus for transitive relations.

The subject of Chapter 5 was the analysis of an approach to bitvector reasoning which arose in the context of the Verisoft project. On the level of bits, one restricts equational literals to the shapes $t \simeq 0$ and $t \simeq 1$, which is possible without loss of generality. We have given an axiomatization the effect of which on compound left-hand sides t during saturation just is to dissolve the encompassing equation, such that for example $\overline{\overline{c}} \ominus c \simeq 0$ leads to the two clauses $c \simeq 0$ and $c \simeq 1$. This way, the theory of bit operators is reduced to that of $0 \not\simeq 1$. On the resulting clauses, superposition with standard simplifications proceeds similar to resolution, but can naturally be augmented with splitting. To this kind of bit-level reasoning, the bitvector level is coupled by a bitwise definition of vector equality.

In practice, superposition-based theorem provers are lost if confronted with the straightforward axiomatization of bitvectors. With our transformational approach, bitvector reasoning becomes possible to some extent. Essentially, our approach on the fly reduces vector expressions to the propositional level, which also includes clausification. Concerning SPASS, the latter has just now smartly been solved in [Rus08] using the renaming module of FLOTTER. As to the propositional reasoning, SPASS is too far behind modern SAT solvers, lacking much of the technology which makes these successful. This issue should be addressed. One could then assess whether the above-sketched reduction on the fly improves over bitblasting. Finally, from a more general point of view, bitvector reasoning truly on the vector level might be promising.

In Chapter 6, we have presented a light-weight adaptation of superposition calculi to the first-order theory of bounded domains, which is based on a modification of lifting. As a consequence, the range of inference unifiers can be restricted to digits and variables. Furthermore, ordering restrictions as well as the general semantic redundancy notion become effective. Most importantly, our refinement can be embedded into any general first-order setting via a sort discipline based on monadic predicates. We have shown that a decision procedure is obtained with a calculus configuration in which non-Horn clauses are dealt with not by equality factoring, but by aggressive splitting, and in which ordered rewriting is combined with some instantiation. Notably, this covers the Bernays-Schönfinkel class as well.

Our primary goal in formulating a decision procedure was to re-establish decidability in a superposition framework; so there should still be room for improvement. From a more practical point of view, one might want to get rid of the coupling of rewriting and instantiation, and minimize the number of splitting steps. Attempting any such variations should, however, be guided by practical experimentations, which requires that our calculus fully be implemented. One should then also attack the detection and elimination of symmetries. The latter in our setting simply is a satisfiability-preserving transformation.

We have started Chapter 7 with a recapitulation of ordered completion. Then we have demonstrated how, in this setting, rewrite systems that are confluent only on the level of ground terms can be exploited for eliminating redundancies. We have shown that this criterion does not carry over to superposition with the commonly used redundancy notion, but that this problem can easily be fixed with a refined literal ordering. Even more, superposition redundancy can be extended such that it subsumes completion redundancy.

In the context of superposition, a natural extension of the mentioned redundancy criterion would be to take conditional equations into account. As to the refinement of superposition redundancy itself, practical relevance of using labels can be expected from preliminary experiments with SPASS, in particular for the application within contextual rewriting.

Bibliography

[AHL03] J. Avenhaus, Th. Hillenbrand, and B. Löchner. On using ground joinable equations in equational theorem proving. *Journal of Symbolic Computation*, 36(1–2):217–233, 2003.

[ARR01] A. Armando, S. Ranise, and M. Rusinowitch. Uniform derivation of decision procedures by superposition. In L. Fribourg, editor, *Proceedings of the 15th International Workshop on Computer Science Logic*, volume 2142 of *LNCS*, pages 513–527. Springer-Verlag, 2001.

[Bac87] L. Bachmair. *Proof Methods for Equational Theories*. PhD thesis, University of Illinois at Urbana-Champaign, 1987.

[Bac89] L. Bachmair. Proof normalization for resolution and paramodulation. In N. Dershowitz, editor, *Proceedings of the Third International Conference on Rewriting Techniques and Applications*, volume 355 of *LNCS*, pages 15–28. Springer-Verlag, 1989.

