Journal of Automated Reasoning manuscript No.
DOI 10.1007/s10817-018-09510-y in 64(3):579–609, March 2020

Conflict-Driven Satisfiability for Theory Combination: Transition System and Completeness

Maria Paola Bonacina · Stéphane Graham-Lengrand · Natarajan Shankar

Received: 4 March 2018 / Accepted: 20 December 2018 / Published online: 4 January 2019

Abstract Many applications depend on solving the satisfiability of formulæ involving propositional logic and first-order theories, a problem known as Satisfiability Modulo Theory (SMT). This article presents a new method for satisfiability modulo a combination of theories, named CDSAT, for *Conflict-Driven SATisfiability*. CDSAT also solves *Satisfiability Modulo Assignment* (SMA) problems that may include assignments to first-order terms. A conflict-driven procedure assigns values to variables to build a model, and performs inferences on demand in order to solve conflicts between assignments and formulæ. CDSAT extends this paradigm to *generic* combinations of *disjoint* theories, each characterized by a collection of inference rules called *theory module*. CDSAT coordinates the theory modules in such a way that the conflict-driven reasoning happens *in the union of the theories*, not only in propositional logic. As there is no fixed hierarchy with propositional logic at the center and the other theories as satellites, CDSAT offers a flexible framework for model search. We prove the *soundness*, *completeness*, and *termination* of CDSAT, identifying sufficient requirements on theories and modules that ensure these properties.

Keywords Theory combination · Conflict-driven decision procedures · Model building · Satisfiability modulo assignment

This research was funded in part by NSF Grants 1528153 and CNS-0917375, by DARPA under Agreement Number FA8750-16-C-0043, and by Grant "Ricerca di base 2017" of the Università degli Studi di Verona.

Maria Paola Bonacina
Università degli Studi di Verona, Strada Le Grazie 15, 37134 Verona, Italy, EU. E-mail: mariapaola.bonacina@univr.it

Stéphane Graham-Lengrand
SRI International, 333 Ravenswood Avenue, Menlo Park, CA 94025, USA. E-mail: graham-lengrand@lix.polytechnique.fr

Natarajan Shankar
SRI International, 333 Ravenswood Avenue, Menlo Park, CA 94025, USA. E-mail: shankar@csl.sri.com

1 Introduction

A growing trend in automated reasoning is the generalization of *conflict-driven reasoning* from propositional to first-order logic (see [4]). A motivation for this trend is the remarkable success of solvers for propositional satisfiability (SAT) [19]. While this success depends on several ingredients, a key one is the *conflict-driven clause learning procedure* (CDCL) [20] at the heart of these solvers. CDCL tries to build a model of the input set of clauses by guessing truth assignments to Boolean variables and propagating their consequences through the clauses. When a conflict between clauses and assignments arises, CDCL performs conflict-driven inferences to explain the conflict, learns new clauses, named *lemmas*, and solves the conflict. In conflict-driven reasoning, inferences are applied *lazily* to respond to conflicts between formulæ and candidate model, and this is considered a crucial advantage of this paradigm.

Example 1 Given the set of propositional clauses $S = \{\neg a \vee b,\ \neg c \vee d,\ \neg e \vee \neg f,\ f \vee \neg e \vee \neg b\}$, say that CDCL guesses $a \leftarrow \mathsf{true}$. In order to satisfy $\neg a \vee b$, CDCL propagates $b \leftarrow \mathsf{true}$ with *justification* $\neg a \vee b$. Similarly, CDCL guesses $c \leftarrow \mathsf{true}$, propagates $d \leftarrow \mathsf{true}$ with justification $\neg c \vee d$, guesses $e \leftarrow \mathsf{true}$, and propagates $f \leftarrow \mathsf{false}$ (for $\neg f \leftarrow \mathsf{true}$) with justification $\neg e \vee \neg f$. Assignments are stored in a *trail* $\Gamma = a \leftarrow \mathsf{true},\ b \leftarrow \mathsf{true},\ c \leftarrow \mathsf{true},\ d \leftarrow \mathsf{true},\ e \leftarrow \mathsf{true},\ f \leftarrow \mathsf{false}$. Each decision opens a new *level*: a, b are on level 1; c, d are on level 2; and e, $\neg f$ are on level 3. Now $\neg b \vee \neg e \vee f$ is *in conflict with* Γ, as all its literals are false in Γ. CDCL explains the conflict by resolving the *conflict clause* $\neg b \vee \neg e \vee f$ with the justification $\neg e \vee \neg f$ of $\neg f$, generating resolvent $\neg b \vee \neg e$, still in conflict. The First Unique Implication Point (1UIP) heuristic continues this process until it generates a conflict clause C where only one literal, say L, is false at the current decision level [20]. CDCL learns C, truncates the trail at the smallest decision level where L is unassigned and all other literals of C are false, and adds $L \leftarrow \mathsf{true}$ with justification C to the trail, solving the conflict. As only $\neg e$ in $\neg b \vee \neg e$ is false at level 3, CDCL learns $\neg b \vee \neg e$, truncates the trail at level 1, and adds $e \leftarrow \mathsf{false}$, yielding $\Gamma = a \leftarrow \mathsf{true},\ b \leftarrow \mathsf{true},\ e \leftarrow \mathsf{false}$. Lemma $\neg b \vee \neg e$ tells that S has no model where both b and e are true. Deciding $c \leftarrow \mathsf{false}$ or $d \leftarrow \mathsf{true}$ lets Γ satisfy all clauses and conclude that S is satisfiable.

The quest for methods that reproduce the success of CDCL beyond SAT inspired decision procedures for the satisfiability of sets of literals in fragments of arithmetic, that also work by guessing assignments to variables, propagating consequences, and performing inferences *on demand* to explain and solve conflicts (e.g., [16,17], and see [4] for more references). In contrast to CDCL, these procedures use *assignments to first-order variables*, not only Boolean ones, and explain conflicts by inferences producing lemmas that may contain *new* (i.e., non-input) atoms. We call these procedures *conflict-driven $\mathcal{T}$-satisfiability procedures* where $\mathcal{T}$ is the relevant theory.

Example 2 Consider the set $R = \{-2{\cdot}x - y < 0,\ x + y < 0,\ x < -1\}$ in linear rational arithmetic (LRA). A conflict-driven LRA-satisfiability procedure

tries to build a model by guessing a value for a variable, say $y \leftarrow 0$. Under this assignment, $-2{\cdot}x - y < 0$ yields $x > 0$, which, together with $x < -1$, empties the set of possible values for x. Thus, $y \leftarrow 0$ caused an LRA-*conflict* that the procedure explains by the *new atom* $-y < -2$, linear combination of $-2{\cdot}x - y < 0$ and $x < -1$. The formula $(-2{\cdot}x - y < 0 \wedge x < -1) \supset -y < -2$ is an LRA-*lemma*, which excludes *all* assignments $y \leftarrow \mathfrak{c}$ where $\mathfrak{c} \leq 2$. If the procedure retracts $y \leftarrow 0$ and guesses $y \leftarrow 4$, constraints $-2{\cdot}x - y < 0$ and $x + y < 0$ yield $x > -2$ and $x < -4$. The procedure explains this LRA-conflict by the *new atom* $y < 0$, linear combination of $-2{\cdot}x - y < 0$ and $x + y < 0$. As $y < 0$ is violated by $y \leftarrow 4$, the procedure retracts $y \leftarrow 4$. Then no assignment to y can satisfy both $-y < -2$ and $y < 0$. This LRA-conflict is explained by the *new atom* $0 < -2$ linear combination of $-y < -2$ and $y < 0$. Since $0 < -2$ is a contradiction, the procedure concludes that R is LRA-unsatisfiable.

Most applications query the $\mathcal{T}$-satisfiability of arbitrary quantifier-free formulæ, a problem known as Satisfiability Modulo Theory (SMT). A standard approach to SMT, named DPLL($\mathcal{T}$) [22], augments CDCL with a $\mathcal{T}$-satisfiability procedure, used as a *black-box* to detect that a set of truth assignments on the trail is $\mathcal{T}$-unsatisfiable ($\mathcal{T}$-*conflict*), and determine that a truth assignment follows in $\mathcal{T}$ from truth assignments on the trail ($\mathcal{T}$-*lemma*). The entailed truth assignment is only for an already existing atom, and the introduction of new atoms in $\mathcal{T}$-lemmas is not allowed. In most applications of SMT, $\mathcal{T}$ is a combination of theories $\mathcal{T}_1, \ldots, \mathcal{T}_n$, and the $\mathcal{T}$-satisfiability procedure is obtained from $\mathcal{T}_k$-satisfiability procedures, $1 \leq k \leq n$, by the *equality sharing* method, also known as Nelson-Oppen scheme [21,2,18,7,14].

Example 3 Set $P = \{f(\mathsf{select}(\mathsf{store}(a, i, v), j)) \simeq w, (w + \frac{1}{2})^2 \simeq f(u), i \simeq j, u \simeq v\}$ involves the combination of the quantifier-free fragment of the theory of equality, known as equality with uninterpreted function symbols (EUF), the theory of arrays with extensionality (Arr), and (nonlinear) real arithmetic (RA).

Equality sharing requires the theories $\mathcal{T}_1, \ldots, \mathcal{T}_n$ to be *disjoint*, which means that they do not share symbols other than equality, and *stably infinite*, which says that every satisfiable formula has a model with a countably infinite domain. Thus, sorts can be interpreted as countably infinite domains, ensuring that the theories agree on the cardinalities of shared sorts. Equality sharing combines n $\mathcal{T}_k$-satisfiability procedures ($1 \leq k \leq n$) as *black-boxes* that only exchange disjunctions of equalities between shared first-order variables. In DPLL($\mathcal{T}$) with equality sharing [2,18], from now on DPLL($\mathcal{T}$) for brevity, CDCL is in charge of splitting these disjunctions as well all disjunctions generated as $\mathcal{T}$-lemmas, which may contain non-input atoms. However, if conflict-driven $\mathcal{T}_k$-satisfiability procedures were integrated in DPLL($\mathcal{T}$), they would be treated as black-boxes: their theory-specific conflict-driven mechanisms would remain private, first-order assignments and new atoms would not be exported and could not guide the global search.

The problem of integrating CDCL with a single conflict-driven $\mathcal{T}$-satisfiability procedure was solved by MCSAT, for *Model-Constructing SATisfiability* [9]. MCSAT uses assignments to first-order variables on a par with Boolean

ones, and coordinates the conflict explanation mechanisms at the Boolean and theory levels in a unified manner, with new atoms produced by $\mathcal{T}$-inferences *on demand* to explain $\mathcal{T}$-conflicts. Unlike DPLL($\mathcal{T}$), where the conflict-driven reasoning is only Boolean, MCSAT lifts CDCL to a conflict-driven procedure for *satisfiability modulo a single theory*, such as bit-vectors [23] and nonlinear integer arithmetic [13]. Although an instance of MCSAT for the combination of EUF and LRA was described [15], and a modular implementation for several theories was developed [11], MCSAT as a formal system is not a combination calculus: the conflict-driven combination of a generic range of theories remained an open problem. Solving this problem is important to leverage conflict-driven reasoning to SMT and is challenging because it requires:

- A *new paradigm for theory combination*, where the conflict-driven theory satisfiability procedures cooperate to build a model by sharing assignments, and perform inferences *on demand* to explain conflicts, exporting lemmas that may include *new atoms*;
- A way to accommodate also theory satisfiability procedures that are not conflict-driven and get combined as *black-boxes*, since some theories may have very efficient procedures that are not conflict-driven;
- Sufficient conditions that the combined theories and their satisfiability procedures need to satisfy to ensure *soundness*, *completeness*, and *termination* of the new combination method, and proofs of these properties.

We address all of these issues by introducing CDSAT, for *Conflict-Driven SATisfiability*, the first formal system for *conflict-driven theory combination*.

2 Overview of CDSAT

CDSAT extends *conflict-driven reasoning* to *generic* combinations of disjoint theories $\mathcal{T}_1, \ldots, \mathcal{T}_n$, solving the problem of combining multiple conflict-driven and black-box $\mathcal{T}_k$-satisfiability procedures ($1 \leq k \leq n$) into a conflict-driven $\mathcal{T}$-satisfiability procedure, where $\mathcal{T}$ is the union of the theories. In the CDSAT architecture, both propositional and theory reasoning are integrated in a similar manner. Formulæ are terms of sort prop (for proposition), and propositional logic, also known as the Boolean theory (Bool), is one of $\mathcal{T}_1, \ldots, \mathcal{T}_n$, with CDCL as its procedure. With formulæ considered as terms, *assignments* take center stage. Problems are presented as assignments: for example, a set $\{l_1, \ldots, l_m\}$ of formulæ to satisfy is represented as the assignment $\{l_1 \leftarrow \mathsf{true}, \ldots, l_m \leftarrow \mathsf{true}\}$. Assignments are used to represent partial candidate models and inferences manipulate assignments. First-order assignments assign values to first-order terms, not only first-order variables. Thus, there are two ways to communicate an equality: making it true and assigning the same value to its sides.

CDSAT works with a *trail* where all $\mathcal{T}_k$-procedures post assignments, so that all assignments are public. Every $\mathcal{T}_k$ has its *theory view* of such a mixed assignment: a $\mathcal{T}_k$-procedure may not interpret $\mathcal{T}_i$-assignments, $i \neq k$, but it understands the equalities and inequalities that they imply, thus included in the $\mathcal{T}_k$-view. Every $\mathcal{T}_k$-procedure can ensure that its model construction *en-*

dorses the $\mathcal{T}_k$-view of the trail, where the new notion of *endorsement* of an assignment by a model extends that of satisfaction of a formula by a model.

Thanks to its uniform treatment of Boolean and first-order assignments, CDSAT solves both SMT and SMA problems, where SMA stands for *Satisfiability Modulo Assignment*. In an SMA problem, the input includes assignments of values to first-order terms. SMA problems arise in connection with *optimization* [10], which is even more common than satisfiability in applications. The idea is to approach an optimization problem by solving a series of SMA problems where each initial first-order assignment contains information generated by the previous runs, so that the series converges towards an optimal solution. In such a context, first-order initial assignments play a different role than equational constraints in the input problem, and, since values are not necessarily part of the language of terms and formulæ, an SMA problem cannot be straightforwardly expressed as an SMT problem.

In order to combine conflict-driven procedures, CDSAT factors out their common conflict-driven mechanisms in the form of a transition system, with transition rules for *decisions*, *deductions*, *conflict detection*, and *conflict solving*. Thus, what is left to be combined are n inference systems for $\mathcal{T}_1, \ldots, \mathcal{T}_n$, here called *theory modules*, whose inference rules deduce Boolean assignments from assignments of any kind. Decisions rely on two new notions: the *relevance* of a term for a theory, and the *acceptability* of an assignment for a theory module. Deductions cover both *propagations* and inferences that *explain* theory conflicts. A $\mathcal{T}_k$-satisfiability procedure that is not conflict-driven is handled by CDSAT as an inference system whose only inference rule invokes the procedure to detect the $\mathcal{T}_k$-unsatisfiability of a set of assignments. We believe that the abstraction of viewing combination of theories as combination of inference systems brings simplicity and elegance.

We prove that the CDSAT transition system is *sound*, *terminating*, and *complete*.[1] Soundness means that if CDSAT returns unsatisfiable, the input assignment is unsatisfiable. CDSAT is sound provided that the theory modules are sound, that is, all their inferences preserve endorsement by models. All CDSAT derivations are guaranteed to terminate, assuming the existence of a *finite global basis* that limits new term generation. If each theory module comes with a *finite local basis*, a finite global basis can be built. Completeness means that if CDSAT does not return unsatisfiable, there is a model that endorses the assignment in the final state of the derivation, hence the input assignment. For completeness, it does not suffice that every $\mathcal{T}_k$-module is complete with respect to its theory $\mathcal{T}_k$, because the existence of $\mathcal{T}_k$-models does not imply the existence of a $\mathcal{T}$-model. CDSAT is complete, provided all $\mathcal{T}_k$-modules are also complete relative to a *leading theory*, which has information about all sorts involved in the combination and their cardinalities. If all theories are stably infinite, this requirement is satisfied vacuously.

This article is organized as follows. After basic definitions in Section 3, Section 4 introduces *assignment*, *theory view*, and *endorsement*. Section 5 presents

[1] In fact, it is an *inference system* in the sense of [12].

theory modules with the notions of *relevance* and *acceptability*. Section 6 describes theory modules for Bool, LRA, EUF, and Arr, showing how to cover black-box satisfiability procedures as theory modules. Section 7 gives the CDSAT transition system and illustrate its behavior with examples. Section 8 defines sufficient conditions for *termination* and *completeness*. Section 9 contains the *soundness*, *termination*, and *completeness* theorems with their proofs. Section 10 compares CDSAT with related work, summarizes our results, and outlines directions for future work. A short version of this article appeared [5]. We extended CDSAT with *learning* and *proof generation* [6].

3 Preliminary Definitions

A *signature* Σ is a pair (S, F), where S is a set of *sorts* including sort prop and F is a set of *sorted symbols* with equality symbols $\simeq_s : (s \times s) \to \mathsf{prop}$ for all $s \in S$. For $f \in F$, the notation $f : (s_1 \times \cdots \times s_m) \to s$ says that f has arity m, input sorts $s_1, \ldots, s_m$ $(m \geq 0)$ and output sort s. Symbols can be *constants* $(m = 0)$, *functions*, and *predicates* which have prop as output sort. Equalities are written in infix notation. We may write $\simeq_S$ for $\{\simeq_s : s \times s \to \mathsf{prop} \mid s \in S\}$, and $\simeq$ for $\simeq_s$ when s is clear. The connectives $\wedge$, $\vee$, and $\neg$, if present, are seen as symbols whose input and output sorts are prop. Two signatures are *disjoint* if they do not share symbols other than equality. We use $\mathcal{V} = (\mathcal{V}^s)_{s \in S}$ for a collection of disjoint sets of *variables*, where $\mathcal{V}^s$ is the set of variables of sort s. If S_1 and S_2 are sets of sorts with collections of sets of variables $\mathcal{V}_1$ and $\mathcal{V}_2$, we write $\mathcal{V}_1 \subseteq \mathcal{V}_2$ if for all sorts $s \in S_1$ we have $s \in S_2$ and $\mathcal{V}_1^s \subseteq \mathcal{V}_2^s$.

Given $\Sigma = (S, F)$ and $\mathcal{V} = (\mathcal{V}^s)_{s \in S}$, for all $s \in S$, every $x \in \mathcal{V}^s$ is a $\Sigma[\mathcal{V}]$-*term* of sort s; and for all $f : (s_1 \times \cdots \times s_m) \to s$ in F, $f(t_1, \ldots, t_m)$ is a $\Sigma[\mathcal{V}]$-*term* of sort s, if $t_1, \ldots, t_m$ are $\Sigma[\mathcal{V}]$-terms of sorts $s_1, \ldots, s_m$. We use t and u for $\Sigma[\mathcal{V}]$-terms, and l for $\Sigma[\mathcal{V}]$-*formulæ* that are the $\Sigma[\mathcal{V}]$-terms of sort prop. We write $t \trianglelefteq u$ if t is a *subterm* of u, and $t \triangleleft u$ if $t \trianglelefteq u$ and $t \neq u$. The set of *free variables* occurring in a $\Sigma[\mathcal{V}]$-term t is defined as usual. The standard formulæ of multi-sorted first-order logic can be obtained as the closure of $\Sigma[\mathcal{V}]$-formulæ under quantifiers and Boolean connectives; *sentences*, or Σ-*sentences* if the signature is relevant, are those with no free variables.

A $\Sigma[\mathcal{V}]$-*interpretation* $\mathcal{M}$ interprets each $s \in S$ as a non-empty *domain* $s^{\mathcal{M}}$ with $\mathsf{prop}^{\mathcal{M}} = \{\mathsf{true}, \mathsf{false}\}$, each $v \in \mathcal{V}^s$ as an element $v^{\mathcal{M}}$ in $s^{\mathcal{M}}$, each $f : (s_1 \times \cdots \times s_m) \to s$ in F as a function $f^{\mathcal{M}}$ from $s_1^{\mathcal{M}} \times \cdots \times s_m^{\mathcal{M}}$ to $s^{\mathcal{M}}$, and each $\simeq_s$ as the function $\simeq_s^{\mathcal{M}}$ from $s^{\mathcal{M}} \times s^{\mathcal{M}}$ to $\{\mathsf{true}, \mathsf{false}\}$ that returns true if and only if its arguments are the same element. The interpretation $\mathcal{M}(t)$ of a $\Sigma[\mathcal{V}]$-term t is defined as usual, and so is the interpretation in $\mathcal{M}$ of any formula of multi-sorted first-order logic whose free variables, if any, are in $\mathcal{V}$. A Σ-*structure* is a $\Sigma[\emptyset]$-interpretation, where $\emptyset$ is a collection of empty sets.

A *theory* $\mathcal{T}$ is defined as a signature-axiomatization pair $(\Sigma, \mathcal{A})$, where $\mathcal{A}$ is a set of Σ-sentences, the $\mathcal{T}$-axioms, that define properties of symbols in Σ. Symbols that do not appear in $\mathcal{T}$-axioms are *free* or *uninterpreted*. $\mathcal{T}$ is also defined as the class of Σ-structures that satisfy $\mathcal{A}$, called models of $\mathcal{T}$ or

$\mathcal{T}$-models. A *$\mathcal{T}[\mathcal{V}]$-model* is a $\Sigma[\mathcal{V}]$-interpretation that is a $\mathcal{T}$-model when the interpretation of variables is ignored. Two theories are *disjoint* if their signatures are. Let $\mathcal{T}_1, \ldots, \mathcal{T}_n$ be the disjoint theories to be combined and $\Sigma_1, \ldots, \Sigma_n$ their signatures, where $\forall k,\ 1 \leq k \leq n$, $\Sigma_k = (S_k, F_k)$. We use $\mathcal{T}_\infty$ for the union of $\mathcal{T}_1, \ldots, \mathcal{T}_n$ to keep $\mathcal{T}$ and Σ generic. $\mathcal{T}_\infty$ has signature $\Sigma_\infty = (S_\infty, F_\infty)$, where $S_\infty = \bigcup_{k=1}^n S_k$ and $F_\infty = \bigcup_{k=1}^n F_k$, and axiomatization $\bigcup_{k=1}^n \mathcal{A}_k$, if $\mathcal{A}_k$ is the axiomatization of $\mathcal{T}_k$, $\forall k,\ 1 \leq k \leq n$. Given the collection of sets of variables $\mathcal{V}_\infty = (\mathcal{V}_\infty^s)_{s \in S_\infty}$, the $\mathcal{T}_\infty$-terms are the $\Sigma_\infty[\mathcal{V}_\infty]$-terms, including $\mathcal{T}_\infty$-formulæ that are the $\mathcal{T}_\infty$-terms of sort prop. From now on, *variable* stands for variable in $\mathcal{V}_\infty$, *term* for $\mathcal{T}_\infty$-term, and *formula* for $\mathcal{T}_\infty$-formula.

Example 4 For Example 2, Σ_{LRA} has sorts $S_{\mathsf{LRA}} = \{\mathsf{prop}, \mathsf{Q}\}$ and symbols $F_{\mathsf{LRA}} = \simeq_{S_{\mathsf{LRA}}} \cup \{(1 : \mathsf{Q}), (+ : (\mathsf{Q}\times\mathsf{Q})\to\mathsf{Q}), (\leq, < : (\mathsf{Q}\times\mathsf{Q})\to\mathsf{prop})\} \cup \{(c\cdot : \mathsf{Q}\to\mathsf{Q}) | c \in \mathbb{Q}\}$, where Q and $\mathbb{Q}$ are the sort and the set of rational numbers, respectively, and $\cdot$ is scalar multiplication. For Example 3, Σ_{RA} has sorts $S_{\mathsf{RA}} = \{\mathsf{prop}, \mathsf{R}\}$ and symbols $F_{\mathsf{RA}} = \simeq_{S_{\mathsf{RA}}} \cup \{(c : \mathsf{R}) | c \in \mathbb{Q}\} \cup \{(+, \cdot : (\mathsf{R}\times\mathsf{R})\to\mathsf{R})\}$, where R is the sort of real numbers, $+$ is addition and $\cdot$ is product. Term $(w+\frac{1}{2})^2$ in P abbreviates $(w+\frac{1}{2})\cdot(w+\frac{1}{2})$. Σ_{Arr} has sorts $S_{\mathsf{Arr}} = \{\mathsf{prop}, V, I, (I\Rightarrow V)\}$ and symbols $F_{\mathsf{Arr}} = \simeq_{S_{\mathsf{Arr}}} \cup \{\mathsf{select} : (I\Rightarrow V)\times I\to V, \mathsf{store} : (I\Rightarrow V)\times I\times V\to(I\Rightarrow V)\}$, where V is the sort of array elements, I that of array indices, and $(I\Rightarrow V)$ that of arrays. Σ_{EUF} has sorts $S_{\mathsf{EUF}} = \{\mathsf{prop}, \mathsf{R}, V\}$ and symbols $F_{\mathsf{EUF}} = \simeq_{S_{\mathsf{EUF}}} \cup \{f : V\to\mathsf{R}\}$, where f ranges from V to R by the first and second equalities in P.

A theory $\mathcal{T}_k$ has a partial understanding of a $\mathcal{T}_\infty$-term u: a subterm of u whose root symbol is not in F_k is a *free variable* for $\mathcal{T}_k$. We call such a term *Σ_k-foreign*, or *foreign* if Σ_k is understood. The replacement of foreign terms with *new variables* in equality sharing [21,1,18,7] is subsumed by defining *free Σ-variables* to include Σ-foreign terms [1]. If $\Sigma = (S, F)$ is a signature with $F \subseteq F_\infty$, for all sorts $s \in S$, the set $\mathsf{fv}_\Sigma^s(u)$ of *free Σ-variables* of sort s in term u is the set of all $\lhd$-maximal subterms of u whose root symbol is Σ-foreign.

Example 5 For Example 3, the free Σ_{RA}-variables are $f(\mathsf{select}(\mathsf{store}(a, i, v), j))$, w, $f(u)$, $i \simeq j$, and $u \simeq v$, where $i \simeq j$ and $u \simeq v$ are included because they are Σ_{RA}-foreign, since $\simeq_I \notin F_{\mathsf{RA}}$ and $\simeq_V \notin F_{\mathsf{RA}}$. The free Σ_{EUF}-variables are $\mathsf{select}(\mathsf{store}(a, i, v), j)$, w, $(w+\frac{1}{2})^2$, u, $i \simeq j$, and v, where $i \simeq j$ is included as above, whereas $u \simeq v$ is not as $\simeq_V \in F_{\mathsf{EUF}}$. The free Σ_{Arr}-variables are $f(\mathsf{select}(\mathsf{store}(a, i, v)), j) \simeq w$, $(w+\frac{1}{2})^2 \simeq f(u)$, i, j, u, and v.

Then $\mathsf{fv}_\Sigma(t) = \bigcup_{s\in S} \mathsf{fv}_\Sigma^s(t)$ and $\mathsf{fv}_\Sigma(X) = \{u \mid u \in \mathsf{fv}_\Sigma(t), t \in X\}$ for a set X of terms. We write $\mathsf{fv}(t)$ and $\mathsf{fv}(X)$ when Σ is Σ_∞: as there are no Σ_∞-foreign terms, $\mathsf{fv}(t)$ and $\mathsf{fv}(X)$ contain only variables x from $\mathcal{V}_\infty$. If t is a $\Sigma_\infty[\mathcal{V}_\infty]$-term such that $\mathsf{fv}_\Sigma(t) \subseteq \mathcal{V}$, term t can be seen as a $\Sigma[\mathcal{V}]$-term and a $\Sigma[\mathcal{V}]$-interpretation $\mathcal{M}$ interprets t as $\mathcal{M}(t)$.

4 Assignments and Models

CDSAT solves $\mathcal{T}_\infty$-satisfiability problems presented as *assignments* of values to terms, and uses assignments to represent candidate partial models of the input

problem and reason about them. Assignments, including input assignments, may contain terms of any sort, and assignable values are not necessarily in the theories' signatures (e.g., consider $x \leftarrow \sqrt{2}$ for the sort R of real numbers). In order to identify the language of assignable values, we enrich the theories' signatures with *new constant symbols* to name whichever values may be necessary to assign in order to solve the problem. Let $\uplus$ denote disjoint union.

Definition 1 (Theory extension) Given a theory $\mathcal{T} = (\Sigma, \mathcal{A})$ with signature $\Sigma = (S, F)$, a theory $\mathcal{T}^+ = (\Sigma^+, \mathcal{A}^+)$ is an *extension* of $\mathcal{T}$ if $\Sigma^+ = (S, F^+)$, $F^+ = F \uplus D$, D is a set of sorted constant symbols called $\mathcal{T}$*-values*, $\mathcal{A}^+ = \mathcal{A} \uplus \mathcal{C}$, and $\mathcal{C}$ is a set of Σ^+-sentences.

For all $s \in S$, let D^s be the set of $\mathcal{T}$-values of sort s. If $D^s \neq \emptyset$, sort s is called $\mathcal{T}$*-public*, as there are $\mathcal{T}$-values that can be assigned to terms of sort s in a way visible to all theories. Since true and false should be assignable values, prop is required to be a $\mathcal{T}$-public sort with $D^{\mathsf{prop}} = \{\mathsf{true}, \mathsf{false}\}$ and $\{\mathsf{true}, \neg\mathsf{false}\} \subseteq \mathcal{C}$. The *trivial extension* of any theory adds only true and false, so that $D = \{\mathsf{true}, \mathsf{false}\}$ and $\mathcal{C} = \{\mathsf{true}, \neg\mathsf{false}\}$. We use $\mathfrak{b}$ for true or false, $\mathfrak{c}$ and $\mathfrak{d}$ for generic values of arbitrary sorts.

Example 6 Let RA^+ be the extension of theory RA that adds a new constant for every algebraic real number. Sort R is RA-public and the RA-values of sort R are the algebraic reals. The axioms of RA^+ are the formulæ that hold in the standard model of the reals interpreting every RA-value as itself.

Extending the signature with names to denote all individuals in a $\mathcal{T}$-model's domain is a standard move in automated reasoning. In such cases a $\mathcal{T}$-value is both the domain element and the constant symbol that names it. We do this for Boolean values and theories with an "intended model" such as the rational or real numbers. For other theories, such as EUF, the trivial extension suffices (see Section 6 for another approach to EUF). For example, theories RA and EUF share sort R in problem P: with RA^+ as in Example 6 and EUF^+ trivial, sort R is RA-public, but not EUF-public.

Definition 2 (Conservativity) Extension $\mathcal{T}^+$ of theory $\mathcal{T}$ with signature Σ is *conservative* if every $\mathcal{T}^+$-unsatisfiable set of Σ-formulæ is $\mathcal{T}$-unsatisfiable.

A conservative extension does not change a problem in the original signature: if CDSAT discovers $\mathcal{T}^+$-unsatisfiability, the problem is $\mathcal{T}$-unsatisfiable; if the problem is $\mathcal{T}$-satisfiable, there is a $\mathcal{T}^+$-model that CDSAT can build. From now on, we assume that each theory $\mathcal{T}_k$, $1 \leq k \leq n$, has a conservative extension $\mathcal{T}_k^+$ with signature $\Sigma_k^+ = (S_k, F_k^+)$ and (possibly empty) sets D_k^s of $\mathcal{T}_k$-values of sort s for all $s \in S_k$. We assume that the extended theories are still disjoint except for true and false. The union of $\mathcal{T}_1^+, \ldots, \mathcal{T}_n^+$ is an extension $\mathcal{T}_\infty^+$ of $\mathcal{T}_\infty$ with signature $\Sigma_\infty^+ = (S_\infty, F_\infty^+)$, where $F_\infty^+ = \bigcup_{k=1}^n F_k^+$, and axiomatization $\bigcup_{k=1}^n \mathcal{A}_k^+$, if $\mathcal{A}_k^+$ is the axiomatization of $\mathcal{T}_k^+$, for all k, $1 \leq k \leq n$.

In the rest of this section, $\mathcal{T}$, $\mathcal{T}^+$, S, Σ and Σ^+ stand for either $\mathcal{T}_k$, $\mathcal{T}_k^+$, S_k, Σ_k and Σ_k^+, for any k, $1 \leq k \leq n$, or $\mathcal{T}_\infty$, $\mathcal{T}_\infty^+$, S_∞, Σ_∞ and Σ_∞^+, so that the definitions apply to both $\mathcal{T}_\infty$ and $\mathcal{T}_k$ for any k, $1 \leq k \leq n$.

Definition 3 (Assignment) A set $J = \{u_1 \leftarrow \mathfrak{c}_1, \ldots, u_m \leftarrow \mathfrak{c}_m\}$ is a $\mathcal{T}$*-assignment* if, $\forall i$, $1 \leq i \leq m$, u_i is a $\mathcal{T}_\infty$-term and $\mathfrak{c}_i$ a $\mathcal{T}$-value of the same sort.

What qualifies J as a $\mathcal{T}$-assignment is that all values are $\mathcal{T}$-values, whereas the terms are $\mathcal{T}_\infty$-terms. For instance, if RA as in Example 6 is combined with a theory featuring a symbol $g : \mathsf{R} \rightarrow \mathsf{R}$ and sharing sort R with RA, $\{x \leftarrow \sqrt{2},\ g(0) \leftarrow \sqrt{2}\}$ is an RA-assignment, even if g is not an RA-symbol. A $\mathcal{T}_\infty$-assignment assigns $\mathcal{T}_\infty$-values to $\mathcal{T}_\infty$-terms. A $\mathcal{T}_k$-assignment is the special case of $\mathcal{T}_\infty$-assignment that assigns $\mathcal{T}_k$-values to $\mathcal{T}_\infty$-terms. We use J for generic $\mathcal{T}$-assignments and reserve H or E for $\mathcal{T}_\infty$-assignments.

A $\mathcal{T}$-assignment is *plausible* if for no formula l it contains both $l \leftarrow \mathsf{true}$ and $l \leftarrow \mathsf{false}$. The set of terms that *occur* in $J = \{u_1 \leftarrow \mathfrak{c}_1, \ldots, u_m \leftarrow \mathfrak{c}_m\}$ is $G(J) = \{t \mid t \trianglelefteq u_i, 1 \leq i \leq m\}$. A singleton $\mathcal{T}$-assignment $\{t \leftarrow \mathfrak{c}\}$ can be written $t \leftarrow \mathfrak{c}$. A $\mathcal{T}$-assignment where all values are Boolean is a *Boolean assignment*. We use A for generic singleton $\mathcal{T}$-assignments, reserving L for Boolean ones. The *flip* of L, written $\overline{L}$, assigns the opposite Boolean value to the same formula. We may abbreviate $l \leftarrow \mathsf{true}$ to l, and then $l \leftarrow \mathsf{false}$ is abbreviated to $\bar{l}$, while $(t \simeq_s u) \leftarrow \mathsf{false}$ can also be abbreviated to $t \not\simeq_s u$. A $\mathcal{T}$-assignment where no value is Boolean is a *first-order* $\mathcal{T}$*-assignment*. Input assignments are assumed to be plausible: an SMT problem is a plausible Boolean assignment (e.g., Examples 2 and 3), and an SMA problem is a plausible assignment with both Boolean and first-order assignments. For instance, $\{x \leftarrow \sqrt{2},\ g(0) \leftarrow \sqrt{2},\ 1 \cdot g(0) \simeq g(0),\ x \cdot y \simeq 1+1\}$ is an RA-assignment and an SMA problem.

In theory combination, assignments are $\mathcal{T}_\infty$-assignments, that mix values from different theories. A theory $\mathcal{T}_k$ has its own *view* of a $\mathcal{T}_\infty$-assignment H, determined by what $\mathcal{T}_k$ understands of H. The $\mathcal{T}_k$*-view* of H includes the $\mathcal{T}_k$-assignments in H, as well as equalities and inequalities between terms of a $\mathcal{T}_k$-sort that are entailed by first-order assignments in H.

Definition 4 (Theory view) The $\mathcal{T}$*-view* of a $\mathcal{T}_\infty$-assignment H is the $\mathcal{T}$-assignment $H_\mathcal{T}$ given by the union of the following sets:
- $\{\ u \leftarrow \mathfrak{c} \mid\ u \leftarrow \mathfrak{c}$ is a $\mathcal{T}$-assignment in $H\}$
- $\bigcup_{k=1}^{n} \{\ u_1 \simeq_s u_2 \mid u_1 \leftarrow \mathfrak{c}, u_2 \leftarrow \mathfrak{c}$ are $\mathcal{T}_k$-assignments in H of sort $s\}$
- $\bigcup_{k=1}^{n} \{\ u_1 \not\simeq_s u_2 \mid u_1 \leftarrow \mathfrak{c}_1, u_2 \leftarrow \mathfrak{c}_2$ are $\mathcal{T}_k$-assignments in H of sort s, $\mathfrak{c}_1 \neq \mathfrak{c}_2\}$

for all $s \in S \backslash \{\mathsf{prop}\}$, where S is the set of sorts of $\mathcal{T}$.

If $\mathcal{T}$ is $\mathcal{T}_k$, we have a $\mathcal{T}_k$-view; if it is $\mathcal{T}_\infty$, we have a $\mathcal{T}_\infty$-view or *global view*. The first-order assignments that put an equality or inequality of sort s in $H_{\mathcal{T}_k}$ are not necessarily $\mathcal{T}_k$-assignments: they can be $\mathcal{T}_i$-assignments for any i, $1 \leq i \neq k \leq n$, if $\mathcal{T}_k$ and $\mathcal{T}_i$ share sort s. If H is Boolean, any $H_\mathcal{T}$ is identical to H, as Boolean assignments are understood by all theories and entail no equalities or inequalities. If H is not Boolean, $H_\mathcal{T}$ and H differ in general. Even the $\mathcal{T}_k$-view $J_{\mathcal{T}_k}$ of a $\mathcal{T}_k$-assignment J can be a strict superset of J.

Example 7 If J is the LRA-assignment $\{y \leftarrow -1,\ z \leftarrow 2\}$, its LRA-view is $J \cup \{y \neq z\}$, and its EUF-view is $\{y \neq z\}$. If H is the assignment $J \cup \{x >$

1, $store(a,i,v) \simeq b$, $select(a,j){\leftarrow}\mathsf{red}\}$, where red is an Arr-value of sort V, its Bool-view is $\{x>1,\ store(a,i,v) \simeq b\}$; its Arr-view is the Bool-view plus $select(a,j){\leftarrow}\mathsf{red}$; its LRA-view is the Bool-view plus J and $\{y \neq z\}$; and its global view is $H \cup \{y \neq z\}$.

We define next a relation between assignments and models that we call *endorsement*. Endorsement is defined for $\mathcal{T}^+$-models to interpret $\mathcal{T}$-values (e.g., $\sqrt{2}$) consistently with the $\mathcal{T}^+$-axioms (e.g., $\sqrt{2}{\cdot}\sqrt{2} \simeq 2$ holds in RA^+ as in Example 6). In order to define endorsement, we extend the notion of free Σ-variables to assignments with $\mathsf{fv}_\Sigma(J) = \{u \mid u \in \mathsf{fv}_\Sigma(t), (t{\leftarrow}\mathfrak{c}) \in J\}$, and we write $\mathsf{fv}(J)$ when Σ is Σ_∞. Then, a model endorses an assignment if the model interprets the two sides of every pair in the assignment as the same element.

Definition 5 (Endorsement) A $\mathcal{T}^+[\mathcal{V}]$-model $\mathcal{M}$ *endorses* a $\mathcal{T}$-assignment J, written $\mathcal{M} \models J$, if for all $(u{\leftarrow}\mathfrak{c}) \in J$, $\mathcal{M}(u) = \mathfrak{c}^{\mathcal{M}}$, assuming $\mathsf{fv}_\Sigma(J) \subseteq \mathcal{V}$.

If J is Boolean, a $\mathcal{T}$-model suffices, and endorsement means that the model interprets the formulæ in J with the truth values given in J. Endorsement and theory view work together as follows. If J is a $\mathcal{T}_k$-assignment and $\mathcal{M}$ is a $\mathcal{T}_k^+[\mathcal{V}]$-model, $\mathcal{M} \models J_{\mathcal{T}_k}$ means that (i) $\mathcal{M} \models J$, (ii) $\mathcal{M} \models u_1 \simeq u_2$ for all $u_1{\leftarrow}\mathfrak{c}$ and $u_2{\leftarrow}\mathfrak{c}$ in J, and (iii) $\mathcal{M} \models u_1 \not\simeq u_2$ for all $u_1{\leftarrow}\mathfrak{c}_1$ and $u_2{\leftarrow}\mathfrak{c}_2$ in J with $\mathfrak{c}_1 \neq \mathfrak{c}_2$. Note that (ii) follows from (i) by definition of endorsement, whereas (iii) does not. It follows that $\mathcal{M} \models J_{\mathcal{T}_k}$ is in general stronger than $\mathcal{M} \models J$, as a model that only endorses J may interpret $\mathfrak{c}_1$ and $\mathfrak{c}_2$ as the same element. Indeed, the new constants added by a theory extension do not necessarily name distinct domain elements. If they did, an extension would impact cardinality, as a $\mathcal{T}^+$-model would have to interpret a $\mathcal{T}$-public sort s with a domain of cardinality at least $|D^s|$. The endorsement $\mathcal{M} \models J_{\mathcal{T}_k}$ only requires $\mathcal{M}$ to distinguish the $\mathcal{T}_k$-values appearing in J. A $\mathcal{T}_k$-assignment J is *satisfiable*, if there is a model $\mathcal{M}$ such that $\mathcal{M} \models J_{\mathcal{T}_k}$, and *unsatisfiable* otherwise.