[Bac91] L. Bachmair. *Canonical Equational Proofs*. Birkhäuser, 1991.

[BCGN07] F. Baader, B. Cook, J. Giesl, and R. Nieuwenhuis, editors. *Deduction and Decision Procedures – Abstracts Collection*, volume 07401 of *Dagstuhl Seminar Proceedings*, 2007.

[BD07] M. P. Bonacina and N. Dershowitz. Abstract canonical inference. *ACM Transactions on Computational Logic*, 8(1), Art. 6, 2007.

[BDH86] L. Bachmair, N. Dershowitz, and J. Hsiang. Orderings for equational proofs. In *Proceedings of the First IEEE Symposium on Logic in Computer Science*, pages 346–357. North-Holland, 1986.

[BdMS05] C. Barrett, L. de Moura, and A. Stump. SMT-COMP: Satisfiability Modulo Theories Competition. In K. Etessami and S. Rajamani, editors, *Proceedings of the 17th International Conference on Computer-Aided Verification*, volume 3576 of *Lecture Notes in Computer Science*, pages 20–23. Springer-Verlag, 2005.

[BDP89] L. Bachmair, N. Dershowitz, and D. A. Plaisted. Completion without failure. In H. Aït-Kaci and M. Nivat, editors, *Resolu-*

tion of Equations in Algebraic Structures, volume 2, pages 1–30. Academic Press, 1989.

[BDS02] C. W. Barrett, D. L. Dill, and A. Stump. A generalization of Shostak's method for combining decision procedures. In A. Armando, editor, *Proceedings of the 4th International Workshop on Frontiers of Combining Systems*, volume 2309 of *LNCS*, pages 35–70. Springer-Verlag, 2002.

[BFdNT06] P. Baumgartner, A. Fuchs, H. de Nivelle, and C. Tinelli. Computing finite models by reduction to function-free clause logic. In W. Ahrendt, P. Baumgartner, and H. de Nivelle, editors, *Proceedings of the Third Workshop on Disproving*, pages 82–99, 2006.

[BG91] L. Bachmair and H. Ganzinger. Perfect model semantics for logic programs with equality. In K. Furukawa, editor, *Proceedings of the 8th International Conference on Logic Programming*, pages 645–659. MIT Press, 1991.

[BG94] L. Bachmair and H. Ganzinger. Rewrite-based equational theorem proving with selection and simplification. *Journal of Logic and Computation*, 4(3):217–247, 1994.

[BG95] L. Bachmair and H. Ganzinger. Associative-commutative superposition. In N. Dershowitz and N. Lindenstrauss, editors, *Proceedings of the 4th International Workshop on Conditional and Typed Rewrite Systems*, volume 968 of *Lecture Notes in Computer Science*, pages 1–14. Springer-Verlag, 1995.

[BG98a] L. Bachmair and H. Ganzinger. Equational reasoning in saturation-based theorem proving. In W. Bibel and P. H. Schmitt, editors, *Automated Deduction: A Basis for Applications*, volume I, chapter 11, pages 353–397. Kluwer Academic Publishers, 1998.

[BG98b] L. Bachmair and H. Ganzinger. Ordered chaining calculi for first-order theories of transitive relations. *Journal of the ACM*, 45(6):1007–1049, 1998.

[BGLS95] L. Bachmair, H. Ganzinger, Chr. Lynch, and W. Snyder. Basic paramodulation. *Information and Computation*, 121(2):172–192, 1995.

[BGN+06] M. P. Bonacina, S. Ghilardi, E. Nicolini, S. Ranise, and D. Zucchelli. Decidability and undecidability results for Nelson-Oppen and rewrite-based decision procedures. In U. Furbach and N. Shankar, editors, *Proceedings of the Third International Joint Conference on Automated Reasoning*, volume 4130 of *LNAI*, pages 513–527. Springer-Verlag, 2006.

[BGW94] L. Bachmair, H. Ganzinger, and U. Waldmann. Refutational theorem proving for hierarchic first-order theories. *Applicable Algebra in Engineering, Communication and Computing*, 5(3/4):193–212, 1994.