If $\mathcal{M}$ is a $\mathcal{T}_\infty^+[\mathcal{V}]$-model such that $\mathcal{M} \models H_{\mathcal{T}_\infty}$, we say that $\mathcal{M}$ *globally endorses* H, written $\mathcal{M} \models^G H$. Unlike the one theory case, global endorsement does not imply that $\mathcal{M}$ distinguishes distinct $\mathcal{T}_\infty$-values, as $\mathcal{M}$ may identify distinct $\mathcal{T}_\infty$-values coming from extensions of different theories. For example, let H contain $\{t{\leftarrow}3.1, u{\leftarrow}5.4, t{\leftarrow}\mathsf{red}, u{\leftarrow}\mathsf{blue}\}$, where the first two pairs are $\mathcal{T}_i$-assignments, the last two are $\mathcal{T}_j$-assignments, $i{\neq}j$, and the sort of t and u is both $\mathcal{T}_i$-public and $\mathcal{T}_j$-public. A $\mathcal{T}_\infty^+$-model $\mathcal{M}$ that identifies 3.1 with red and 5.4 with blue can globally endorse H. Such a model can be obtained by combining a $\mathcal{T}_i^+$-model endorsing $H_{\mathcal{T}_i}$ and a $\mathcal{T}_j^+$-model endorsing $H_{\mathcal{T}_j}$. A $\mathcal{T}_\infty$-assignment is *satisfiable*, if there is a model that globally endorses it, and *unsatisfiable* otherwise. From now on *assignment* stands for $\mathcal{T}_\infty$-assignment.

5 Theory Modules

CDSAT models theory reasoning via the notion of *theory module*, an abstraction of *theory satisfiability procedure*, *theory solver*, or *theory plugin*. Let theory

$$
\begin{aligned}
t_1{\leftarrow}\mathfrak{c}, t_2{\leftarrow}\mathfrak{c} &\vdash t_1 \simeq_s t_2 \text{ if } \mathfrak{c} \text{ is a } \mathcal{T}\text{-value of sort } s\\
t_1{\leftarrow}\mathfrak{c}_1, t_2{\leftarrow}\mathfrak{c}_2 &\vdash t_1 \not\simeq_s t_2 \text{ if } \mathfrak{c}_1 \text{ and } \mathfrak{c}_2 \text{ are distinct } \mathcal{T}\text{-values of sort } s\\
&\vdash t_1 \simeq_s t_1 \text{ (reflexivity)}\\
t_1 \simeq_s t_2 &\vdash t_2 \simeq_s t_1 \text{ (symmetry)}\\
t_1 \simeq_s t_2, t_2 \simeq_s t_3 &\vdash t_1 \simeq_s t_3 \text{ (transitivity)}
\end{aligned}
$$

where t_1, t_2, and t_3 are terms of sort s.

Fig. 1 Equality inference rules

$\mathcal{T}$ with signature Σ be one of $\mathcal{T}_1, \ldots, \mathcal{T}_n$, and let $\mathcal{T}^+$ be its extension. A *theory module* for $\mathcal{T}$, or $\mathcal{T}$*-module*, is an *inference system* $\mathcal{I}$ that manipulates $\mathcal{T}$-assignments. Its inferences, called $\mathcal{I}$*-inferences*, are of the form $J \vdash_{\mathcal{I}} L$, where a singleton Boolean assignment L is derived from a $\mathcal{T}$-assignment J. It is not necessary to have a first-order assignment $u{\leftarrow}\mathfrak{c}$ as conclusion, since an inference can have $u \simeq \mathfrak{c}$ as conclusion. Since all theories feature equality, all theory modules include the *equality inferences* given by the rules in Figure 1. For example, $\{x{\leftarrow}\sqrt{2},\ g(0){\leftarrow}\sqrt{2}\} \vdash_{\mathsf{RA}} x{\cdot}g(0) \simeq 1{+}1$, $\{x{\leftarrow}\sqrt{2},\ g(0){\leftarrow}\sqrt{2}\} \vdash_{\mathsf{RA}} g(0) \simeq x$, and $\{x{\leftarrow}\sqrt{3},\ g(0){\leftarrow}\sqrt{2}\} \vdash_{\mathsf{RA}} g(0) \not\simeq x$ are RA-inferences.

An $\mathcal{I}$-inference $J \vdash_{\mathcal{I}} L$ is *sound*, if all $\mathcal{T}^+[\mathcal{V}]$-models that endorse $J_{\mathcal{T}}$ endorse L, assuming $\mathsf{fv}_{\Sigma}(J{\cup}\{L\}) \subseteq \mathcal{V}$. If J is Boolean, the soundness of $J \vdash_{\mathcal{I}} L$ implies that all $\mathcal{T}[\mathcal{V}]$-models endorsing J endorse L, as $\mathcal{T}^+$ is conservative. Indeed, if there were a $\mathcal{T}[\mathcal{V}]$-model $\mathcal{M}$ such that $\mathcal{M} \models J$ and $\mathcal{M} \not\models L$, hence $\mathcal{M} \models \overline{L}$, by conservativity there would be a $\mathcal{T}^+[\mathcal{V}]$-model $\mathcal{M}_1$ such that $\mathcal{M}_1 \models J$ and $\mathcal{M}_1 \models \overline{L}$, hence $\mathcal{M}_1 \not\models L$, violating soundness. A theory module is *sound* if all of its inferences are.

CDSAT works with a $\mathcal{T}_\infty$-assignment H, of which each theory module $\mathcal{I}_k$ understands the $\mathcal{T}_k$-view $H_{\mathcal{T}_k}$, for $1 \le k \le n$. In the rest of this section, $\mathcal{T}$, S, and $\mathcal{I}$ stand for $\mathcal{T}_k$, S_k, and $\mathcal{I}_k$, for any k, $1 \le k \le n$; and definitions are given for a generic $\mathcal{T}$-assignment J so that they apply to any $\mathcal{T}_k$-assignment and therefore any $H_{\mathcal{T}_k}$. A basic CDSAT operation consists of adding to H a singleton $\mathcal{T}$-assignment $u{\leftarrow}\mathfrak{c}$. The corresponding $\mathcal{T}$-module $\mathcal{I}$ determines which pairs $u{\leftarrow}\mathfrak{c}$ are *acceptable* for addition, based on its view $H_{\mathcal{T}}$. The first condition for $u{\leftarrow}\mathfrak{c}$ to be acceptable is that u is *relevant* to $\mathcal{T}$. The simpler way to be relevant is if u occurs in J and its sort is $\mathcal{T}$-public, which means that there are $\mathcal{T}$-values that can be assigned to u. However, an equality $u_1 \simeq_s u_2$ that does not necessarily occur in J is still relevant to $\mathcal{T}$, if its sides occur in J but are not relevant to $\mathcal{T}$ because sort s is not $\mathcal{T}$-public.

Definition 6 (Relevance) A term u is *relevant* to theory $\mathcal{T}$ in a $\mathcal{T}$-assignment J, if (i) either u has a $\mathcal{T}$-public sort and $u \in G(J)$, (ii) or u is an equality $u_1 \simeq_s u_2$ of sort $s \in S$ such that $u_1, u_2 \in G(J)$ and sort s is not $\mathcal{T}$-public.

In equality sharing the $\mathcal{T}$-satisfiability procedures communicate by exchanging equalities between shared variables. In CDSAT this mechanism is generalized to equalities between terms, and the presence of first-order assignments implies that an equality $u_1 \simeq_s u_2$ can be communicated with either the assignment $(u_1 \simeq_s u_2){\leftarrow}\mathsf{true}$ or an assignment $\{u_1{\leftarrow}\mathfrak{c}, u_2{\leftarrow}\mathfrak{c}\}$. Given two theories sharing sort s, relevance determines which theory uses which way: if u_1

and u_2 are relevant by Condition (i) the theory uses $\{u_1 \leftarrow \mathfrak{c}, u_2 \leftarrow \mathfrak{c}\}$; if $u_1 \simeq_s u_2$ is relevant by Condition (ii) the theory uses $(u_1 \simeq_s u_2) \leftarrow \mathsf{true}$.

Example 8 Consider the assignment $H = \{x \leftarrow 5, f(x) \leftarrow 2, f(y) \leftarrow 3\}$ with x, y, $f(x)$ and $f(y)$ of sort Q, shared by LRA and EUF. H_{EUF} contains $x \not\simeq f(x)$, $x \not\simeq f(y)$, and $f(x) \not\simeq f(y)$, while $H_{\mathsf{LRA}} = H \cup H_{\mathsf{EUF}}$. Terms x and y are relevant to LRA by Condition (i) of Definition 6, as they occur in H_{LRA} and Q is LRA-public, but not relevant to EUF, as Q is not EUF-public. Term $x \simeq_{\mathsf{Q}} y$ is relevant to EUF by Condition (ii), as x and y occur in H_{EUF} and Q is not EUF-public, but not relevant to LRA: Condition (i) does not apply, as $x \simeq_{\mathsf{Q}} y$ does not occur in H_{LRA}, and Condition (ii) does not apply, as Q is LRA-public. Each theory has a way to fix and communicate equalities between terms of a known sort such as x and y: EUF can assign a truth value to $x \simeq_{\mathsf{Q}} y$ and LRA can assign values, either the same or different, to x and y.

Definition 7 (Acceptability) A singleton $\mathcal{T}$-assignment $u \leftarrow \mathfrak{c}$ is *acceptable* for $\mathcal{I}$ in a $\mathcal{T}$-assignment J, if (i) u is relevant to $\mathcal{T}$ in J, (ii) J does not assign a $\mathcal{T}$-value to u, and (iii) if $u \leftarrow \mathfrak{c}$ is first-order, there are no $\mathcal{I}$-inferences $J' \cup \{u \leftarrow \mathfrak{c}\} \vdash_{\mathcal{I}} L$ for $J' \subseteq J$ and $\overline{L} \in J$.

Condition (ii) prevents adding $u \leftarrow \mathfrak{c}$ to an assignment that already contains it; and prevents adding $u \leftarrow \mathfrak{c}_1$ to an assignment that already contains $u \leftarrow \mathfrak{c}_2$, for $\mathfrak{c}_1$ and $\mathfrak{c}_2$ distinct $\mathcal{T}_k$-values for the same $\mathcal{T}_k$, which preserves plausibility if $\mathfrak{c}_1$ and $\mathfrak{c}_2$ are true and false. Condition (iii) ensures that the addition of a first-order assignment, which does not have a flip, does not trigger an inference causing a contradiction.

6 A Suite of Specific Theory Modules

Let $\bot$ stand for the assignment $(x \simeq_{\mathsf{prop}} x) \leftarrow \mathsf{false}$, where x is an arbitrary variable. For brevity we omit equality symbols from signatures and equality inference rules from modules. For *propositional logic* (Bool), Σ_{Bool} has sort prop and symbols $\neg : \mathsf{prop} \rightarrow \mathsf{prop}$, and $\vee, \wedge : (\mathsf{prop} \times \mathsf{prop}) \rightarrow \mathsf{prop}$, and Bool^+ is the trivial extension. $\mathcal{I}_{\mathsf{Bool}}$ features an *evaluation* rule that derives the truth value $\mathfrak{b}$ of formula l, given truth values $\mathfrak{b}_1, \ldots, \mathfrak{b}_m$ of subformulæ $l_1, \ldots, l_m$, provided l is in the closure of $l_1, \ldots, l_m$ with respect to the Σ_{Bool}-connectives:

$$l_1 \leftarrow \mathfrak{b}_1, \ldots, l_m \leftarrow \mathfrak{b}_m \vdash_{\mathsf{Bool}} l \leftarrow \mathfrak{b}.$$

Then, $\mathcal{I}_{\mathsf{Bool}}$ includes two rules for negation, two rules for conjunction elimination, and two rules for unit propagation as in CDCL:

$$\begin{array}{lll} \neg l \vdash_{\mathsf{Bool}} \overline{l} & \overline{l_1 \vee \cdots \vee l_m} \vdash_{\mathsf{Bool}} \overline{l_i} & l_1 \vee \cdots \vee l_m, \{\overline{l_j} \mid j \neq i\} \vdash_{\mathsf{Bool}} l_i \\ \overline{\neg l} \vdash_{\mathsf{Bool}} l & l_1 \wedge \cdots \wedge l_m \vdash_{\mathsf{Bool}} l_i & \overline{l_1 \wedge \cdots \wedge l_m}, \{l_j \mid j \neq i\} \vdash_{\mathsf{Bool}} \overline{l_i} \end{array}$$

where $1 \leq j, i \leq m$. Although the evaluation rule alone is sufficient for completeness, the other rules are obviously desirable.

For *linear rational arithmetic* (LRA) with Σ_{LRA} as in Example 4, and LRA^+ adding constant $\tilde{q}$ and axiom $\tilde{q} \simeq_{\mathsf{Q}} q \cdot 1$ for each $q \in \mathbb{Q}$, $\mathcal{I}_{\mathsf{LRA}}$ features an *evaluation* rule that derives the value $\mathfrak{b}$ of a formula l from the values $\tilde{q_1}, \ldots, \tilde{q_m}$ of

its subterms $t_1, \ldots, t_m$ of sort Q:

$$t_1 \leftarrow \tilde{q_1}, \ldots, t_m \leftarrow \tilde{q_m} \vdash_{\mathsf{LRA}} l \leftarrow \mathfrak{b}$$

provided l is a formula whose reduced form is in the closure of $t_1, \ldots, t_m$ with respect to the symbols of F_{LRA}. For example, $w+2 \simeq_{\mathsf{Q}} w$ can be reduced to $2 \simeq_{\mathsf{Q}} 0$ and evaluates to false. Let t_1 and t_2 be terms of sort Q. The *positivization* rules handle the flip of a disequality, while *equality elimination* replaces an equality by disequalities:

$$\overline{t_1 < t_2} \vdash_{\mathsf{LRA}} t_2 \leq t_1, \qquad \overline{t_1 \leq t_2} \vdash_{\mathsf{LRA}} t_2 < t_1, \qquad t_1 \simeq_{\mathsf{Q}} t_2 \vdash_{\mathsf{LRA}} t_1 \leq t_2, t_2 \leq t_1.$$

Let t_0 also be a term of sort Q and x a free Σ_{LRA}-variable of sort Q that is not free in t_0, t_1, and t_2. *Disequality elimination* detects a contradiction, while *Fourier-Motzkin (FM) resolution* eliminates a variable:

$$t_1 \leq x, x \leq t_2, t_1 \simeq_{\mathsf{Q}} t_0, t_2 \simeq_{\mathsf{Q}} t_0, x \not\simeq_{\mathsf{Q}} t_0 \vdash_{\mathsf{LRA}} \bot,$$
$$t_1 \lessdot_1 x, x \lessdot_2 t_2 \vdash_{\mathsf{LRA}} t_1 \lessdot_3 t_2,$$

where $\lessdot_1, \lessdot_2, \lessdot_3 \in \{<, \leq\}$ and $\lessdot_3$ is $<$ if and only if either $\lessdot_1$ or $\lessdot_2$ is $<$. Each premise l in the last two rules stands for any formula reducible to l: FM-resolution applies to $2{\cdot}y{-}x < y$ and $2{\cdot}x < 3$ as they reduce to $y < x$ and $x < \frac{3}{2}$ yielding $y < \frac{3}{2}$. A linear combination $e_1{+}z{<}c_1,\ e_2{-}z{<}c_2 \vdash\ e_1{+}e_2{<}c_1{+}c_2$ is captured by the FM-resolution step $e_2{-}c_2{<}z,\ z{<}c_1{-}e_1 \vdash_{\mathsf{LRA}} e_2{-}c_2{<}c_1{-}e_1$.

For the *theory of equality* (EUF) with $\Sigma_{\mathsf{EUF}} = (S, \simeq_S \cup F)$, $\mathcal{I}_{\mathsf{EUF}}$ has rules

$$(t_i \simeq u_i)_{i=1\ldots m}, f(t_1, \ldots, t_m) \not\simeq f(u_1, \ldots, u_m) \vdash_{\mathsf{EUF}} \bot$$
$$(t_i \simeq u_i)_{i=1\ldots m} \vdash_{\mathsf{EUF}} f(t_1, \ldots, t_m) \simeq f(u_1, \ldots, u_m)$$
$$(t_i \simeq u_i)_{i=1\ldots m, i\neq j}, f(t_1, \ldots, t_m) \not\simeq f(u_1, \ldots, u_m) \vdash_{\mathsf{EUF}} t_j \not\simeq u_j$$

for all $f \in F$. The first rule is sufficient for completeness, as it captures a lazy approach that does not propagate anything until equalities between existing terms contradict a congruence axiom [15]. Since $\mathcal{I}_{\mathsf{EUF}}$ does not use first-order assignments, no sort needs to be EUF-public, and the only assignments give truth values to equalities. Alternatively, one may make the sorts in S EUF-public, with a countably infinite set of EUF-values for each sort and no axioms. Equality inferences employ assignments of EUF-values to determine whether terms are equal, using EUF-values as identifiers of congruence classes of terms. Assume that $\mathfrak{c}_1$, $\mathfrak{c}_2$, and $\mathfrak{c}_3$ are distinct EUF-values: the assignment $\{x \leftarrow \mathfrak{c}_1, y \leftarrow \mathfrak{c}_1, f(x) \leftarrow \mathfrak{c}_2\}$ places x and y in congruence class $\mathfrak{c}_1$, and $f(x)$ in class $\mathfrak{c}_2$; if $f(y) \leftarrow \mathfrak{c}_3$ is added, two equality inferences and the first rule of $\mathcal{I}_{\mathsf{EUF}}$ expose a conflict in the above-mentioned lazy style.

For the theory of *arrays with extensionality* (Arr) (e.g., [7,8]), given sorts I for indices and V for elements, the array sort constructor builds the sort $I{\Rightarrow}V$ of arrays with indices in I and elements in V. For $\Sigma_{\mathsf{Arr}} = (S, F)$, S is the free closure of a set of basic sorts with respect to the array sort constructor, and F contains $\mathsf{select}_{I\Rightarrow V} : (I{\Rightarrow}V){\times}I{\to}V$, $\mathsf{store}_{I\Rightarrow V} : (I{\Rightarrow}V){\times}I{\times}V{\to}(I{\Rightarrow}V)$, and $\mathsf{diff}_{I\Rightarrow V} : (I{\Rightarrow}V){\times}(I{\Rightarrow}V){\to}I$ for all $(I{\Rightarrow}V) \in S$, where $\mathsf{diff}_{I\Rightarrow V}$ is the Skolem function symbol arising from turning the extensionality axiom for $I{\Rightarrow}V$ into a clause. We write $\mathsf{select}(a, i)$ as $a[i]$ and $\mathsf{store}(a, i, v)$ as $a[i]{:=}v$. Let a, b, c, and d be variables of any $I{\Rightarrow}V$ sort, u and v variables of sort V, and i, j, and k variables of sort I. $\mathcal{I}_{\mathsf{Arr}}$ has rules capturing *congruence axioms*, *read-over-write*

axioms, and the *extensionality axiom*:

$$\begin{aligned}
a \simeq b,\ i \simeq j,\ a[i] \not\simeq b[j] \ &\vdash_{\mathsf{Arr}} \bot \\
a \simeq b,\ i \simeq j,\ u \simeq v,\ (a[i]{:=}u) \not\simeq (b[j]{:=}v) \ &\vdash_{\mathsf{Arr}} \bot \\
a \simeq c,\ b \simeq d,\ \mathsf{diff}(a,b) \not\simeq \mathsf{diff}(c,d) \ &\vdash_{\mathsf{Arr}} \bot \\
b \simeq (a[i]{:=}u),\ i \simeq j,\ b[j] \not\simeq u \ &\vdash_{\mathsf{Arr}} \bot \\
b \simeq (a[i]{:=}u),\ i \not\simeq j,\ j \simeq k,\ a[j] \not\simeq b[k] \ &\vdash_{\mathsf{Arr}} \bot \\
a \not\simeq b \ &\vdash_{\mathsf{Arr}} a[\mathsf{diff}(a,b)] \not\simeq b[\mathsf{diff}(a,b)]
\end{aligned}$$

where the last rule is the only one that can produce new terms. Similar to $\mathcal{I}_{\mathsf{EUF}}$, in order to determine whether equalities hold, one may declare all sorts to be Arr-public, with infinitely many Arr-values used as identifiers of congruence classes. Inference rules for eager propagation of equalities can be added.

Assume $\mathcal{T}$ is a theory equipped with a $\mathcal{T}$-satisfiability procedure. A *black-box theory module* $\mathcal{I}_{\mathcal{T}}$ comprises the rule

$$l_1 \leftarrow \mathfrak{b}_1, \ldots, l_m \leftarrow \mathfrak{b}_m \vdash_{\mathcal{T}} \bot$$

that fires when the Boolean assignment on the left is found $\mathcal{T}$-unsatisfiable by invoking the $\mathcal{T}$-satisfiability procedure. As with EUF and Arr, non-Boolean sorts can be declared $\mathcal{T}$-public to determine equalities. If the $\mathcal{T}$-satisfiability procedure produces *unsatisfiable cores*, the above rule can be restricted to unsatisfiable cores to make conflict resolution more precise. A black-box theory module can be given for any theory combined by equality sharing.

7 The CDSAT Transition System

CDSAT transforms a *trail*, denoted Γ, which is a sequence of distinct singleton assignments that are either *decisions*, written ${}_{?}A$ to convey guessing, or *justified assignments*, written ${}_{H\vdash}A$. Decisions can be either Boolean or first-order assignments. The justification H in ${}_{H\vdash}A$ is a set of singleton assignments that appear *before* A in the trail. A theory inference $J \vdash_{\mathcal{I}_k} L$ for some k, $1 \leq k \leq n$, can justify adding ${}_{J\vdash}L$ to the trail. Input assignments are justified assignments with empty justification. Other justified assignments can be placed on the trail by conflict-solving transitions. All justified assignments are Boolean except for the input first-order assignments of an SMA problem. A trail can be treated as an assignment by ignoring the order and the justifications of its elements.

Definition 8 (Level) Given a trail $\Gamma = A_0, \ldots, A_m$, the *level* of singleton assignment A_i, $\forall i$, $0 \leq i \leq m$, is $\mathsf{level}_\Gamma(A_i) = 1 + \max\{\mathsf{level}_\Gamma(A_j) \mid j < i\}$, if A_i is a decision, and $\mathsf{level}_\Gamma(A_i) = \mathsf{level}_\Gamma(H)$, if A_i is a justified assignment with justification H. The *level* of a set of singleton assignments $H \subseteq \Gamma$ is $\mathsf{level}_\Gamma(H) = 0$, if $H = \emptyset$, and $\mathsf{level}_\Gamma(H) = \max\{\mathsf{level}_\Gamma(A) \mid A \in H\}$, otherwise.

Similarly to MCSAT and in contrast with CDCL and DPLL($\mathcal{T}$), the levels of assignments on a CDSAT trail are not necessarily in increasing order, as the level of a justified assignment is independent from where it stands on the trail. ${}_{H\vdash}L$ of level q may appear after ${}_{?}A$ of level z, with $z > q$, if $A \notin H$. In

TRAIL RULES		
Decide	$\Gamma \longrightarrow \Gamma, {}_?A$	if A is an acceptable $\mathcal{T}_k$-assignment for $\mathcal{I}_k$ in $\Gamma_{\mathcal{T}_k}$ for some k, $1 \leq k \leq n$
The next three rules assume: $J \vdash_{\mathcal{I}_k} L$, for some k, $1 \leq k \leq n$, $J \subseteq \Gamma$, and $L \notin \Gamma$.		
Deduce	$\Gamma \longrightarrow \Gamma, {}_J\vdash L$	if $\overline{L} \notin \Gamma$ and L is $l \leftarrow \mathfrak{b}$ for some $l \in \mathcal{B}$
Fail	$\Gamma \longrightarrow$ unsat	if $\overline{L} \in \Gamma$ and $\mathsf{level}_\Gamma(J \cup \{\overline{L}\}) = 0$
ConflictSolve	$\Gamma \longrightarrow \Gamma'$	if $\overline{L} \in \Gamma$, $\mathsf{level}_\Gamma(J \cup \{\overline{L}\}) > 0$, $\langle \Gamma; J \cup \{\overline{L}\}\rangle \Longrightarrow^* \Gamma'$
CONFLICT STATE RULES		
UndoClear		
	$\langle \Gamma; E \uplus \{A\}\rangle \Longrightarrow \Gamma^{\leq m-1}$	if A is a first-order decision of level $m > \mathsf{level}_\Gamma(E)$
Resolve		
	$\langle \Gamma; E \uplus \{A\}\rangle \Longrightarrow \langle \Gamma; E \cup H\rangle$	if ${}_H\vdash A$ is in Γ and H does not contain a first-order decision A' whose level is $\mathsf{level}_\Gamma(E \uplus \{A\})$
Backjump		
	$\langle \Gamma; E \uplus \{L\}\rangle \Longrightarrow \Gamma^{\leq m}, {}_E\vdash \overline{L}$	if $\mathsf{level}_\Gamma(L) > m$, where $m = \mathsf{level}_\Gamma(E)$
UndoDecide		
	$\langle \Gamma; E \uplus \{L\}\rangle \Longrightarrow \Gamma^{\leq m-1}, {}_?\overline{L}$	if ${}_H\vdash L$ is in Γ, $m = \mathsf{level}_\Gamma(E) = \mathsf{level}_\Gamma(L)$ and H contains a first-order decision A' of level m

Fig. 2 The CDSAT transition system

this case, ${}_H\vdash L$ is called a *late propagation*. Accordingly, the *restriction* of Γ to its elements of level at most m, written $\Gamma^{\leq m}$, is not necessarily a prefix of Γ.

The *state* of a CDSAT derivation is either a *trail* or a *conflict state*. A *conflict* is an unsatisfiable assignment. A *conflict state* is a pair $\langle \Gamma; E\rangle$, where Γ is a trail and E is a conflict such that $E \subseteq \Gamma$. The CDSAT transition system in Figure 2 comprises *trail rules* denoted $\longrightarrow$ and *conflict state rules* denoted $\Longrightarrow$ with transitive closure $\Longrightarrow^*$. The system relies on a set $\mathcal{B}$ of terms, called *global basis*, to limit the range of terms that can be generated. The global basis depends on the input and is fixed throughout a derivation.

Rule Decide expands a trail Γ with a decision ${}_?A$ provided A is acceptable for a theory module $\mathcal{I}_k$ in its view $\Gamma_{\mathcal{T}_k}$ of Γ. Assume an inference $J \vdash_{\mathcal{I}_k} L$ is applicable as $J \subseteq \Gamma$. The system distinguishes two cases:

1. If $\overline{L} \notin \Gamma$, Deduce expands Γ with L provided its term is in $\mathcal{B}$. Deduce covers both *propagation* of assignments and *conflict explanation*. For the latter, if the $\mathcal{T}_k$-satisfiability procedure detects a conflict in $\Gamma_{\mathcal{T}_k}$, and its inference system $\mathcal{I}_k$ makes available an inference $J \vdash_{\mathcal{I}_k} L$ to explain it, Deduce posts the consequence of this inference on the trail.
2. If $\overline{L} \in \Gamma$, the inference $J \vdash_{\mathcal{I}_k} L$ reveals a conflict in Γ, as $J \subseteq \Gamma$, $J \vdash_{\mathcal{I}_k} L$, and $\overline{L} \in \Gamma$. The conflict is given by $J \cup \{\overline{L}\}$. If $\mathsf{level}_\Gamma(J \cup \{\overline{L}\}) = 0$, Fail returns unsat. If $\mathsf{level}_\Gamma(J \cup \{\overline{L}\}) > 0$, ConflictSolve transfers control to the conflict state rules and returns the resulting trail Γ'.

CDSAT searches for a model by performing decisions with Decide transitions and propagating their consequences with Deduce transitions (Case 1). When a $\mathcal{T}_k$-satisfiability procedure detects a conflict in its view of the trail, one or more $\mathcal{T}_k$-inferences encapsulated in Deduce transitions explain the conflict (Case 1), until an inference $J \vdash_{\mathcal{I}_k} L$ with $\overline{L} \in \Gamma$ becomes applicable so that the conflict surfaces on the trail as the contradiction between L and $\overline{L}$ (Case 2).

Phase 1			
id	trail items	just	lev
0	$-2{\cdot}x-y<0$	{}	0
1	$x+y<0$	{}	0
2	$x<-1$	{}	0
3	$y\leftarrow 0$		1
4	$-y<-2$	{0, 2}	0
	conflict E^1: {3, 4}		1

Phase 2			
id	trail items	just	lev
0	$-2{\cdot}x-y<0$	{}	0
1	$x+y<0$	{}	0
2	$x<-1$	{}	0
3	$-y<-2$	{0, 2}	0
4	$y\leftarrow 4$		1
5	$y<0$	{0, 1}	0
	conflict E^2: {4, 5}		1

Phase 3			
id	trail items	just	lev
0	$-2{\cdot}x-y<0$	{}	0
1	$x+y<0$	{}	0
2	$x<-1$	{}	0
3	$-y<-2$	{0, 2}	0
4	$y<0$	{0, 1}	0
5	$0<-2$	{3, 4}	0
	conflict E^3: {5}		0

Fig. 3 CDSAT derivation in one theory (LRA) for problem R of Example 2

Different conflict state rules apply depending on the conflict. If the conflict contains a *first-order decision* A of level m greater than that of the rest of the conflict, UndoClear undoes A and clears Γ of all assignments of level greater than or equal to m.[2] The removed assignments are not necessarily the most recent ones, due to late propagations. The resulting trail $\Gamma^{\leq m-1}$ is new, even if UndoClear does not add anything: since A was acceptable when decided, it did not cause a conflict when it was added to the trail. If A is now in a conflict, it means that some justified assignment L was deduced *after* decision A even if $\mathsf{level}_\Gamma(L) < m$ (late propagation): $\Gamma^{\leq m-1}$ is new because it contains L.

We exemplify the CDSAT rules introduced so far with the derivation in Figure 3. In figures showing derivations, the trail grows downward and shrinks upward; the columns list the identifiers of the assignments on the trail, the assignments, the justifications, if any, as sets of identifiers, and the levels of the assignments; exits from conflicts break the derivation into phases: in phase 1 the part above the horizontal line is the input; in phase n ($n>1$) the part above the horizontal line is inherited from phase $n-1$ after exiting a conflict.

In Figure 3 the CDSAT derivation begins with a Decide step that places $y\leftarrow 0$ on the trail. The LRA-procedure detects that $\{-2{\cdot}x-y<0,\ x<-1,\ y\leftarrow 0\}$ is a conflict and explains it by inferring $-y<-2$ from $\{-2{\cdot}x-y<0,\ x<-1\}$ by FM-resolution. Deduce places $-y<-2$ on the trail. The $\mathcal{I}_{\mathsf{LRA}}$-evaluation inference $y\leftarrow 0 \vdash \overline{-y<-2}$ reveals conflict E^1 on the trail. Since $\mathsf{level}_\Gamma(E^1)>0$, ConflictSolve fires and UndoClear removes $y\leftarrow 0$ to solve E^1: $y\leftarrow 0$ is A and $-y<-2$ is the late propagation in the above discussion of UndoClear. In phase 2, Decide tries $y\leftarrow 4$. The LRA-procedure finds that $\{-2{\cdot}x-y<0,\ x+y<0,\ y\leftarrow 4\}$ is a conflict and explains it by inferring $y<0$ from $\{-2{\cdot}x-y<0,\ x+y<0\}$ by FM-resolution. Deduce adds $y<0$ to the trail. The $\mathcal{I}_{\mathsf{LRA}}$-evaluation inference $y\leftarrow 4 \vdash \overline{y<0}$ uncovers conflict E^2 that UndoClear solves by removing $y\leftarrow 4$. In phase 3, the LRA-procedure sees that $\{-y<-2,\ y<0\}$ is a conflict and explains it by inferring $0<-2$ by FM-resolution. Deduce puts $0<-2$ on the trail. The $\mathcal{I}_{\mathsf{LRA}}$-evaluation inference $\emptyset \vdash \overline{0<-2}$ discovers conflict E^3. As $\mathsf{level}_\Gamma(E^3)=0$, Fail returns unsat.

Resuming with the CDSAT transition system (see Figure 2), rule Backjump applies if the conflict contains a *Boolean assignment* L whose level is greater

[2] UndoClear is a renaming of *Undo* [5].

Phase 1			
id	trail items	just	lev
0	$\neg a \vee b$	$\{\}$	0
1	$\neg c \vee d$	$\{\}$	0
2	$\neg e \vee \neg f$	$\{\}$	0
3	$f \vee \neg e \vee \neg b$	$\{\}$	0
4	a		1
5	b	$\{0,4\}$	1
6	c		2
7	d	$\{1,6\}$	2
8	e		3
9	$\overline{f}$	$\{2,8\}$	3
conflict	E_1: $\{3,5,8,9\}$		3
conflict	E_2: $\{2,3,5,8\}$		3

Phase 2			
id	trail items	just	lev
0	$\neg a \vee b$	$\{\}$	0
1	$\neg c \vee d$	$\{\}$	0
2	$\neg e \vee \neg f$	$\{\}$	0
3	$f \vee \neg e \vee \neg b$	$\{\}$	0
4	a		1
5	b	$\{0,4\}$	1
6	$\overline{e}$	$\{2,3,5\}$	1
7	d		2

Fig. 4 CDSAT derivation in one theory (Bool) for problem S of Example 1

than that of the rest E of the conflict: the system jumps back to the level of E and adds ${}_{E}\!\vdash\overline{L}$ to the trail. Since $E \uplus \{L\}$ is a conflict, $\overline{L}$ is justified by E. Assignment $\overline{L}$ is a *Unique Implication Point* (UIP) [20]. Rule Resolve unfolds a conflict by replacing a justified assignment A in the conflict with its justification H. Assignment A can be either Boolean or first-order. As the only first-order justified assignments are input assignments with empty justification, if A is first-order, Resolve only removes it from the conflict (not from the trail). Resolve requires that H does not include a first-order decision A' of the same level as that of the conflict. This condition avoids a loop. Assume that A is ${}_{\{A'\}}\!\vdash L$, A' is first-order, $\mathsf{level}_\Gamma(A') = m$, $\mathsf{level}_\Gamma(E \uplus \{A\}) = m$, and $\mathsf{level}_\Gamma(E) < m$: if Resolve unfolds $E \uplus \{A\}$ into $E \uplus \{A'\}$, UndoClear removes A', Decide reiterates A', and Deduce adds ${}_{\{A'\}}\!\vdash L$, the system falls back in the same conflict. This kind of loop can arise only with first-order assignments: if A' is Boolean and Resolve unfolds $E \uplus \{A\}$ into $E \uplus \{A'\}$, Backjump applies with A' as UIP and puts ${}_{E}\!\vdash\overline{A'}$ on the trail, preventing Decide from retrying A'. The point is that a first-order assignment has no flip, as its complement may be an infinite set of values.

We illustrate Backjump and Resolve in the Boolean case, where CDSAT reduces to CDCL, with the CDSAT derivation in Figure 4: it begins with three Decide and Deduce pairs of steps that bring the system to conflict E_1, made of the first CDCL conflict clause $f \vee \neg e \vee \neg b$ and the flips of its literals. Backjump does not apply to E_1, because it contains two elements of level 3, namely e and $\overline{f}$. Thus, Resolve applies and transforms E_1 into E_2, by replacing $\overline{f}$ with its justification $\{e,\ \neg e \vee \neg f\}$. This Resolve transition encodes the resolution step that infers the second CDCL conflict clause $\neg e \vee \neg b$ from $f \vee \neg e \vee \neg b$ and $\neg e \vee \neg f$: clause $\neg e \vee \neg b$ can be read off E_2 as the negation of its subset (conjunction) $\{e, b\}$. Since E_2 has only one element of level 3, namely e, Backjump applies to E_2 and opens phase 2 by placing $\overline{e}$ on the trail with the rest of E_2 as justification. Another decision suffices to find that S is satisfiable.

Whenever CDCL resolves conflict clause $L_1 \vee \ldots \vee L_k$ with justification $\neg L_1 \vee B_1 \vee \ldots \vee B_q$ of $\neg L_1$ to generate conflict clause $C = L_2 \vee \ldots \vee L_k \vee$

$B_1 \vee \ldots \vee B_q$, CDSAT unfolds conflict $E = \{L_1 \vee \ldots \vee L_k, \overline{L_1}, \ldots, \overline{L_k}\}$ with justified assignment $\{\neg L_1 \vee B_1 \vee \ldots \vee B_q, \overline{B_1}, \ldots, \overline{B_q}\} \vdash \overline{L_1}$ to get conflict $E' = \{L_1 \vee \ldots \vee L_k, \overline{L_2}, \ldots, \overline{L_k}, \neg L_1 \vee B_1 \vee \ldots \vee B_q, \overline{B_1}, \ldots, \overline{B_q}\}$, where C can be read off E' as the negation of its subset (conjunction) $\{\overline{L_2}, \ldots, \overline{L_k}, \overline{B_1}, \ldots, \overline{B_q}\}$.

Phase 1

id	trail items	just	lev
0	$f((a[i]{:=}v)[j]) \simeq w$	{}	0
1	$(w{+}\frac{1}{2})^2 \simeq f(u)$	{}	0
2	$i \simeq j$	{}	0
3	$u \simeq v$	{}	0
4	$u{\leftarrow}\mathfrak{c}$		1
5	$v{\leftarrow}\mathfrak{c}$		2
6	$(a[i]{:=}v)[j]{\leftarrow}\mathfrak{c}$		3
7	$w{\leftarrow}0$		4
8	$f((a[i]{:=}v)[j]){\leftarrow}0$		5
9	$f(u){\leftarrow}\frac{1}{4}$		6
10	$u \simeq (a[i]{:=}v)[j]$	{4, 6}	3
11	$f(u) \not\simeq f((a[i]{:=}v)[j])$	{8, 9}	6
	conflict E^1: {10, 11}		6

Phase 2

id	trail items	just	lev
0	$f((a[i]{:=}v)[j]) \simeq w$	{}	0
1	$(w{+}\frac{1}{2})^2 \simeq f(u)$	{}	0
2	$i \simeq j$	{}	0
3	$u \simeq v$	{}	0
4	$u{\leftarrow}\mathfrak{c}$		1
5	$v{\leftarrow}\mathfrak{c}$		2
6	$(a[i]{:=}v)[j]{\leftarrow}\mathfrak{c}$		3
7	$u \simeq (a[i]{:=}v)[j]$	{4, 6}	3
8	$f(u) \simeq f((a[i]{:=}v)[j])$	{7}	3
9	$f(u) \simeq w$	{0, 8}	3
10	$(w{+}\frac{1}{2})^2 \simeq w$	{1, 9}	3
	conflict E_1^2: {10}		3
	conflict E_2^2: {1, 9}		3
	conflict E_3^2: {0, 1, 8}		3
	conflict E_4^2: {0, 1, 7}		3

Phase 3

id	trail items	just	lev
0	$f((a[i]{:=}v)[j]) \simeq w$	{}	0
1	$(w{+}\frac{1}{2})^2 \simeq f(u)$	{}	0
2	$i \simeq j$	{}	0
3	$u \simeq v$	{}	0
4	$u \not\simeq (a[i]{:=}v)[j]$	{0, 1}	0
5	$u{\leftarrow}\mathfrak{c}$		1
6	$v{\leftarrow}\mathfrak{c}$		2
7	$(a[i]{:=}v)[j]{\leftarrow}\mathfrak{d}$		3
8	$v \not\simeq (a[i]{:=}v)[j]$	{6, 7}	3
	conflict E^3: {2, 8}		3

Phase 4

id	trail items	just	lev
0	$f((a[i]{:=}v)[j]) \simeq w$	{}	0
1	$(w{+}\frac{1}{2})^2 \simeq f(u)$	{}	0
2	$i \simeq j$	{}	0
3	$u \simeq v$	{}	0
4	$u \not\simeq (a[i]{:=}v)[j]$	{0, 1}	0
5	$v \simeq (a[i]{:=}v)[j]$	{2}	0
	conflict E^4: {3, 4, 5}		0

Fig. 5 CDSAT derivation in three theories (EUF, Arr, and RA) for problem P of Example 3