[Bir35] G. Birkhoff. On the structure of abstract algebras. *Proceedings of the Cambridge Philosophical Society*, 31:433–454, 1935.

[BK95] D. A. Basin and N. Klarlund. Hardware verification using monadic second-order logic. In P. Wolper, editor, *Proceedings of the 7th International Conference on Computer-Aided Verification*, volume 939 of *Lecture Notes in Computer Science*, pages 31–41. Springer-Verlag, 1995.

[BN98] F. Baader and T. Nipkow. *Term Rewriting and All That*. Cambridge University Press, 1998.

[BP98] N. S. Bjørner and M. C. Pichora. Deciding fixed and non-fixed size bit-vectors. In B. Steffen, editor, *Proceedings of the 4th International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, volume 1384 of *Lecture Notes in Computer Science*, pages 376–392. Springer-Verlag, 1998.

[BS28] P. Bernays and M. Schönfinkel. Zum Entscheidungsproblem der mathematischen Logik. *Mathematische Annalen*, 99:342–372, 1928.

[BS06] P. Baumgartner and R. Schmidt. Blocking and other enhancements for bottom-up model generation methods. In U. Furbach and N. Shankar, editors, *Proceedings of the Third International Joint Conference on Automated Reasoning*, volume 4130 of *LNAI*, pages 125–139. Springer-Verlag, 2006.

[Bür90] H.-J. Bürckert. A resolution principle for clauses with constraints. In M. E. Stickel, editor, *Proceedings of the 10th International Conference on Automated Deduction*, volume 449 of *LNAI*, pages 178–192. Springer-Verlag, 1990.

[Che86] Ph. Le Chenadec. *Canonical Forms in Finitely Presented Algebras*. John Wiley & Sons, 1986.

[CHM51] S. Chowla, I. N. Herstein, and K. Moore. On recursions connected with symmetric groups I. *Canadian Journal of Mathematics*, 3:328–334, 1951.

[CK73] C. C. Chang and H. J. Keisler. *Model Theory*, volume 73 of *Studies in Logic and the Foundation of Mathematics*. North-Holland, 1973.

[CMR97] D. Cyrluk, O. Möller, and H. Rueß. An efficient decision procedure for the theory of fixed-sized bit-vectors. In O. Grumberg, editor, *Proceedings of the 9th International Conference on*

Computer-Aided Verification, volume 1254 of *Lecture Notes in Computer Science*, pages 60–71. Springer-Verlag, 1997.

[CMSM04] S. Colton, A. Meier, V. Sorge, and R. McCasland. Automatic generation of classification theorems for finite algebras. In D. Basin and M. Rusinowitch, editors, *Proceedings of the Second International Joint Conference on Automatic Reasoning*, volume 3097 of *LNAI*, pages 400–414. Springer-Verlag, 2004.

[Com90] H. Comon. Solving symbolic ordering constraints. *International Journal of Foundations of Computer Science*, 1(4):387–411, 1990.

[CS03] K. Claessen and N. Sörensson. New techniques that improve MACE-style finite model finding. In P. Baumgartner and Chr. Fermueller, editors, *Proceedings of the Workshop on Model Computation*, 2003.

[DdM06] B. Dutertre and L. de Moura. The YICES SMT solver, 2006. Available via `http://yices.csl.sri.com`.

[Der79] N. Dershowitz. A note on simplification orderings. *Information Processing Letters*, 9:212–215, 1979.

[Der87] N. Dershowitz. Termination of rewriting. *Journal of Symbolic Computation*, 3(1/2):69–116, 1987. Corrigendum: JSC 4(3): 409–410, 1987.

[Der91] N. Dershowitz. A maximal-literal unit strategy for Horn clauses. In S. Kaplan and M. Okada, editors, *Proceedings of the Second International Workshop on Conditional and Typed Rewriting Systems*, volume 516 of *LNCS*, pages 14–25. Springer-Verlag, 1991.

[DK06] N. Dershowitz and C. Kirchner. Abstract canonical presentations. *Theoretical Computer Science*, 357(1–3):53–69, 2006.

[DLL62] M. Davis, G. Logemann, and D. Loveland. A machine program for theorem proving. *Communications of the ACM*, 5(7):394–397, 1962.