We demonstrate the behavior of CDSAT on a combination problem with the derivation in Figure 5: it starts with a series of decisions, from $u{\leftarrow}\mathfrak{c}$ through $f(u){\leftarrow}\frac{1}{4}$, where $\mathfrak{c}$ is an Arr-value of sort V, and $v{\leftarrow}\mathfrak{c}$ is the only acceptable choice given $u \simeq v$ and $u{\leftarrow}\mathfrak{c}$. The pair $v{\leftarrow}\mathfrak{c}$ is a decision, not a deduction, because theory module inferences only deduce Boolean assignments. The EUF-procedure sees that $u{\leftarrow}\mathfrak{c}$, $(a[i]{:=}v)[j]{\leftarrow}\mathfrak{c}$, $f((a[i]{:=}v)[j]){\leftarrow}0$, and $f(u){\leftarrow}\frac{1}{4}$ form a conflict, and explains it by equality inferences generating $u \simeq (a[i]{:=}v)[j]$ and $f(u) \not\simeq f((a[i]{:=}v)[j])$. Deduce puts them on the trail and $\mathcal{I}_{\mathsf{EUF}}$ reveals conflict E^1 as $E^1 \vdash_{\mathcal{I}_{\mathsf{EUF}}} \bot$. Backjump solves E^1 with $f(u) \not\simeq f((a[i]{:=}v)[j])$ as L: the system jumps back to $\mathsf{level}_\Gamma(u \simeq (a[i]{:=}v)[j]) = 3$ with $f(u) \simeq f((a[i]{:=}v)[j])$ on the trail justified by $u \simeq (a[i]{:=}v)[j]$ (phase 2). The RA-procedure finds that $(w{+}\frac{1}{2})^2 \simeq f(u)$, $f(u) \simeq f((a[i]{:=}v)[j])$, and $f((a[i]{:=}v)[j]) \simeq w$ make a conflict, and explains it by inferring by transitivity $f(u) \simeq w$ from $\{f(u) \simeq f((a[i]{:=}v)[j]),\ f((a[i]{:=}v)[j]) \simeq w\}$ and $(w{+}\frac{1}{2})^2 \simeq w$ from $\{(w{+}\frac{1}{2})^2 \simeq f(u),\ f(u) \simeq w\}$. Deduce adds them to the trail. $\mathcal{I}_{\mathsf{RA}}$ reveals conflict E_1^2 as $(w{+}\frac{1}{2})^2 \simeq w$ reduces to $w^2{+}\frac{1}{4} \simeq 0$ and $\{w^2{+}\frac{1}{4} \simeq 0\} \vdash_{\mathsf{RA}} \bot$ in

Phase 1			
id	trail items	just	lev
0	$(x{>}1) \vee (y{<}0)$	{}	0
1	$(x{<}{-}1) \vee (y{>}0)$	{}	0
2	$x{\leftarrow}0$		1
3	$\overline{x{>}1}$	{2}	1
4	$\overline{x{<}{-}1}$	{2}	1
5	$y{<}0$	{0, 3}	1
6	$y{>}0$	{1, 4}	1
7	$0{<}0$	{5, 6}	1
	conflict E_1: {7}		1
	conflict E_2: {5, 6}		1
	conflict E_3: {0, 3, 6}		1
	conflict E_4: {0, 1, 3, 4}		1

Phase 2			
id	trail items	just	lev
0	$(x{>}1) \vee (y{<}0)$	{}	0
1	$(x{<}{-}1) \vee (y{>}0)$	{}	0
2	$x{>}1$		1
3	$x{\leftarrow}2$		2
4	$\overline{x{<}{-}1}$	{3}	2
5	$y{>}0$	{1, 4}	2
6	$y{\leftarrow}1$		3
7	$\overline{y{<}0}$	{6}	3

Fig. 6 CDSAT derivation in two theories (Bool and LRA)

any $\mathcal{I}_{\mathsf{RA}}$. Three Resolve steps transform conflict E_1^2 into conflict E_4^2. Resolve does not apply to $u \simeq (a[i]{:=}v)[j]$ in E_4^2 because its justification contains $(a[i]{:=}v)[j]{\leftarrow}\mathfrak{c}$ which is first-order and has level 3, the same as E_4^2. Then Backjump solves E_4^2 by jumping back to level 0 and flipping $u \simeq (a[i]{:=}v)[j]$ into $u \not\simeq (a[i]{:=}v)[j]$ of phase 3, whose justification was built by Resolve steps. Phase 3 opens with three decisions, where $v{\leftarrow}\mathfrak{c}$ is forced by $u \simeq v$ and $u{\leftarrow}\mathfrak{c}$, and another Arr-value $\mathfrak{d}$ of sort V is used for $(a[i]{:=}v)[j]$, since $u \not\simeq (a[i]{:=}v)[j]$ and $u{\leftarrow}\mathfrak{c}$ exclude $\mathfrak{c}$. The Arr-procedure catches that $\{i \simeq j,\ u \simeq v,\ u \not\simeq (a[i]{:=}v)[j]\}$ is a conflict and explains it by the equality inference yielding $v \not\simeq (a[i]{:=}v)[j]$, that Deduce places on the trail. Since $i \simeq j, v \not\simeq (a[i]{:=}v)[j] \vdash_{\mathsf{Arr}} \bot$, module $\mathcal{I}_{\mathsf{Arr}}$ exposes conflict E^3. Backjump solves E^3 by jumping back to level 0 and flipping $v \not\simeq (a[i]{:=}v)[j]$ into $v \simeq (a[i]{:=}v)[j]$ of phase 4. Conflict E^4 violates transitivity of equality, and because its level is 0, Fail halts the derivation.

We complete the description of the CDSAT transition system in Figure 2 with rule UndoDecide.[3] This rule applies to a conflict containing a *Boolean justified assignment* L (no UndoClear) whose level is the same as that of the rest of the conflict (no Backjump), and whose justification includes a first-order decision A' whose level is the same as that of the conflict (no Resolve). UndoDecide undoes A' and replaces it by a Boolean decision on $\overline{L}$. UndoClear and UndoDecide have no match in CDCL or DPLL($\mathcal{T}$), where the conflict-driven reasoning is only propositional without first-order assignments.

In order to elucidate the application of UndoDecide, we consider the CDSAT derivation in Figure 6, which begins with one Decide and four Deduce transitions doing propagations. The LRA-procedure sees conflict $\{y{<}0,\ y{>}0\}$ and explains it by inferring $0{<}0$ that Deduce puts on the trail. Three Resolve steps turn conflict E_1 into E_4, which contains two Boolean justified assignments $\overline{x{>}1}$ and $\overline{x{<}{-}1}$ that fit with UndoDecide: both have level 1 and are justified by first-order decision $x{\leftarrow}0$ of level 1. UndoDecide chooses to flip $\overline{x{>}1}$ and the derivation proceeds to find the problem satisfiable.

[3] UndoDecide replaces *T-backjump-decide* [9] or *Semantic split* [15].

$$
\begin{array}{lrll}
l_0 : & -2{\cdot}x - y < 0 & & \\
l_1 : & x + y < 0 & & \\
l_2 : & x < -1 & & \\
l_3 : & -y < -2 & & (l_0 + 2l_2) \\
l_4 : & x < -2 & & (l_1 + l_3) \\
l_5 : & -y < -4 & & (l_0 + 2l_4) \\
l_6 : & x < -4 & & (l_1 + l_5) \\
l_7 : & -y < -8 & & (l_0 + 2l_6) \\
\ldots & & &
\end{array}
$$

Fig. 7 A diverging series of inferences in linear rational arithmetic

8 Theory Module Requirements for Termination and Completeness

Since a motivation for CDSAT is combining conflict-driven theory satisfiability procedures that may introduce *new* terms, theory module inferences need to have this capability: $J \vdash_{\mathcal{I}} L$ may introduce in L new terms that occur neither in J nor in the input. The generation of new terms can jeopardize *termination*, as shown in Figure 7 where FM-resolution generates an infinite series of steps from problem R of Example 2. The known solution is to restrict inferences to draw new terms from a *finite basis* [2,9]. In this section we apply this approach to CDSAT modules in order to extend it to theory combination. We begin by saying that a set X of terms is *closed* if (i) for all $u \in X$, $t \lhd u$ implies $t \in X$, and (ii) for all $t, u \in X$ of sort s other than prop, $(t \simeq_s u) \in X$. The first property will be used to capture all terms in $G(H)$ for an assignment H, and the second one to capture equalities in theory views of H and relevant equalities. The *closure* of a set X of terms, denoted $\Downarrow X$, is the smallest closed set containing X. The closure operation is *idempotent*, as $\Downarrow(\Downarrow X) = \Downarrow X$, and *monotone*: if $X \subseteq Y$ then $\Downarrow X \subseteq \Downarrow Y$. Then a *local basis* for a theory module $\mathcal{I}$ is a function $\mathsf{basis}_{\mathcal{I}}$ that maps any finite set X of terms to the *finite* set of terms that $\mathcal{I}$-inferences may introduce from an input whose terms are in X.

Definition 9 (Local basis) A *local basis* for signature Σ is a function basis from sets of terms to sets of terms, such that for all sets X and Y of terms:
- $X \subseteq \mathsf{basis}(X)$ (*extensiveness*),
- If X is finite then $\mathsf{basis}(X)$ is finite (*finiteness*),
- $\mathsf{basis}(X) = \mathsf{basis}(\Downarrow X) = \Downarrow \mathsf{basis}(X)$ (*closure*),
- If $X \subseteq Y$ then $\mathsf{basis}(X) \subseteq \mathsf{basis}(Y)$ (*monotonicity*),
- $\mathsf{basis}(\mathsf{basis}(X)) = \mathsf{basis}(X)$ (*idempotence*), and
- $\mathsf{fv}_{\Sigma}(\mathsf{basis}(X)) \subseteq \mathsf{fv}_{\Sigma}(X) \cup \mathcal{V}_{\infty}$ (*no additional free Σ-variables*),

where the last requirement excludes the introduction of foreign terms.

All theory modules $\mathcal{I}_k$, $1 \leq k \leq n$, must be equipped with a local basis named $\mathsf{basis}_{\mathcal{I}_k}$ or basis_k (e.g., we write $\mathsf{basis}_{\mathsf{LRA}}$ for $\mathsf{basis}_{\mathcal{I}_{\mathsf{LRA}}}$). Local bases only play a role in the proof of termination of CDSAT and in the completeness requirement for theory modules, and it is not necessary to implement them. The divergence in Figure 7 can be avoided by assuming a total precedence $<_{\mathsf{LRA}}$ on free Σ_{LRA}-variables [15], and taking as $\mathsf{basis}_{\mathsf{LRA}}$ the function that

adds to its argument all the terms generated by positivization inferences and FM-resolution inferences $t_1 \preccurlyeq_1 x, x \preccurlyeq_2 t_2 \vdash_{\mathsf{LRA}} t_1 \preccurlyeq_3 t_2$, where x is the $<_{\mathsf{LRA}}$-maximum free Σ_{LRA}-variable in both $t_1 \preccurlyeq_1 x$ and $x \preccurlyeq_2 t_2$. For Figure 7, assume that $y <_{\mathsf{LRA}} x$. Then l_3, generated by $-y < 2{\cdot}x,\ 2{\cdot}x < -2 \vdash_{\mathsf{LRA}} -y < -2$, is in the local basis, whereas l_4, generated by $x < -y,\ -y < -2 \vdash_{\mathsf{LRA}} x < -2$, is not, so that the series of inferences halts. For the rest of this section, let theory $\mathcal{T}$ with signature Σ be one of $\mathcal{T}_1, \ldots, \mathcal{T}_n$, and let $\mathcal{T}^+$ be its extension. Towards *completeness*, we define how a $\mathcal{T}$-module *expands* a $\mathcal{T}$-assignment.

Definition 10 (Assignment expansion) A $\mathcal{T}$-module $\mathcal{I}$ with local basis $\mathsf{basis}_{\mathcal{I}}$ *expands* a $\mathcal{T}$-assignment J by adding either (1) a $\mathcal{T}$-assignment A that is acceptable for J and $\mathcal{I}$, or (2) a Boolean assignment $l{\leftarrow}\mathfrak{b}$ derived by an $\mathcal{I}$-inference $J' \vdash_{\mathcal{I}} (l{\leftarrow}\mathfrak{b})$ from a $\mathcal{T}$-assignment J', $J' \subseteq J$, such that $(l{\leftarrow}\mathfrak{b}) \notin J$ and $l \in \mathsf{basis}_{\mathcal{I}}(J)$.

Case (1) covers Decide and Case (2) covers Deduce, Fail, and ConflictSolve. Local bases contribute to completeness by providing enough terms in Case (2).

Definition 11 (One-theory-consistency) Given theory $\mathcal{T}$ and extension $\mathcal{T}^+$, a $\mathcal{T}$-assignment J is *consistent with* $\mathcal{T}$ if there exists a $\mathcal{T}^+[\mathcal{V}]$-model $\mathcal{M}$ such that $\mathcal{M} \models J$, assuming $\mathsf{fv}_{\Sigma}(J) \subseteq \mathcal{V}$.

For a $\mathcal{T}$-module to be complete with respect to its own theory it suffices to be capable of expanding any plausible assignment that is not consistent.

Definition 12 (One-theory-completeness) Given theory $\mathcal{T}$, a $\mathcal{T}$-module $\mathcal{I}$ is *complete for* $\mathcal{T}$, if, for all plausible $\mathcal{T}$-assignments J, either J is consistent with $\mathcal{T}$ or $\mathcal{I}$ can expand J.

Note that expanding a consistent assignment is not needed for completeness. For theory combination, completeness requires a stronger property, as the existence of a model for each theory does not guarantee the existence of a model for their union: for disjoint theories, the models have to agree on equalities between shared terms and on the cardinalities of the domains interpreting shared sorts. In equality sharing the only shared terms are variables and the theories are *stably infinite*, so that the domains are assumed to be countably infinite for all shared sorts except prop. However, combinations involving non-stably infinite theories are useful: consider combining the theory of arrays with a theory that shares the sort of array elements and requires it to be finite.

CDSAT generalizes equality sharing in at least two ways. First, CDSAT integrates conflict-driven satisfiability procedures whose communication is not limited to equalities between shared variables, as they have full access to the trail. Thus, we consider equalities between *shared terms*. Second, CDSAT handles also *non-stably infinite* theories, such as the theory of *bitvectors*, by requiring that one of the theories in the combination, called *leading theory*, knows all sorts. Let $\mathcal{T}_1$ be the leading theory, so that $S_1{=}S_\infty$. For a $\mathcal{T}_\infty$-model to endorse globally an assignment H, it suffices that there are a $\mathcal{T}_1$-model endorsing $H_{\mathcal{T}_1}$, and, for all other theories $\mathcal{T}_k$, a $\mathcal{T}_k$-model endorsing $H_{\mathcal{T}_k}$ and agreeing with the $\mathcal{T}_1$-model on equalities between shared terms and cardinalities of shared sorts.

Definition 13 (Leading-theory-compatibility) Let $\mathcal{T}_1$ be the leading theory, $\mathcal{T}$ and S stand for $\mathcal{T}_k$ and S_k, $2 \leq k \leq n$, and N be a set of terms. A $\mathcal{T}$-assignment J is *leading-theory-compatible with* $\mathcal{T}$ *sharing* N, if for all $\mathcal{T}_1^+[\mathcal{V}_1]$-model $\mathcal{M}_1$ such that $\mathcal{M}_1 \models J_{\mathcal{T}_1}$ with $\mathsf{fv}_{\Sigma_1}(J \cup N) \subseteq \mathcal{V}_1$, there exists a $\mathcal{T}^+[\mathcal{V}]$-model $\mathcal{M}$ with $\mathsf{fv}_{\Sigma}(J \cup N) \subseteq \mathcal{V}$, such that (i) $\mathcal{M} \models J$, (ii) for all sorts $s \in S$, $|s^{\mathcal{M}}| = |s^{\mathcal{M}_1}|$, and (iii) for all $s \in S$ and terms $u, u' \in N$ of sort s, $\mathcal{M}(u) = \mathcal{M}(u')$ if and only if $\mathcal{M}_1(u) = \mathcal{M}_1(u')$.

Definition 14 (Leading-theory-completeness) For all nonleading theories $\mathcal{T}$, a $\mathcal{T}$-module $\mathcal{I}$ is *leading-theory-complete* for $\mathcal{T}$, if for all plausible $\mathcal{T}$-assignments J, either J is leading-theory-compatible with $\mathcal{T}$ sharing $G(J)$ or $\mathcal{I}$ can expand J.

In a combination of stably infinite theories, the leading theory is a theory $\mathcal{T}_{\mathbb{N}}$, whose models interpret all sorts except prop as domains with the cardinality of $\mathbb{N}$, the set of the natural numbers. In practice, $\mathcal{T}_{\mathbb{N}}$ is not included in the combination. In other cases, a leading theory is either present or added.

Example 9 Suppose $\mathcal{T}_2$ and $\mathcal{T}_3$ share sort s, $|s^{\mathcal{M}_2}| \leq 4$ for all $\mathcal{T}_2$-models $\mathcal{M}_2$, and $|s^{\mathcal{M}_3}| \geq 2$ for all $\mathcal{T}_3$-models $\mathcal{M}_3$. The leading theory $\mathcal{T}_1$ has $2 \leq |s^{\mathcal{M}_1}| \leq 4$ for all $\mathcal{T}_1$-models $\mathcal{M}_1$. Its theory module $\mathcal{I}_1$ includes a rule $\{u_i \not\simeq u_j \mid 1 \leq i \neq j \leq 5\} \vdash \bot$, where $u_1, \ldots, u_5$ are any five distinct terms of sort s. If $|s^{\mathcal{M}_3}| \geq 5$ for all $\mathcal{T}_3$-models $\mathcal{M}_3$, the leading theory is the inconsistent theory with false as axiom and $\vdash \bot$ as only inference, so that CDSAT returns unsat for all inputs.

Example 10 Suppose $\mathcal{T}_2$ is the theory of bitvectors and $\mathcal{T}_3$ has a unary injective function $f\colon s \to bv[2]$ from a sort s unknown to $\mathcal{T}_2$ into the sort $bv[2]$ of bitvectors of length 2. The leading theory $\mathcal{T}_1$ knows all sorts and for all its models $\mathcal{M}_1$, $|(bv[k])^{\mathcal{M}_1}| = 2^k$ and $|s^{\mathcal{M}_1}| \leq 4$. Its theory module has the 5-term rule of Example 9 and a similar 2^k-term rule for each $bv[k]$. A black-box theory module for bitvectors and a theory module with rules $x \simeq y, f(x) \not\simeq f(y) \vdash \bot$ (congruence) and $x \not\simeq y, f(x) \simeq f(y) \vdash \bot$ (injectivity) for $\mathcal{T}_3$ are leading-theory-complete.

9 Soundness, Termination, and Completeness of CDSAT

9.1 Soundness of CDSAT

We begin with soundness, for which it suffices that all theory modules are sound. We write $\mathcal{T}_\infty^+$-model for a $\mathcal{T}_\infty^+[\mathcal{V}]$-model such that $\mathsf{fv}(J) \subseteq \mathcal{V}$ for all involved assignments J, and in all statements global endorsement by a $\mathcal{T}_\infty^+$-model can be replaced by endorsement by a $\mathcal{T}_\infty$-model if all involved assignments are Boolean. The first lemma follows from soundness of the theory modules.

Lemma 1 *For all k, $1 \leq k \leq n$, $\mathcal{I}_k$-inferences $J \vdash_{\mathcal{I}_k} L$, and $\mathcal{T}_\infty^+$-model $\mathcal{M}$, if $\mathcal{M} \models^G J$ then $\mathcal{M} \models L$.*

Proof Let $\mathcal{M}$ be a $\mathcal{T}_\infty^+[\mathcal{V}]$-model, with $\mathsf{fv}(J \cup \{L\}) \subseteq \mathcal{V}$, such that $\mathcal{M} \models^G J$, which means $\mathcal{M} \models J_{\mathcal{T}_\infty}$. Since J is a $\mathcal{T}_k$-assignment, $\mathcal{M} \models J_{\mathcal{T}_\infty}$ is equivalent to $\mathcal{M} \models J_{\mathcal{T}_k}$. Let $\mathcal{V}' = \mathsf{fv}_{\Sigma_k}(J \cup \{L\})$. Since $\Sigma_k \subseteq \Sigma_\infty$, $\mathsf{fv}(\mathcal{V}') = \mathsf{fv}(J \cup \{L\}) \subseteq \mathcal{V}$. Let $\mathcal{M}'$ be the $\Sigma_k^+[\mathcal{V}']$-interpretation defined by: $s^{\mathcal{M}'} = s^{\mathcal{M}}$ for all $s \in S_k$, $f^{\mathcal{M}'} = f^{\mathcal{M}}$ for all $f \in F_k^+$, and $t^{\mathcal{M}'} = \mathcal{M}(t)$ for all terms $t \in \mathcal{V}'$. $\mathcal{M}'$ interprets $\Sigma_k^+[\mathcal{V}']$-terms and Σ_k^+-sentences (for $\mathcal{T}_k^+$-axioms) like $\mathcal{M}$. Thus, $\mathcal{M} \models J_{\mathcal{T}_k}$ implies $\mathcal{M}' \models J_{\mathcal{T}_k}$. By soundness of $\mathcal{I}_k$, it follows that $\mathcal{M}' \models L$, which implies $\mathcal{M} \models L$ by construction.