[DM79] N. Dershowitz and Z. Manna. Proving termination with multiset orderings. *Communications of the ACM*, 22(8):465–476, 1979.

[dNM06] H. de Nivelle and J. Meng. Geometric resolution: A proof procedure based on finite model search. In U. Furbach and N. Shankar, editors, *Proceedings of the Third International Joint Conference on Automated Reasoning*, volume 4130 of *LNAI*, pages 303–317. Springer-Verlag, 2006.

[DP60] M. Davis and H. Putnam. A computing procedure for quantification theory. *Journal of the ACM*, 7(3):201–215, 1960.

[End02] H. B. Enderton. *A Mathematical Introduction to Logic*. Academic Press, second edition, 2002.

[FGR05] M.-L. Fernández, G. Godoy, and A. Rubio. Recursive path orderings can also be incremental. In G. Sutcliffe and A. Voronkov, editors, *Proceedings of the 12th International Conference on Logic for Programming, Artificial Intelligence and Reasoning*, volume 3835 of *LNAI*, pages 230–245. Springer-Verlag, 2005.

[Gan02] H. Ganzinger. Shostak light. In A. Voronkov, editor, *Proceedings of the 18th International Conference on Automated Deduction*, volume 2392 of *LNAI*, pages 332–346. Springer-Verlag, 2002.

[GHW03] H. Ganzinger, Th. Hillenbrand, and U. Waldmann. Superposition modulo a Shostak theory. In F. Baader, editor, *Proceedings of the 19th International Conference on Automated Deduction*, volume 2741 of *LNAI*, pages 182–196. Springer-Verlag, 2003.

[GN00] G. Godoy and R. Nieuwenhuis. Paramodulation with built-in Abelian groups. In *Proceedings of the 15th IEEE Symposium on Logic in Computer Science*, pages 413–424. IEEE Computer Society Press, 2000.

[GNZ05] S. Ghilardi, E. Nicolini, and D. Zucchelli. A comprehensive framework for combined decision procedures. In B. Gramlich, editor, *Proceedings of the 5th International Workshop on Frontiers of Combining Systems*, volume 3717 of *LNAI*, pages 1–30. Springer-Verlag, 2005.

[HBF96] Th. Hillenbrand, A. Buch, and R. Fettig. On gaining efficiency in completion-based theorem proving. In H. Ganzinger, editor, *Proceedings of the 7th International Conference on Rewriting Techniques and Applications*, volume 1103 of *LNCS*, pages 432–435. Springer-Verlag, 1996.

[Hil00] Th. Hillenbrand. Schnelles Gleichheitsbeweisen: Vom Vervollständigungskalkül zum WALDMEISTER-System. Diplomarbeit, Universität Kaiserslautern, Fachbereich Informatik, 2000.

[Hil04] Th. Hillenbrand. A superposition view on Nelson-Oppen. In U. Sattler, editor, *Contributions to the Doctoral Programme of the Second International Joint Conference on Automated Reasoning*, volume 106 of *CEUR Workshop Proceedings*, pages 16–20, 2004.

[HP96] J. L. Hennessy and D. A. Patterson. *Computer Architecture: A Quantitative Approach*. Morgan Kaufmann, second edition, 1996.

[HR87] J. Hsiang and M. Rusinowitch. On word problems in equational theories. In Th. Ottmann, editor, *Proceedings of the 14th In-*

ternational Colloquium on Automata, Languages and Programming, volume 267 of *LNCS*, pages 54–71. Springer-Verlag, 1987.

[HTW06] Th. Hillenbrand, D. Topic, and Chr. Weidenbach. Sudokus as logical puzzles. In W. Ahrendt, P. Baumgartner, and H. de Nivelle, editors, *Proceedings of the Third Workshop on Disproving*, pages 2–12, 2006.

[JM92] J.-P. Jouannaud and C. Marché. Termination and completion modulo associativity, commutativity and identity. *Theoretical Computer Science*, 104(1):29–51, 1992.