Then, we define *soundness* of justified assignments, trails, and states, and we show that CDSAT transforms sound states into sound states.

Definition 15 (Sound states) For all input assignments H_0 and states Γ or $\langle \Gamma; E \rangle$ derived by CDSAT from H_0: (1) a justified assignment $({}_{H\vdash}A) \in \Gamma$ is *sound* if for all $\mathcal{T}_\infty^+$-model $\mathcal{M}$, $\mathcal{M} \models^G H_0 \cup H$ implies $\mathcal{M} \models A$; (2) a trail or state Γ is *sound* if all its justified assignments are; and (3) a state $\langle \Gamma; E \rangle$ is *sound* if Γ is sound and $H_0 \cup E$ is unsatisfiable.

Lemma 2 *For all input assignments H_0 and states Γ or $\langle \Gamma; E \rangle$ derived by CDSAT from H_0, (1) if $\langle \Gamma; E \rangle$ is sound and $\langle \Gamma; E \rangle \Longrightarrow^* \Gamma'$, then Γ' is sound; (2) if Γ is sound and $\Gamma \longrightarrow \Gamma'$, then Γ' is sound.*

Proof Claim (1) is proved by induction on the length of $\langle \Gamma; E \rangle \Longrightarrow^* \Gamma'$. The base case covers UndoClear, UndoDecide, and Backjump. For UndoClear we have $\langle \Gamma; E \uplus \{A\} \rangle \Longrightarrow \Gamma^{\leq m-1}$, and $\Gamma^{\leq m-1}$ is sound, because Γ is. For UndoDecide we have $\langle \Gamma; E \uplus \{L\} \rangle \Longrightarrow \Gamma^{\leq m-1}, {}_?\overline{L}$: $\Gamma^{\leq m-1}$ is sound, because Γ is, and $\Gamma^{\leq m-1}, {}_?\overline{L}$ is sound as decisions do not affect soundness. For Backjump we have $\langle \Gamma; E \uplus \{L\} \rangle \Longrightarrow \Gamma^{\leq m}, {}_{E\vdash}\overline{L}$: $\Gamma^{\leq m}$ is sound because trail Γ is; since $H_0 \cup (E \uplus \{L\})$ is unsatisfiable, if $\mathcal{M} \models^G H_0 \cup E$ then $\mathcal{M} \models \overline{L}$, and ${}_{E\vdash}\overline{L}$ is also sound. The induction step covers a derivation starting with a Resolve replacing ${}_{H\vdash}A$ in the conflict with its justification: $\langle \Gamma; E \uplus \{A\} \rangle \Longrightarrow \langle \Gamma; E \cup H \rangle \Longrightarrow^* \Gamma'$. We show that if $\langle \Gamma; E \uplus \{A\} \rangle$ is sound then $\langle \Gamma; E \cup H \rangle$ is sound, and then Γ' is sound by induction hypothesis. Since Γ does not change, it suffices to show that $H_0 \cup E \cup H$ is unsatisfiable. By way of contradiction, assume that $\mathcal{M} \models^G H_0 \cup E \cup H$ for a $\mathcal{T}_\infty^+$-model $\mathcal{M}$. Since $({}_{H\vdash}A) \in \Gamma$ and Γ is sound, it follows that $\mathcal{M} \models^G H_0 \cup (E \uplus \{A\})$ contradicting the soundness of $\langle \Gamma; E \uplus \{A\} \rangle$. Claim (2) is vacuously true for a Decide or Fail transition. A Deduce or ConflictSolve transition relies on an inference $J \vdash_{\mathcal{I}_k} L$: for Deduce, the claim follows from Lemma 1; for ConflictSolve, by Lemma 1, if $\mathcal{M} \models^G J$ then $\mathcal{M} \models L$, so that $H_0 \cup J \cup \{\overline{L}\}$ is unsatisfiable. Thus, $\langle \Gamma; J \cup \{\overline{L}\} \rangle$ is sound, and Γ' is sound by the first part of this lemma.

The next lemma is used in the soundness theorem.

Lemma 3 *For all input assignments H_0, trails Γ derived by CDSAT from H_0, and $\mathcal{T}_\infty^+$-models $\mathcal{M}$, if $\mathcal{M} \models^G H_0$ then $\mathcal{M} \models^G \Gamma^{\leq 0}$.*

Proof For all trails Γ derived by CDSAT from H_0, a singleton assignment in $\Gamma^{\leq 0}$ is either an assignment in H_0 or a Boolean justified assignment ${}_{H\vdash}L$, due to either a theory inference or a Backjump transition, and such that $\mathsf{level}_\Gamma(H) = 0$. The proof is by induction on the inference depth of justified assignments. An assignment with inference depth 0 is an input assignment, for which the claim is trivially true. Assume the claim is true for inference depth z. Let $({}_{H\vdash}L) \in \Gamma^{\leq 0}$ have inference depth $z+1$. For all $\mathcal{T}_\infty^+$-models $\mathcal{M}$, $\mathcal{M} \models^G H_0$ implies $\mathcal{M} \models^G H$ by induction hypothesis, and $\mathcal{M} \models^G H_0 \cup H$ implies $\mathcal{M} \models L$ because ${}_{H\vdash}L$ is sound by Lemma 2.

Theorem 1 (Soundness) *For all input assignments H, if a CDSAT derivation from H reaches state* unsat*, assignment H is unsatisfiable.*

Proof By way of contradiction, assume that there is a $\mathcal{T}_\infty^+[\mathcal{V}]$-model $\mathcal{M}$ with $\mathsf{fv}(H) \subseteq \mathcal{V}$ such that $\mathcal{M} \models^G H$. Let $\Gamma \longrightarrow \mathsf{unsat}$ be the last step of the derivation: it is a Fail transition based on an inference $J \vdash_{\mathcal{I}_k} L$, for some $\mathcal{I}_k$, $1 \leq k \leq n$, such that $J \subseteq \Gamma$, $\overline{L} \in \Gamma$, and $\mathsf{level}_\Gamma(J \cup \{\overline{L}\}) = 0$ (see Figure 2). By picking at random elements in the domains of $\mathcal{M}$ to interpret the variables in $\mathsf{fv}(\Gamma) \setminus \mathsf{fv}(H)$, we get a model $\mathcal{M}'$ that interprets all variables in $\mathsf{fv}(\Gamma)$ and such that $\mathcal{M}' \models^G H$. By Lemma 3, $\mathcal{M}' \models^G \Gamma^{\leq 0}$, which includes $\mathcal{M}' \models^G J$ and $\mathcal{M}' \models \overline{L}$, as $\mathsf{level}_\Gamma(J \cup \{\overline{L}\}) = 0$. Since $J \vdash_{\mathcal{I}_k} L$, by Lemma 1 it follows that $\mathcal{M}' \models L$, a contradiction.

9.2 Termination of CDSAT

We begin the proof of termination with two preliminary lemmas.

Lemma 4 *For all input assignments H, trails Γ derived by CDSAT from H, theories $\mathcal{T}_k$ $(1 \leq k \leq n)$, terms t, and $\mathcal{T}_k$-values $\mathfrak{c}_1$ and $\mathfrak{c}_2$ with $\mathfrak{c}_1 \neq \mathfrak{c}_2$, if $(t \leftarrow \mathfrak{c}_1) \in \Gamma$ and $(t \leftarrow \mathfrak{c}_2) \in \Gamma$, then $(t \leftarrow \mathfrak{c}_1) \in H$ and $(t \leftarrow \mathfrak{c}_2) \in H$.*

Proof By induction on the length of the CDSAT derivation: the base case is trivial, and the induction hypothesis is that Γ satisfies the claim. For the induction step, only Decide, Deduce, Backjump, or UndoDecide add assignments, but none adds a $t \leftarrow \mathfrak{c}_2$ if $t \leftarrow \mathfrak{c}_1$ with $\mathfrak{c}_1 \neq \mathfrak{c}_2$ is in Γ: Decide only adds acceptable assignments; Deduce adds a Boolean assignment provided its flip is not in Γ; and Backjump and UndoDecide add a Boolean assignment removing its flip.

It follows as a corollary that CDSAT preserves plausibility. We say that an assignment H *is in* $\mathcal{B}$ if $(t \leftarrow \mathfrak{c}) \in H$ implies $t \in \mathcal{B}$. If $\mathcal{B}$ is closed, H *is in* $\mathcal{B}$ implies $G(H) \subseteq \mathcal{B}$. If $\mathcal{B}$ is finite, the length of derived trails is upper bounded:

Lemma 5 *For all input assignments H and trails Γ derived by CDSAT from H with a closed global basis $\mathcal{B}$, if H is in $\mathcal{B}$ then Γ is in $\mathcal{B}$.*

Proof By induction on the length of the CDSAT derivation, H is in $\mathcal{B}$ by hypothesis (base case), Γ is in $\mathcal{B}$ (induction hypothesis), and for the induction

step, only Decide and Deduce may introduce new terms. Deduce requires the new formula to belong to $\mathcal{B}$. Decide requires the assignment to be acceptable, hence that the assigned term is relevant: it is either a term $u \in G(\Gamma)$ or an equality $u_1 \simeq u_2$ with $u_1, u_2 \in G(\Gamma)$. In the first case, $u \in \mathcal{B}$ by induction hypothesis as $\mathcal{B}$ is closed. In the second case, $u_1, u_2 \in \mathcal{B}$ by induction hypothesis as $\mathcal{B}$ is closed, and then $(u_1 \simeq u_2) \in \mathcal{B}$ because $\mathcal{B}$ is closed.

Corollary 1 *For all input assignments H and trails Γ derived by CDSAT from H with a closed finite global basis $\mathcal{B}$, if H is in $\mathcal{B}$ then $|\Gamma| \leq |H| + n{\cdot}|\mathcal{B}|$.*

Proof Lemma 4 prevents distinct assignments to the same term, except from any inherited from H, hence the $|H|$ in the upper bound. As Lemma 5 ensures that all terms come from $\mathcal{B}$, the result follows as there are n theory modules.

Let $max = |H| + n{\cdot}|\mathcal{B}|$. A first-order decision is *bad* for a level, if it would be unacceptable should the trail roll back to that level:

Definition 16 (Bad decision) A decision $_?A$ in Γ is *bad for level i*, if it is a first-order $\mathcal{T}_k$-assignment that is not acceptable for $\mathcal{I}_k$ in $(\Gamma^{\leq i})_{\mathcal{T}_k}$.

Decision $_?A$ is bad for level i if $(\Gamma^{\leq i})_{\mathcal{T}_k}$ contains late propagations that make $_?A$ unacceptable. In Figure 3, UndoClear solves conflict E^1 by undoing $y{\leftarrow}0$, and $y{\leftarrow}0$ is bad for level 0 because level 0 contains the late propagation $-y < -2$. Given a trail Γ the *measure of its decision $_?A$ of level i* is the triple $MA_i = (\mathsf{so}_i^\Gamma, w_i^\Gamma, \mathsf{bad}_i^\Gamma)$, where $\mathsf{so}_i^\Gamma = 0$ if $_?A$ is Boolean, $\mathsf{so}_i^\Gamma = 1$ if $_?A$ is first-order; $w_i^\Gamma = max - |\Gamma^{\leq i}|$; and bad_i^Γ is the number of decisions in Γ that are bad for level i. Note that w_0^Γ and bad_0^Γ are defined. The *measure of a trail Γ* with m decisions is the $3{\cdot}max$-tuple $\mathsf{Tuple}(\Gamma) = (w_0^\Gamma, \mathsf{bad}_0^\Gamma, NA_1, \ldots, NA_m, 2, \ldots, 2)$, where for all i, $1 \leq i \leq m$, NA_i is MA_i stripped of parentheses. The measure of a trail Γ is a tuple of length $3{\cdot}max$, because the decision measure is a triple. By convention, the measure of unsat is a $3{\cdot}max$-tuple filled with 0's. Let $>$ denote the well-founded ordering on $\mathbb{N}$ and $>_{\mathsf{lex}}$ its lexicographic extension. The ordering on trails is defined by $\Gamma \succ \Gamma'$ if and only if $\mathsf{Tuple}(\Gamma) >_{\mathsf{lex}} \mathsf{Tuple}(\Gamma')$. A conflict state is *safe*, if all decisions that could trigger UndoClear are bad:

Definition 17 (Safe conflict state) A conflict state $\langle \Gamma; E \rangle$ is *safe* if, for all first-order $_?A \in E$ with $\mathsf{level}_\Gamma(_?A) > i = \mathsf{level}_\Gamma(E \backslash \{_?A\})$, $_?A$ is bad for level i.

The key lemma simultaneously shows that CDSAT transitions reduce trails with respect to the trail ordering and keep conflict states safe.

Lemma 6 *If $\langle \Gamma; E \rangle \Longrightarrow^* \Gamma'$ and $\langle \Gamma; E \rangle$ is safe then $\Gamma' \prec \Gamma$; if $\Gamma \longrightarrow_{\mathcal{B}} \Gamma'$ then $\Gamma' \prec \Gamma$.*

Proof The first claim is proved by induction on the length of the derivation $\langle \Gamma; E \rangle \Longrightarrow^* \Gamma'$. The base case covers UndoClear, UndoDecide, and Backjump. For UndoClear we have $\langle \Gamma; E \uplus \{A\} \rangle \Longrightarrow \Gamma^{\leq m-1}$, where A is the undone first-order decision and Γ' is $\Gamma^{\leq m-1}$ with $m{=}\mathsf{level}_\Gamma(A)$. Let $i{=}\mathsf{level}_\Gamma(E)$, where $i < m$. Since $\langle \Gamma; E \uplus \{A\} \rangle$ is safe, A is bad for level i. We have: $\forall j$, $1 \leq$

$j \leq i$, $\mathsf{so}_j^{\Gamma'}=\mathsf{so}_j^{\Gamma}$; $\forall j$, $0 \leq j \leq i$, $w_j^{\Gamma'}=w_j^{\Gamma}$; $\forall j$, $0 \leq j < i$, $\mathsf{bad}_j^{\Gamma'} \leq \mathsf{bad}_j^{\Gamma}$, as removing assignments may make decisions acceptable. $\Gamma' \prec \Gamma$ follows from either $\mathsf{bad}_j^{\Gamma'} < \mathsf{bad}_j^{\Gamma}$ for some $j < i$ or $\mathsf{bad}_i^{\Gamma'} < \mathsf{bad}_i^{\Gamma}$ as A is undone. For UndoDecide we have $\langle \Gamma; E \uplus \{L\}\rangle \Longrightarrow \Gamma^{\leq m-1}, {}_?\overline{L}$. Let A' be the undone first-order decision; Γ' is $\Gamma^{\leq m-1}, {}_?\overline{L}$ with $m{=}\mathsf{level}_\Gamma(E){=}\mathsf{level}_\Gamma(L){=}\mathsf{level}_\Gamma(A')$. We have: $\forall j$, $1 \leq j \leq m-1$, $\mathsf{so}_j^{\Gamma'}=\mathsf{so}_j^{\Gamma}$; $\forall j$, $0 \leq j \leq m-1$, $w_j^{\Gamma'}=w_j^{\Gamma}$ and $\mathsf{bad}_j^{\Gamma'} \leq \mathsf{bad}_j^{\Gamma}$. $\Gamma' \prec \Gamma$ follows from either $\mathsf{bad}_j^{\Gamma'} < \mathsf{bad}_j^{\Gamma}$ for some $j \leq m-1$ as before or $\mathsf{so}_m^{\Gamma'}=0 < \mathsf{so}_m^{\Gamma}=1$ as $\overline{L}$ is Boolean while A' is first-order. For Backjump we have $\langle \Gamma; E \uplus \{L\}\rangle \Longrightarrow \Gamma^{\leq m}, {}_{E\vdash}\overline{L}$, and Γ' is $\Gamma^{\leq m}, {}_{E\vdash}\overline{L}$ with $m{=}\mathsf{level}_\Gamma(E)$. We have: $\forall j$, $1 \leq j \leq m$, $\mathsf{so}_j^{\Gamma'}=\mathsf{so}_j^{\Gamma}$; $\forall j$, $0 \leq j \leq m-1$, $w_j^{\Gamma'}=w_j^{\Gamma}$ and $\mathsf{bad}_j^{\Gamma'} \leq \mathsf{bad}_j^{\Gamma}$. $\Gamma' \prec \Gamma$ follows from either $\mathsf{bad}_j^{\Gamma'} < \mathsf{bad}_j^{\Gamma}$ for some $j \leq m-1$ as before or $w_m^{\Gamma'} < w_m^{\Gamma}$ as $\overline{L}$ is added to level m in Γ'. The induction step covers a derivation starting with Resolve unfolding ${}_{H\vdash}A$ in the conflict: $\langle \Gamma; E \uplus \{A\}\rangle \Longrightarrow \langle \Gamma; E \cup H\rangle \Longrightarrow^* \Gamma'$. We show that if $\langle \Gamma; E \uplus \{A\}\rangle$ is safe then $\langle \Gamma; E \cup H\rangle$ is safe. Then $\Gamma' \prec \Gamma$ follows by induction hypothesis. Let A' be any first-order decision in $E \cup H$ such that $\mathsf{level}_\Gamma(A') > i = \mathsf{level}_\Gamma((E \cup H)\backslash\{A'\})$. A' is not A because $A \notin E$ and $A \notin H$, and $A' \notin H$ by the side-condition of Resolve. Thus, $A' \in E$ and $i{=}\mathsf{level}_\Gamma((E \uplus \{A\})\backslash\{A'\})$. Since $\langle \Gamma; E \uplus \{A\}\rangle$ is safe, A' is bad for level i. The second claim is trivially true for Fail. For Decide we have $\Gamma \longrightarrow \Gamma, {}_?A$ and Γ' is $\Gamma, {}_?A$. Let m be the number of decisions in Γ, so that $\mathsf{level}_\Gamma(A){=}m+1$. We have: $\forall j$, $1 \leq j \leq m$, $\mathsf{so}_j^{\Gamma'}=\mathsf{so}_j^{\Gamma}$; $\forall j$, $0 \leq j \leq m$, $w_j^{\Gamma'}=w_j^{\Gamma}$. Since A is acceptable, for no level j, $0 \leq j \leq m$, A is bad. Thus, $\forall j$, $0 \leq j \leq m$, $\mathsf{bad}_j^{\Gamma'}=\mathsf{bad}_j^{\Gamma}$, and $\Gamma' \prec \Gamma$ as $\mathsf{so}_{m+1}^{\Gamma'} < 2$. For Deduce we have $\Gamma \longrightarrow \Gamma, {}_{J\vdash}L$ and Γ' is $\Gamma, {}_{J\vdash}L$. Let i be $\mathsf{level}_\Gamma(L)$. We have: $\forall j$, $1 \leq j \leq i$, $\mathsf{so}_j^{\Gamma'}=\mathsf{so}_j^{\Gamma}$; $\forall j$, $0 \leq j < i$, $w_j^{\Gamma'}=w_j^{\Gamma}$ and $\mathsf{bad}_j^{\Gamma'}=\mathsf{bad}_j^{\Gamma}$. The addition of ${}_{J\vdash}L$ gives $w_i^{\Gamma'} < w_i^{\Gamma}$, so that $\Gamma' \prec \Gamma$. For ConflictSolve we have $\Gamma \longrightarrow \Gamma'$, provided $\langle \Gamma; J \cup \{\overline{L}\}\rangle \Longrightarrow^* \Gamma'$, where $J \vdash_{\mathcal{I}_k} L$ and $\overline{L} \in \Gamma$. We prove that $\langle \Gamma; J \cup \{\overline{L}\}\rangle$ is safe. Let A be any first-order decision in J with $\mathsf{level}_\Gamma(A) > i{=}\mathsf{level}_\Gamma((J \cup \{\overline{L}\})\backslash\{A\})$. Since $\overline{L} \in \Gamma$ and $i{=}\mathsf{level}_\Gamma((J \cup \{\overline{L}\})\backslash\{A\})$ we have $\overline{L} \in (\Gamma^{\leq i})_{\mathcal{T}_k}$. Since $J \vdash_{\mathcal{I}_k} L$, A is not acceptable for $(\Gamma^{\leq i})_{\mathcal{T}_k}$, and therefore it is bad for level i. Thus, $\langle \Gamma; J \cup \{\overline{L}\}\rangle$ is safe, and $\Gamma' \prec \Gamma$ by the first claim.

Thus, we can say that a state is *CDSAT-reducible*, if a CDSAT transition rule applies, and *CDSAT-irreducible* otherwise.

Theorem 2 (Termination) *For all input assignments H, if the global basis $\mathcal{B}$ is finite, closed, and such that H is in $\mathcal{B}$, then all CDSAT derivations from H are guaranteed to halt.*

9.3 Completeness of CDSAT

Completeness requires that there is a leading theory $\mathcal{T}_1$, that its module $\mathcal{I}_1$ is complete for $\mathcal{T}_1$, and that all other modules $\mathcal{I}_k$'s, $2 \leq k \leq n$, are leading-

theory-complete. The latter property is defined in terms of leading-theory-compatibility, which involves a shared set N of terms. Definition 14 is concerned only with the relation of $\mathcal{T}_k$ and $\mathcal{I}_k$ $(2 \leq k \leq n)$ with $\mathcal{T}_1$, and therefore it instantiates N to $G(J)$ for a $\mathcal{T}_k$-assignment J. We instantiate N with the set of *shared terms* for a $\mathcal{T}_\infty$-assignment H, defined inductively as follows.

Definition 18 (Shared terms) The set of *shared terms* for an assignment H, denoted $\mathcal{V}_{\mathsf{sh}}(H)$, is the smallest set N closed under the following rules

$$\frac{(t \leftarrow \mathfrak{c}) \in H}{t \in N} \qquad \frac{u, u' \in N,\ t \in \mathsf{fv}_{\Sigma_i}(u) \cap \mathsf{fv}_{\Sigma_j}(u'),\ i \neq j}{t \in N} \qquad \frac{u \in N,\ t \in \mathsf{fv}_{\Sigma_k}(u) \setminus \mathcal{V}_\infty}{t \in N}$$

where $1 \leq i, j, k \leq n$; also, $\mathcal{V}^s_{\mathsf{sh}}(H)$ is the set of shared terms of sort s, $s \in S_\infty$.

The inductive rules add shared variables and foreign terms, with shared foreign terms added by both rules. Note that $\mathsf{fv}(\mathcal{V}_{\mathsf{sh}}(H)) = \mathsf{fv}(H)$ holds.

Example 11 For problem P of Example 3, the base of the inductive construction of $\mathcal{V}_{\mathsf{sh}}(P)$ is that $f(\mathsf{select}(\mathsf{store}(a, i, v), j)) \simeq w$, $(w{+}\frac{1}{2})^2 \simeq f(u)$, $i \simeq j$, and $u \simeq v$ are shared. The third rule adds $f(\mathsf{select}(\mathsf{store}(a, i, v), j))$ and $f(u)$ as they are Σ_{RA}-foreign, $(w{+}\frac{1}{2})^2$ as it is Σ_{EUF}-foreign. The second rule adds w, as it is a free Σ_{EUF}-variable and Σ_{RA}-variable, and u and v, as they are free Σ_{EUF}-variables and Σ_{Arr}-variables. The third rule adds $\mathsf{select}(\mathsf{store}(a, i, v), j)$ as it is Σ_{EUF}-foreign. In contrast, $\mathsf{store}(a, i, v)$, a, i, j, and $w{+}\frac{1}{2}$ are not shared: the first four are seen only by Arr and $w{+}\frac{1}{2}$ only by RA.

Definition 19 (Model-describing assignment) Let $\mathcal{T}_1$ be the leading theory. An assignment H is *model-describing* if $H_{\mathcal{T}_1}$ is consistent with $\mathcal{T}_1$ and for all k, $2 \leq k \leq n$, $H_{\mathcal{T}_k}$ is leading-theory-compatible with $\mathcal{T}_k$ sharing $\mathcal{V}_{\mathsf{sh}}(H)$.

For completeness, we require the global basis $\mathcal{B}$ to be *stable*, meaning that for all k, $1 \leq k \leq n$, $\mathsf{basis}_k(\mathcal{B}) \subseteq \mathcal{B}$. A stable $\mathcal{B}$ is also closed: for all k, $1 \leq k \leq n$, $\mathcal{B} \subseteq \mathsf{basis}_k(\mathcal{B})$ holds by extensiveness of basis_k, which, together with stability, implies $\mathsf{basis}_k(\mathcal{B}) = \mathcal{B}$, so that $\mathcal{B}$ is closed by the closure property of basis_k (see Definition 9).

Theorem 3 *For all input assignments H, if the global basis $\mathcal{B}$ is stable and H is in $\mathcal{B}$, whenever a CDSAT derivation from H halts in a state Γ other than* unsat*, Γ is model-describing.*

Proof Since $\mathcal{B}$ is closed, by Lemma 5, all trails Γ derived by CDSAT from H are in $\mathcal{B}$. We prove the claim by showing that all states $\langle \Gamma; E \rangle$ with $\mathsf{level}_\Gamma(E) > 0$ and all states Γ such that Γ is not model-describing are CDSAT-reducible. Consider a conflict state $\langle \Gamma; E \rangle$ with $i = \mathsf{level}_\Gamma(E) > 0$. Since $i > 0$, $E \neq \emptyset$. If there is only one assignment A of level i in E, either Backjump or UndoClear apply. Otherwise, let A and A' be two assignments of level i in E. If Resolve applies to either of them, we are done. Otherwise, each of them is either the decision of level i, or a Boolean justified assignment L whose justification contains the decision of level i, which is first-order (otherwise Resolve applies).

Since there is only one decision of level i, either A or A' is such an L, to which UndoDecide applies. Consider next a state Γ. If it is not model-describing, there is a k, $1 \leq k \leq n$, such that either $k = 1$ and $\Gamma_{\mathcal{T}_1}$ is not consistent with $\mathcal{T}_1$, or $k > 1$ and $\Gamma_{\mathcal{T}_k}$ is not leading-theory-compatible with $\mathcal{T}_k$ sharing $\mathcal{V}_{\mathsf{sh}}(\Gamma)$. For this k, either $\Gamma_{\mathcal{T}_k} \not\subseteq \Gamma$ or $\Gamma_{\mathcal{T}_k} \subseteq \Gamma$. If $\Gamma_{\mathcal{T}_k} \not\subseteq \Gamma$, there is an $L = ((t_1 \simeq_s t_2) \leftarrow \mathfrak{b})$, $s \neq \mathsf{prop}$, such that $L \in \Gamma_{\mathcal{T}_k}$, but $L \notin \Gamma$. Since every $\mathcal{I}_k$ features the equality inference rules, there is an inference $J \vdash_{\mathcal{I}_k} L$ for some $J \subseteq \Gamma$. If $\overline{L} \in \Gamma$, either Fail or ConflictSolve applies. If $\overline{L} \notin \Gamma$, Deduce applies, provided $(t_1 \simeq_s t_2) \in \mathcal{B}$, which is the case: $t_1, t_2 \in \Gamma$ hence $t_1, t_2 \in \mathcal{B}$; by extensiveness of basis_k, $t_1, t_2 \in \mathsf{basis}_k(\mathcal{B})$; by closure of basis_k, $(t_1 \simeq t_2) \in \mathsf{basis}_k(\mathcal{B})$; and by stability of $\mathcal{B}$, $(t_1 \simeq t_2) \in \mathcal{B}$. If $\Gamma_{\mathcal{T}_k} \subseteq \Gamma$, then $\Gamma_{\mathcal{T}_k}$ is plausible since Γ is. By completeness of $\mathcal{I}_k$ (see Definition 12 for $k=1$ and Definition 14 for $k>1$), $\mathcal{I}_k$ can expand $\Gamma_{\mathcal{T}_k}$ (see Definition 10). If $\mathcal{I}_k$ can expand $\Gamma_{\mathcal{T}_k}$ with an acceptable $\mathcal{T}_k$-assignment, Decide applies. If $\mathcal{I}_k$ can expand $\Gamma_{\mathcal{T}_k}$ with an inference $J \vdash_{\mathcal{I}_k} L$, then $J \subseteq \Gamma_{\mathcal{T}_k} \subseteq \Gamma$ and L is a Boolean assignment for a formula in $\mathsf{basis}_k(\Gamma_{\mathcal{T}_k}) \subseteq \mathsf{basis}_k(\Gamma) \subseteq \mathsf{basis}_k(\mathcal{B}) \subseteq \mathcal{B}$ (by monotonicity of basis_k and stability of $\mathcal{B}$). If $\overline{L} \notin \Gamma$, Deduce applies, if $\overline{L} \in \Gamma$, either Fail or ConflictSolve applies.

The following technical lemma is used by the next theorem.

Lemma 7 *For all assignments H, $\mathsf{fv}(H) \subseteq \bigcup_{k=1}^{n} \mathsf{fv}_{\Sigma_k}(\mathcal{V}_{\mathsf{sh}}(H))$.*

Proof By way of contradiction, suppose that there is a variable x such that $x \in \mathsf{fv}(H)$, but $x \notin \bigcup_{k=1}^{n} \mathsf{fv}_{\Sigma_k}(\mathcal{V}_{\mathsf{sh}}(H))$. From $x \in \mathsf{fv}(H)$ it follows that x occurs in some term t_0 such that $(t_0 \leftarrow \mathfrak{c}) \in H$. By structural induction on t_0, there is a term $t \in \mathsf{fv}_{\Sigma_k}(t_0)$, for some k, $1 \leq k \leq n$, such that $x \trianglelefteq t \trianglelefteq t_0$. By the first rule of Definition 18, $t_0 \in \mathcal{V}_{\mathsf{sh}}(H)$, and therefore the set $Y = \{t \mid x \trianglelefteq t,\ t \in \bigcup_{k=1}^{n} \mathsf{fv}_{\Sigma_k}(\mathcal{V}_{\mathsf{sh}}(H))\}$ is not empty. For all terms $t \in Y$, let $d(t)$ denote the distance between the root position of t and the position of the leftmost occurrence of x in t. Let $t \in Y$ be such that $d(t)$ is minimal. If $x = t$ we have a contradiction and we are done. If $x \triangleleft t$, then certainly $t \notin \mathcal{V}_\infty$. By definition of Y, $t \in \mathsf{fv}_{\Sigma_k}(u)$ for some $u \in \mathcal{V}_{\mathsf{sh}}(H)$ and k, $1 \leq k \leq n$. In words, t is a foreign subterm of u. Therefore, $t \in \mathcal{V}_{\mathsf{sh}}(H)$ by the third rule of Definition 18. The fact that $x \triangleleft t$ also means that t has a root symbol $g \in F_j$ for some j, $1 \leq j \leq n$, and there is a $t' \in \mathsf{fv}_{\Sigma_j}(t)$ such that $x \trianglelefteq t' \triangleleft t$ (if there is no Σ_j-foreign symbol on the path from the root symbol to x in t, then $t' = x$). Therefore we have $t' \in Y$ with $d(t') < d(t)$, contradicting the assumption that $d(t)$ is minimal.

Moreover, for all k, $1 \leq k \leq n$, $\mathsf{fv}_{\Sigma_k}(H_{\mathcal{T}_k}) = \mathsf{fv}_{\Sigma_k}(H) \subseteq \mathsf{fv}_{\Sigma_k}(\mathcal{V}_{\mathsf{sh}}(H))$ $(*)$, so that $\mathsf{fv}_{\Sigma_k}(H_{\mathcal{T}_k} \cup \mathcal{V}_{\mathsf{sh}}(H)) = \mathsf{fv}_{\Sigma_k}(\mathcal{V}_{\mathsf{sh}}(H))$ $(**)$.

Theorem 4 *If an assignment H is model-describing, there exists a $\mathcal{T}_\infty^+[\mathsf{fv}(H)]$-model $\mathcal{M}$ such that $\mathcal{M} \models^G H$.*

Proof We structure the proof in several steps.

1. *Existence of a leading-theory model $\mathcal{M}_1$*: by Definitions 19 and 11, there exists a $\mathcal{T}_1^+[\mathcal{V}_1]$-model $\mathcal{M}_1'$, with $\mathsf{fv}_{\Sigma_1}(H) \subseteq \mathcal{V}_1$, such that $\mathcal{M}_1' \models H_{\mathcal{T}_1}$. Since $\mathsf{fv}_{\Sigma_1}(H) \subseteq \mathsf{fv}_{\Sigma_1}(\mathcal{V}_{\mathsf{sh}}(H))$ by $(*)$, we pick arbitrary elements in the

domains of $\mathcal{M}'_1$ to interpret terms in $\mathsf{fv}_{\Sigma_1}(\mathcal{V}_{\mathsf{sh}}(H)) \setminus \mathcal{V}_1$, if any, and we extend $\mathcal{M}'_1$ into a $\mathcal{T}_1^+[\mathsf{fv}_{\Sigma_1}(\mathcal{V}_{\mathsf{sh}}(H))]$-model $\mathcal{M}_1$ such that $\mathcal{M}_1 \models H_{\mathcal{T}_1}$.

2. *Existence of the other $\mathcal{T}_k$-models $\mathcal{M}_k$*: by Definitions 19 and 13, $\forall k$, $2 \leq k \leq n$, there exists a $\mathcal{T}_k^+[\mathcal{V}_k]$-model $\mathcal{M}_k$, with $\mathsf{fv}_{\Sigma_k}(\mathcal{V}_{\mathsf{sh}}(H)) \subseteq \mathcal{V}_k$ (see $(**)$), such that (i) $\mathcal{M}_k \models H_{\mathcal{T}_k}$, (ii) for all $s \in S_k$, $|s^{\mathcal{M}_k}| = |s^{\mathcal{M}_1}|$, and (iii) for all $s \in S_k$ and $u, u' \in \mathcal{V}_{\mathsf{sh}}^s(H)$, $\mathcal{M}_k(u) = \mathcal{M}_k(u')$ if and only if $\mathcal{M}_1(u) = \mathcal{M}_1(u')$.
3. *Bijection between any $\mathcal{M}_k$ and $\mathcal{M}_1$*: $\forall k$, $1 \leq k \leq n$, and for all $s \in S_k$, we establish a bijection $\phi_k^s : s^{\mathcal{M}_k} \to s^{\mathcal{M}_1}$ such that ϕ_1^s is identity, and $\forall k$, $2 \leq k \leq n$, $\phi_k^s(\mathcal{M}_k(t)) = \mathcal{M}_1(t)$ for all shared terms $t \in \mathcal{V}_{\mathsf{sh}}^s(H)$ of sort s.
4. *Construction of a $\mathcal{T}_\infty^+[\mathsf{fv}(H)]$-model $\mathcal{M}$*: first, $\mathcal{M}$ adopts the domains of $\mathcal{M}_1$ and interprets all sorts and shared variables as $\mathcal{M}_1$ does:
   (a) For all sorts $s \in S_\infty$: $s^{\mathcal{M}} = s^{\mathcal{M}_1}$;
   (b) For all variables $x \in \mathsf{fv}(H)$ such that $x \in \mathcal{V}_{\mathsf{sh}}(H)$: $x^{\mathcal{M}} = x^{\mathcal{M}_1}$.

   Second, $\mathcal{M}$ interprets everything else as in the appropriate $\mathcal{M}_k$, using the bijection ϕ_k^s or its inverse $(\phi_k^s)^{-1}$ to reach elements in its domains:
   (c) For a non-Boolean $\mathcal{T}_k^+$-value $\mathfrak{c}$ of sort s: $\mathfrak{c}^{\mathcal{M}} = \phi_k^s(\mathfrak{c}^{\mathcal{M}_k})$;
   (d) For all variables $x \in \mathsf{fv}^s(H)$ such that $x \notin \mathcal{V}_{\mathsf{sh}}(H)$: $x^{\mathcal{M}} = \phi_k^s(x^{\mathcal{M}_k})$, for the unique k, $1 \leq k \leq n$, for which $x \in \mathsf{fv}_{\Sigma_k}(\mathcal{V}_{\mathsf{sh}}(H))$; such a k exists by Lemma 7, and it is unique, otherwise $x \in \mathcal{V}_{\mathsf{sh}}(H)$ by Definition 18;
   (e) For all non-equality symbols $f : (s_1 \times \cdots \times s_m) \to s$ $(m \geq 0)$ in signature Σ_k, where k is unique as the theories are disjoint:

$$f^{\mathcal{M}}(a_1, \ldots, a_m) = \phi_k^s(f^{\mathcal{M}_k}((\phi_k^{s_1})^{-1}(a_1), \ldots, (\phi_k^{s_m})^{-1}(a_m))).$$

5. *$\mathcal{M}$ and $\mathcal{M}_k$ agree on terms, if they agree on their shared free Σ_k-variables*: $\forall k$, $1 \leq k \leq n$, if t is a term of sort $s \in S_k$ such that:
   (h1) Its free Σ_k-variables occur in shared terms: $\mathsf{fv}_{\Sigma_k}(t) \subseteq \mathsf{fv}_{\Sigma_k}(\mathcal{V}_{\mathsf{sh}}(H))$, and
   (h2) $\mathcal{M}$ and $\mathcal{M}_k$ agree on the shared free Σ_k-variables of t: for all sorts $r \in S_k$ and all terms $u \in \mathsf{fv}_{\Sigma_k}^r(t) \cap \mathcal{V}_{\mathsf{sh}}^r(H)$ it holds that $\mathcal{M}(u) = \phi_k^r(\mathcal{M}_k(u))$,

   then $\mathcal{M}(t) = \phi_k^s(\mathcal{M}_k(t))$. The proof is by structural induction.
   – If $t \in \mathsf{fv}_{\Sigma_k}(t) \cap \mathcal{V}_{\mathsf{sh}}(H)$, the claim holds by (h2).
   – If $t \in \mathsf{fv}_{\Sigma_k}(t) \setminus \mathcal{V}_{\mathsf{sh}}(H)$, then by (h1) we get $t \in \mathsf{fv}_{\Sigma_k}(\mathcal{V}_{\mathsf{sh}}(H))$, and $t \in \mathcal{V}_\infty$ must hold, otherwise from $t \in \mathsf{fv}_{\Sigma_k}(\mathcal{V}_{\mathsf{sh}}(H))$, the third rule of Definition 18 would conclude $t \in \mathcal{V}_{\mathsf{sh}}(H)$. Then from $t \in \mathsf{fv}_{\Sigma_k}(\mathcal{V}_{\mathsf{sh}}(H)))$ and $t \in \mathcal{V}_\infty$, we have $t \in \mathsf{fv}(\mathcal{V}_{\mathsf{sh}}(H)) = \mathsf{fv}(H)$, and by Item (d) in the construction of $\mathcal{M}$ we have $\mathcal{M}(t) = \phi_k^s(\mathcal{M}_k(t))$.
   – If t is a term $f(t_1, \ldots, t_m)$ with $f : (s_1 \times \cdots \times s_m) \to s$ $(m \geq 0)$ from F_k, for some k, $1 \leq k \leq n$, then Item (e) in the construction of $\mathcal{M}$ gives

$$\mathcal{M}(f(t_1, \ldots, t_m)) = \phi_k^s(f^{\mathcal{M}_k}((\phi_k^{s_1})^{-1}(\mathcal{M}(t_1)), \ldots, (\phi_k^{s_m})^{-1}(\mathcal{M}(t_m)))).$$

     The induction hypothesis is that $\forall i$, $1 \leq i \leq m$, $\mathcal{M}(t_i) = \phi_k^{s_i}(\mathcal{M}_k(t_i))$, so that $(\phi_k^{s_i})^{-1}(\mathcal{M}(t_i)) = \mathcal{M}_k(t_i)$. It follows that

$$\mathcal{M}(f(t_1, \ldots, t_m)) = \phi_k^s(f^{\mathcal{M}_k}(\mathcal{M}_k(t_1), \ldots, \mathcal{M}_k(t_m))) = \phi_k^s(\mathcal{M}_k(t)).$$

6. *$\mathcal{M}$ and $\mathcal{M}_1$ agree on shared terms*: for all $t \in \mathcal{V}_{\mathsf{sh}}(H)$ it holds that $\mathcal{M}(t) = \mathcal{M}_1(t)$. The proof is by structural induction. If $t \in \mathcal{V}_\infty$ then by Item (b) in the construction of $\mathcal{M}$ we have $\mathcal{M}(t) = \mathcal{M}_1(t)$. For the induction step, assume t is a term with root symbol $f \in F_k$, for some k, $1 \leq k \leq n$, with

arity m ($m \geq 0$) and output sort s. We prove that t satisfies (h1) and (h2). (h1) follows from $t \in \mathcal{V}_{\mathsf{sh}}(H)$. For (h2), consider a $u \in \mathsf{fv}^r_{\Sigma_k}(t) \cap \mathcal{V}^r_{\mathsf{sh}}(H)$ for any $r \in S_k$. Since $f \in F_k$, it is $u \lhd t$. Since $u \lhd t$, by induction hypothesis $\mathcal{M}(u)=\mathcal{M}_1(u)$. From $u \in \mathcal{V}^r_{\mathsf{sh}}(H)$ it follows by definition of ϕ that $\mathcal{M}_1(u)=\phi^r_k(\mathcal{M}_k(u))$. By transitivity, $\mathcal{M}(u)=\phi^r_k(\mathcal{M}_k(u))$. Then, by Part (5) of this proof, $\mathcal{M}(t)=\phi^s_k(\mathcal{M}_k(t))$. Since $t \in \mathcal{V}^s_{\mathsf{sh}}(H)$, by definition of ϕ, we have $\mathcal{M}_1(t)=\phi^s_k(\mathcal{M}_k(t))$, hence $\mathcal{M}(t)=\mathcal{M}_1(t)$ by transitivity.