[Kap02] D. Kapur. A rewrite rule based framework for combining decision procedures. In A. Armando, editor, *Proceedings of the 4th International Workshop on Frontiers of Combining Systems*, volume 2309 of *LNAI*, pages 87–102. Springer-Verlag, 2002.

[KB70] D. E. Knuth and P. B. Bendix. Simple word problems in universal algebras. In J. Leech, editor, *Computational Problems in Abstract Algebra*, pages 263–297. Pergamon Press, 1970.

[KC03] S. Krstić and S. Conchon. Canonization for disjoint unions of theories. In F. Baader, editor, *Proceedings of the 19th International Conference on Automated Deduction*, volume 2741 of *LNAI*, pages 197–211. Springer-Verlag, 2003.

[KL80] S. Kamin and J.-J. Levy. Attempts for generalizing the recursive path orderings. Available electronically from `http://perso.ens-lyon.fr/pierre.lescanne/not_accessible.html`. University of Illinois, Department of Computer Science. Unpublished note, 1980.

[KMN85] D. Kapur, D. R. Musser, and P. Narendran. Only prime superpositions need be considered in the Knuth-Bendix procedure. Unpublished manuscript, Computer Science Branch, Corporate Research and Development, General Electric, Schenectady, New York, 1985.

[KN92] D. Kapur and P. Narendran. Complexity of unification problems with associative-commutative operators. *Journal of Automated Reasoning*, 9(2):261–288, 1992.

[Kru60] J. B. Kruskal. Well-quasi-orderings, the tree theorem, and Vazsonyi's conjecture. *Transactions of the American Mathematical Society*, 95:210–225, 1960.

[KV07] K. Korovin and A. Voronkov. Integrating linear arithmetic into superposition calculus. In J. Duparc and Th. A. Henzinger, editors, *Proceedings of the 21st International Workshop on Computer Science Logic*, volume 4646 of *LNCS*, pages 223–237. Springer-Verlag, 2007.

[Lan75] D. S. Lankford. Canonical inference. Technical Report ATP-32, Department of Mathematics and Computer Science, University of Texas, Austin, 1975.

[LB77] D. S. Lankford and A. M. Ballantyne. Decision procedures for simple equational theories with commutative-associative axioms: Complete sets of commutative-associative reductions. Technical Report ATP-39, University of Texas, Austin, 1977.

[LH02] B. Löchner and Th. Hillenbrand. A phytography of WALDMEISTER. *AI Communications*, 15(2–3):127–133, 2002.

[LO06] Inês Lynce and Joël Ouaknine. Sudoku as a SAT problem. In *Electronic Proceedings of the 9th International Symposium on Artificial Intelligence and Mathematics*, 2006. Available from `http://anytime.cs.umass.edu/aimath06/proceedings.html`.

[Löc04] B. Löchner. A redundancy criterion based on ground reducibility by ordered rewriting. In D. Basin and M. Rusinowitch, editors, *Proceedings of the Second International Joint Conference on Automated Reasoning*, volume 3097 of *LNAI*, pages 45–59. Springer-Verlag, 2004.

[Löc05] B. Löchner. *Advances in Equational Theorem Proving – Architecture, Algorithms, and Redundancy Avoidance*. PhD thesis, Technische Universität Kaiserslautern, 2005.

[Mar94] C. Marché. Normalised rewriting and normalised completion. In *Proceedings of the 9th IEEE Symposium on Logic in Computer Science*, pages 394–403. IEEE Computer Society Press, 1994.

[MB88] R. Manthey and F. Bry. SATCHMO: a theorem prover implemented in Prolog. In E. Lusk and R. Overbeek, editors, *Proceedings of the 9th International Conference on Automated Deduction*, volume 310 of *LNCS*, pages 415–434. Springer-Verlag, 1988.

[McC03] W. McCune. MACE4 reference manual and guide. Technical Report ANL/MCS-TM-264, Argonne National Laboratory, 2003.

[McC08] W. McCune. PROVER9 manual, 2008. Available via `http://www.prover9.org`.

[MN90] U. Martin and T. Nipkow. Ordered rewriting and confluence. In M. E. Stickel, editor, *Proceedings of the 10th International Conference on Automated Deduction*, volume 449 of *LNAI*, pages 366–380. Springer-Verlag, 1990.

[MZ94] A. Middeldorp and H. Zantema. Simple termination revisited. In A. Bundy, editor, *Proceedings of the 12th International Con-*

ference on Automated Deduction, volume 814 of *LNAI*, pages 451–465. Springer-Verlag, 1994.

[New42] M. H. A. Newman. On theories with a combinatorial definition of "equivalence". *Annals of Mathematics*, 43(2):223–243, 1942.

[Nie93] R. Nieuwenhuis. Simple LPO constraint solving methods. *Information Processing Letters*, 47(2):65–69, 1993.

[NO79] G. Nelson and D. C. Oppen. Simplification by cooperating decision procedures. *ACM Transactions on Programming Languages and Systems*, 1(2):245–257, 1979.

[NPW02] T. Nipkow, L. C. Paulson, and M. Wenzel. ISABELLE/*HOL – A Proof Assistant for Higher-Order Logic*, volume 2283 of *LNCS*. Springer-Verlag, 2002.

[NR94] R. Nieuwenhuis and A. Rubio. AC-superposition with constraints: No AC-unifiers needed. In A. Bundy, editor, *Proceedings of the 12th International Conference on Automated Deduction*, volume 814 of *LNAI*, pages 545–559. Springer-Verlag, 1994.

[NR95] R. Nieuwenhuis and A. Rubio. Theorem proving with ordering and equality constrained clauses. *Journal of Symbolic Computation*, 19(4):321–351, 1995.

[NR01] R. Nieuwenhuis and A. Rubio. Paramodulation-based theorem proving. In Robinson and Voronkov [RV01], chapter 7, pages 371–443.

[NW01] A. Nonnengart and Chr. Weidenbach. Computing small clause normal forms. In Robinson and Voronkov [RV01], chapter 6, pages 335–367.

[Opp80] D. Oppen. Complexity, convexity and combination of theories. *Theoretical Computer Science*, 12:291–302, 1980.

[Pau92] E. Paul. A general refutational completeness result for an inference procedure based on associative-commutative unification. *Journal of Symbolic Computation*, 14(6):577–618, 1992.

[Pic03] M. Pichora. *Automated Reasoning About Hardware Data Types Using Bit-vectors of Symbolic Lengths*. PhD thesis, University of Toronto, 2003.

[Plo72] G. Plotkin. Building in equational theories. In B. Meltzer and D. Michie, editors, *Machine Intelligence*, volume 7, pages 73–90. Edinburgh University Press, 1972.

[PS81] G. E. Peterson and M. E. Stickel. Complete sets of reduction for some equational theories. *Journal of the Association for Computing Machinery*, 28:233–264, 1981.

[PWZ96] M. Petkovsek, H. S. Wilf, and D. Zeilberger. $A{=}B$. A. K. Peters, Ltd., 1996. Available online via `http://www.cis.upenn.edu/~wilf/AeqB.html`.

[RS01] H. Rueß and N. Shankar. Deconstructing Shostak. In *Proceedings of the 16th IEEE Symposium on Logic in Computer Science*, pages 19–28. IEEE Computer Society Press, 2001.

[Rub95] A. Rubio. Extension orderings. In Z. Fülöp and F. Gécseg, editors, *Proceedings of the 22nd International Colloquium on Automata, Languages and Programming*, volume 944 of *LNCS*, pages 511–522. Springer-Verlag, 1995.

[Rus08] R. Rusev. Bitvector reasoning with SPASS. Master's thesis, Universität des Saarlandes, Fachrichtung Informatik, 2008. In preparation.

[RV91] M. Rusinowitch and L. Vigneron. Automated deduction with associative commutative operators. In Ph. Jorrand and J. Kelemen, editors, *Proceedings of the International Workshop on Fundamentals of Artificial Intelligence Research*, volume 535 of *LNCS*, pages 185–199. Springer-Verlag, 1991.

[RV01] A. Robinson and A. Voronkov, editors. *Handbook of Automated Reasoning*, volumes I and II. Elsevier, 2001.

[Sch01] S. Schulz. System abstract: E 0.61. In R. Goré, A. Leitsch, and T. Nipkow, editors, *Proceedings of the First International Joint Conference on Automated Reasoning*, volume 2083 of *LNAI*, pages 370–375. Springer-Verlag, 2001.

[Sch07] S. Schulz. Personal communication, Dagstuhl, October 2007.

[SD07] A. Stump and M. Deters. The SMT-COMP web page, 2007. See `http://www.smtcomp.org`.

[Sha02] N. Shankar. Little engines of proof. In L.-H. Eriksson and P. A. Lindsay, editors, *Proceedings of the 11th International Symposium of Formal Methods Europe*, volume 2391 of *LNCS*, pages 1–20. Springer-Verlag, 2002.

[Sho84] R. E. Shostak. Deciding combinations of theories. *Journal of the ACM*, 31(1):1–12, 1984.

[SR02] N. Shankar and H. Rueß. Combining Shostak theories. In S. Tison, editor, *Proceedings of the 13th International Conference on Rewriting Techniques and Applications*, volume 2378 of *LNCS*, pages 1–18. Springer-Verlag, 2002.

[SS98] G. Sutcliffe and C. B. Suttner. The TPTP problem library. CNF release v1.2.1. *Journal of Automated Reasoning*, 21:177–203, 1998.

[SS06] G. Sutcliffe and C. Suttner. The state of CASC. *AI Communications*, 19(1):35–48, 2006. Competition archive available via `http://www.tptp.org/CASC`.

[Sti85] M. E. Stickel. Automated deduction by theory resolution. *Journal of Automated Reasoning*, 1(4):333–355, 1985.

[Stu00] J. Stuber. Deriving theory superposition calculi from convergent term rewriting systems. In L. Bachmair, editor, *Proceedings of the 11th International Conference on Rewriting Techniques and Applications*, volume 1833 of *LNCS*, pages 229–245. Springer-Verlag, 2000.

[TH96] C. Tinelli and M. Harandi. A new correctness proof of the Nelson-Oppen combination procedure. In F. Baader and K. Schulz, editors, *Proceedings of the First International Workshop on Frontiers of Combining Systems*, pages 103–120. Kluwer Academic Publishers, 1996.

[Ver08] The Verisoft Consortium. The Verisoft project, 2008. See `http://www.verisoft.de`.

[Vig94] L. Vigneron. Associative-commutative deduction with constraints. In A. Bundy, editor, *Proceedings of the 12th International Conference on Automated Deduction*, volume 814 of *LNAI*, pages 530–544. Springer-Verlag, 1994.

[Wal01] U. Waldmann. Superposition and chaining for totally ordered divisible Abelian groups. In R. Goré, A. Leitsch, and T. Nipkow, editors, *Proceedings of the First International Joint Conference on Automated Reasoning*, volume 2083 of *LNAI*, pages 226–241. Springer-Verlag, 2001.

[Wal02] U. Waldmann. Cancellative Abelian monoids and related structures in refutational theorem proving (Part I). *Journal of Symbolic Computation*, 33(6):777–829, 2002.

[WB86] F. Winkler and B. Buchberger. A criterion for eliminating unnecessary reductions in the Knuth-Bendix algorithm. *Colloquia Mathematica Societatis János Bolyai*, 42:849–869, 1986.

[Wei01] Chr. Weidenbach. Combining superposition, sorts and splitting. In Robinson and Voronkov [RV01], chapter 27, pages 1965–2012. An extended version is part of the SPASS distribution, which is available from `http://www.spass-prover.org/download`.

[Wer92] U. Wertz. First-order theorem proving modulo equations. Research Report MPI-I-92-216, Max-Planck-Institut für Informatik, 1992.

[Wol02] S. Wolfram. *A New Kind of Science*. Wolfram Media, 2002. Avalaible online via `http://www.wolframscience.com`.

[WSH+07] Chr. Weidenbach, R. Schmidt, Th. Hillenbrand, R. Rusev, and D. Topić. SPASS version 3.0. In F. Pfenning, editor, *Proceedings of the 21st International Conference on Automated Deduction*, volume 4603 of *LNAI*, pages 514–520. Springer-Verlag, 2007.