7. *$\mathcal{M}$ and $\mathcal{M}_k$ agree on terms in extended signatures*: $\forall k$, $1 \leq k \leq n$, for all $\Sigma^+_k[\mathsf{fv}_{\Sigma_k}(\mathcal{V}_{\mathsf{sh}}(H))]$-terms t of sort $s \in S_k$, it holds that $\mathcal{M}(t) = \phi^s_k(\mathcal{M}_k(t))$. As before, the proof is by structural induction.
   - If t is a free Σ^+_k-variable in $\mathsf{fv}_{\Sigma_k}(\mathcal{V}_{\mathsf{sh}}(H))$, the result follows from Part (5) of this proof provided (h1) and (h2) hold for t. (h1) follows from $\mathsf{fv}_{\Sigma_k}(t) = \{t\} \subseteq \mathsf{fv}_{\Sigma_k}(\mathcal{V}_{\mathsf{sh}}(H))$. For (h2), by applying Parts (6) and (3) of this proof to any term $u \in \mathsf{fv}^r_{\Sigma_k}(t) \cap \mathcal{V}^r_{\mathsf{sh}}(H)$ with $r \in S_k$, we get $\mathcal{M}(u) = \mathcal{M}_1(u) = \phi^r_k(\mathcal{M}_k(u))$.
   - If t is a non-Boolean $\mathcal{T}^+_k$-value, then by Item (c) in the construction of $\mathcal{M}$, we have $\mathcal{M}(t) = \phi^s_k(\mathcal{M}_k(t))$.
   - If t is a term $f(t_1, \ldots, t_m)$ with $f : (s_1 \times \cdots \times s_m) \to s$ ($m \geq 0$) in F_k, for some k, $1 \leq k \leq n$, then by Item (e) in the construction of $\mathcal{M}$
   $$\mathcal{M}(f(t_1, \ldots, t_m)) = \phi^s_k(f^{\mathcal{M}_k}((\phi^{s_1}_k)^{-1}(\mathcal{M}(t_1)), \ldots, (\phi^{s_m}_k)^{-1}(\mathcal{M}(t_m)))).$$
   The induction hypothesis is that $\forall i$, $1 \leq i \leq m$, $\mathcal{M}(t_i) = \phi^{s_i}_k(\mathcal{M}_k(t_i))$, so that $(\phi^{s_i}_k)^{-1}(\mathcal{M}(t_i)) = \mathcal{M}_k(t_i)$. It follows that
   $$\mathcal{M}(f(t_1, \ldots, t_m)) = \phi^s_k(f^{\mathcal{M}_k}(\mathcal{M}_k(t_1)), \ldots, \mathcal{M}_k(t_1)) = \phi^s_k(\mathcal{M}_k(t)).$$
8. *Global endorsement*: we show that $\mathcal{M} \models^G H$ by showing that for all $(u \leftarrow \mathfrak{c}) \in H_{\mathcal{T}_\infty}$ it holds that $\mathcal{M}(u)=\mathfrak{c}^{\mathcal{M}}$. For all $(u \leftarrow \mathfrak{c}) \in H_{\mathcal{T}_\infty}$, either $(u \leftarrow \mathfrak{c}) \in H$, or u is an equality $t_1 \simeq_s t_2$ and $\mathfrak{c}$ is a Boolean value $\mathfrak{b}$. If $(u \leftarrow \mathfrak{c}) \in H$, then $\mathfrak{c}$ is a $\mathcal{T}^+_k$-value for some k, $1 \leq k \leq n$, and $(u \leftarrow \mathfrak{c}) \in H_{\mathcal{T}_k}$: the assigned value determines to which $\mathcal{T}_k$-view $u \leftarrow \mathfrak{c}$ belongs. If u is an equality $t_1 \simeq_s t_2$, the sort s of the equality determines to which $\mathcal{T}_k$-view $u \leftarrow \mathfrak{b}$ belongs. Sort s must belong to at least one of the signatures: say that $s \in S_k$ for some k, $1 \leq k \leq n$; then $(u \leftarrow \mathfrak{b}) \in H_{\mathcal{T}_k}$. Either way, by Part (1) of this proof, if $k = 1$, or by Part (2) of this proof, if $2 \leq k \leq n$, we have (i) $\mathcal{M}_k(u) = \mathfrak{c}^{\mathcal{M}_k}$. Let r be the sort of term u. By Part (7) of this proof, we have (ii) $\mathcal{M}(u) = \phi^r_k(\mathcal{M}_k(u))$. By Item (c) of Part (4) of this proof, it is (iii) $\phi^r_k(\mathfrak{c}^{\mathcal{M}_k}) = \mathfrak{c}^{\mathcal{M}}$. Thus, by chaining (ii), (i), and (iii), one gets $\mathcal{M}(u) = \phi^r_k(\mathcal{M}_k(u)) = \phi^r_k(\mathfrak{c}^{\mathcal{M}_k}) = \mathfrak{c}^{\mathcal{M}}$, which means that $\mathcal{M}$ endorses the assignment.

Theorem 5 (Completeness) *For all input assignments H, if the global basis $\mathcal{B}$ is stable and H is in $\mathcal{B}$, whenever a CDSAT derivation from H halts in a state Γ other than* unsat*, there exists a $\mathcal{T}^+_\infty[\mathsf{fv}(\Gamma)]$-model $\mathcal{M}$ such that $\mathcal{M} \models^G \Gamma$ hence $\mathcal{M} \models^G H$ (input assignments never quit the trail).*

Theorems 3 and 5 cover a worst-case scenario as CDSAT can reach a model-describing Γ before reaching a CDSAT-irreducible state, and $\mathcal{T}_k$-procedures can notify the center as soon as Γ is leading-theory-compatible with $\mathcal{T}_k$.

10 Discussion

We conclude with related work, directions for future research, and a summary.

10.1 Relation of CDSAT with DPLL($\mathcal{T}$) and Equality Sharing

DPLL($\mathcal{T}$) with equality sharing [22,2,18] is the special case of CDSAT that CDSAT reduces to if the combination of theories only involves the theory Bool and stably infinite theories $\mathcal{T}_1, \ldots, \mathcal{T}_n$ with black-box modules $\mathcal{I}_1, \ldots \mathcal{I}_n$. The leading theory is $\mathcal{T}_{\mathbb{N}}$ (see Section 8), and $\mathcal{I}_{\mathsf{Bool}}, \mathcal{I}_1, \ldots \mathcal{I}_n$ are leading-theory complete. Since only sort prop is public, all decisions are Boolean decisions about atoms from the input set of clauses S or relevant equalities. The set of shared terms is the set $\mathcal{V}_{\mathsf{sh}}(S)$ of shared terms for S, because no new terms are shared by black-box theory modules.

CDSAT simulates CDCL by making decisions for atoms in the trail Γ and using Deduce for Boolean Clausal Propagation (BCP), while each $\mathcal{T}_k$-module fires its inference rule if Boolean assignments in Γ are $\mathcal{T}_k$-inconsistent. CDSAT subsumes equality sharing by making decisions about equalities between terms in $\mathcal{V}_{\mathsf{sh}}(S)$, and applying equality inferences to propagate entailed equalities and inequalities. In equality sharing, a set of equalities and inequalities that determines which shared variables are equal and which are not is called *arrangement* (e.g., [7], Ch. 10). CDSAT builds incrementally a partial arrangement of terms in $\mathcal{V}_{\mathsf{sh}}(S)$, and tries by backjumping different (partial or full) arrangements until it finds one for which Γ is *model-describing*: Γ contains a (possibly partial) Boolean assignment for atoms in S and a (possibly partial) arrangement for terms in $\mathcal{V}_{\mathsf{sh}}(S)$, such that S is satisfied, and there are $\mathcal{T}_k$-models of the literals in Γ that agree with the arrangement in Γ and all its completions. Since Γ can be model-describing without a full arrangement, Theorem 4 generalizes the Nelson-Oppen completeness theorem (e.g., [7], Ch. 10): if the arrangement in Γ is complete, the two theorems concur; if it is partial, Theorem 4 still holds, as the $\mathcal{T}_k$'s do not care about unassigned equalities between terms in $\mathcal{V}_{\mathsf{sh}}(S)$. For the same reason, CDSAT can emulate the *sharing is caring* technique [14], as *care functions* can be employed to focus CDSAT on decisions about cared-for equalities. The definition of a care function involves the preservation of satisfiability for all completions of a partial arrangement, an idea captured in CDSAT by the notion of leading-theory-compatibility.

10.2 Relation of CDSAT with MCSAT

CDSAT generalizes MCSAT [9,15,23,13] to theory combination. MCSAT [9] is the special case of CDSAT that CDSAT reduces to if the combination only involves the theory Bool and another theory $\mathcal{T}_1$ with a conflict-driven satisfiability procedure, hence CDSAT modules $\mathcal{I}_{\mathsf{Bool}}$ and $\mathcal{I}_1$. MCSAT stores clauses in a set, so that an MCSAT state is given by a pair $\langle \mathit{trail}, \mathit{set} \rangle$. An MCSAT

trail is similar to a CDSAT trail where justified assignments are generated only by BCP or conflict analyses where $\mathcal{T}_1$-inferences explain $\mathcal{T}_1$-conflicts.

The MCSAT $\mathcal{T}_1$-plugin features an *evaluation* mechanism mapping trail Γ and literal L to a truth value $v[\Gamma](L)$ defined if Γ assigns values to all variables of L. The CDSAT module $\mathcal{I}_1$ captures these evaluations with the inference rule $J \vdash_{\mathcal{T}_1} L'$, where J is the restriction of Γ to the variables of L, and L' is L or $\overline{L}$ depending on whether $v[J](L)$ is true or false. The MCSAT $\mathcal{T}_1$-plugin also provides an *explanation function* mapping L and Γ to a $\mathcal{T}_1$-valid clause $\mathsf{explain}(L, \Gamma) = L_1 \vee \cdots \vee L_m \vee L$, such that $v[\Gamma](L_i) = \mathsf{false}$, $\forall i$, $1 \leq i \leq m$. This function is applied only if all $\mathcal{T}_1$-models endorsing $\Gamma_{\mathcal{T}_1}$ (the $\mathcal{T}_1$-view of Γ) endorse L. In practice, the system checks that the current candidate $\mathcal{T}_1$-model endorses L as MCSAT focuses on theories with *model-constructing* plugins. The CDSAT module $\mathcal{I}_1$ captures the clauses produced by the $\mathcal{T}_1$-explanation function with inferences $\overline{L_1} \ldots, \overline{L_m} \vdash_{\mathcal{T}_1} L$.

In this special case where only Bool and $\mathcal{T}_1$ are combined, the MCSAT and CDSAT transition systems have only few differences. MCSAT can make Boolean decisions about any formula in the local basis of $\mathcal{I}_1$, while CDSAT only decides *relevant* ones. MCSAT applies Deduce explicitly for BCP and implicitly for evaluation inferences: given input clause $C = (x{<}0) \vee (y{<}0)$, a CDSAT trail $_{\emptyset}\vdash C, {}_{?}x{\leftarrow}1, {}_{\{x\leftarrow 1\}}\vdash(\overline{x{<}0}), {}_{\{\overline{x<0}, C\}}\vdash(y{<}0)$ corresponds to the MCSAT state $\langle [\![(x{\leftarrow}1), (C \rightarrow (y{<}0))]\!], \{C\}\rangle$ where the falsity of $x{<}0$ is implicit and only C justifies $y{<}0$. MCSAT applies neither Fail nor $\mathsf{ConflictSolve}$ with evaluation inferences, so that conflicts are made of Boolean assignments $C_1, \ldots, C_p, L_1, \ldots, L_m$ where $C_1, \ldots, C_p$ are clauses of level 0. MCSAT presents such a conflict as the conflict clause $C = \overline{L_1} \vee \cdots \vee \overline{L_m}$, leaving implicit the 0-level premises $C_1, \ldots, C_p$ of C. Clause learning and forgetting [9] can be added to CDSAT as already done for learning [6].

The MCSAT combination of Bool, LRA, and EUF [15] is a special case of CDSAT: consider a variant of the CDSAT-MCSAT relation described above with an EUF-plugin added to MCSAT and $\mathcal{I}_{\mathsf{EUF}}$ added to CDSAT. MCSAT was implemented in Z3 [9], CVC4 [15], and Yices [11]. As all these solvers are DPLL($\mathcal{T}$)-based, either MCSAT does not interact with the rest (as in Yices), or it interacts as a black box. CDSAT provides a theoretical foundation for the cooperation of MCSAT and DPLL($\mathcal{T}$) components. The MCSAT implementation in Yices organizes a generic collaboration of several conflict-driven theory plugins, for which CDSAT provides theoretical underpinnings. Yices implements a variant of MCSAT that allows the propagation of first-order assignments [13]: the CDSAT extension allowing theory modules to derive first-order assignments is future work.

10.3 Summary of Contributions and Directions for Future Research

We presented a new framework for SMT named CDSAT, which solves the hitherto open problem of integrating conflict-driven and black-box theory procedures. CDSAT lifts CDCL to *satisfiability modulo multiple theories* by com-

bining theory inference systems termed *theory modules*. We presented several theory modules, including one for *arrays* which is the *first* integration of this theory in a conflict-driven combination. Since it accepts also input first-order assignments, CDSAT solves a class of problems, called SMA for *Satisfiability Modulo Assignments*, that extends SMT. We proved that CDSAT is *sound*, *complete*, and *terminating*, assuming a finite global basis for new terms.

In a forthcoming journal article, we provide local bases and completeness proofs for the theory modules presented here, we show how to construct a global basis from local ones, and we extend CDSAT with *lemma learning* and *proof generation* [6]. Directions for future work include implementing CDSAT, for which preparatory work is ongoing [3], heuristics for decisions, theory search plans, and efficient techniques to detect the acceptability of assignments and the applicability of theory inferences.

Acknowledgements We thank Dejan Jovanović for fruitful discussions. Part of this work was done when the first author was visiting the Computer Science Lab of SRI International, whose support is greatly appreciated. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of NSF, DARPA, or the U.S. Government.

References

1. Clark W. Barrett, David L. Dill, and Aaron Stump. A generalization of Shostak's method for combining decision procedures. In Alessandro Armando, editor, *Proceedings of the Fourth International Workshop on Frontiers of Combining Systems (FroCoS)*, volume 2309 of *Lecture Notes in Artificial Intelligence*, pages 132–146. Springer, 2002.
2. Clark W. Barrett, Robert Nieuwenhuis, Alberto Oliveras, and Cesare Tinelli. Splitting on demand in SAT modulo theories. In Miki Hermann and Andrei Voronkov, editors, *Proceedings of the Thirteenth International Conference on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR)*, volume 4246 of *Lecture Notes in Artificial Intelligence*, pages 512–526. Springer, 2006.
3. François Bobot, Stéphane Graham-Lengrand, Bruno Marre, and Guillaume Bury. Centralizing equality reasoning in MCSAT. Presented at the Sixteenth International Workshop on Satisfiability Modulo Theories (SMT) at the Ninth International Joint Conference on Automated Reasoning (IJCAR), July 2018.
4. Maria Paola Bonacina. On conflict-driven reasoning. In Natarajan Shankar and Bruno Dutertre, editors, *Proceedings of the Sixth Workshop on Automated Formal Methods (AFM) at the Ninth NASA Formal Methods Symposium (NFM)*, volume 5 of *Kalpa Publications*, pages 31–49. EasyChair, 2018.
5. Maria Paola Bonacina, Stéphane Graham-Lengrand, and Natarajan Shankar. Satisfiability modulo theories and assignments. In Leonardo de Moura, editor, *Proceedings of the Twenty-Sixth Conference on Automated Deduction (CADE)*, volume 10395 of *Lecture Notes in Artificial Intelligence*, pages 42–59. Springer, 2017.
6. Maria Paola Bonacina, Stéphane Graham-Lengrand, and Natarajan Shankar. Proofs in conflict-driven theory combination. In June Andronick and Amy Felty, editors, *Proceedings of the Seventh ACM International Conference on Certified Programs and Proofs (CPP)*, pages 186–200. ACM Press, 2018.
7. Aaron R. Bradley and Zohar Manna. *The Calculus of Computation - Decision Procedures with Applications to Verification*. Springer, 2007.
8. Robert Brummayer and Armin Biere. Lemmas on demand for the extensional theory of arrays. *Journal on Satisfiability, Boolean Modeling and Computation*, 6:165–201, 2009.

9. Leonardo de Moura and Dejan Jovanović. A model-constructing satisfiability calculus. In Roberto Giacobazzi, Josh Berdine, and Isabella Mastroeni, editors, *Proceedings of the Fourteenth International Conference on Verification, Model Checking and Abstract Interpretation (VMCAI)*, volume 7737 of *Lecture Notes in Computer Science*, pages 1–12. Springer, 2013.
10. Leonardo de Moura and Grant Olney Passmore. Exact global optimization on demand (presentation only). In Silvio Ghilardi, Viorica Sofronie-Stokkermans, and Ashish Tiwari, editors, *Notes of the Third Workshop on Automated Deduction: Decidability, Complexity, Tractability (ADDCT)*, pages 50–50, 2013. Available at `https://userpages.uni-koblenz.de/~sofronie/addct-2013/`, last seen on December 19, 2018.
11. Bruno Dutertre. Yices 2.2. In Armin Biere and Roderick Bloem, editors, *Proceedings of the Twenty-Sixth International Conference on Computer Aided Verification (CAV)*, volume 8559 of *Lecture Notes in Computer Science*, pages 737–744. Springer, 2014.
12. Harald Ganzinger, Harald Rueß, and Natarajan Shankar. Modularity and refinement in inference systems. Technical Report CSL-SRI-04-02, CSL, SRI International, Menlo Park, CA, USA, 2004.
13. Dejan Jovanović. Solving nonlinear integer arithmetic with MCSAT. In Ahmed Bouajjani and David Monniaux, editors, *Proceedings of the Eighteenth International Conference on Verification, Model Checking and Abstract Interpretation (VMCAI)*, volume 10145 of *Lecture Notes in Computer Science*, pages 330–346. Springer, 2017.
14. Dejan Jovanović and Clark W. Barrett. Sharing is caring: combination of theories. In Cesare Tinelli and Viorica Sofronie-Stokkermans, editors, *Proceedings of the Eighth International Symposium on Frontiers of Combining Systems (FroCoS)*, volume 6989 of *Lecture Notes in Artificial Intelligence*, pages 195–210. Springer, 2011.
15. Dejan Jovanović, Clark W. Barrett, and Leonardo de Moura. The design and implementation of the model-constructing satisfiability calculus. In Barbara Jobstman and Sandip Ray, editors, *Proceedings of the Thirteenth Conference on Formal Methods in Computer Aided Design (FMCAD)*. ACM and IEEE, 2013.
16. Dejan Jovanović and Leonardo de Moura. Cutting to the chase: solving linear integer arithmetic. In Nikolaj Bjørner and Viorica Sofronie-Stokkermans, editors, *Proceedings of the Twenty-Third International Conference on Automated Deduction (CADE)*, volume 6803 of *Lecture Notes in Artificial Intelligence*, pages 338–353. Springer, 2011.
17. Dejan Jovanović and Leonardo de Moura. Solving non-linear arithmetic. In Bernhard Gramlich, Dale Miller, and Ulrike Sattler, editors, *Proceedings of the Sixth International Joint Conference on Automated Reasoning (IJCAR)*, volume 7364 of *Lecture Notes in Artificial Intelligence*, pages 339–354. Springer, 2012.
18. Sava Krstić and Amit Goel. Architecting solvers for SAT modulo theories: Nelson-Oppen with DPLL. In Frank Wolter, editor, *Proceedings of the Sixth International Symposium on Frontiers of Combining Systems (FroCoS)*, volume 4720 of *Lecture Notes in Artificial Intelligence*, pages 1–27. Springer, 2007.
19. Sharad Malik and Lintao Zhang. Boolean satisfiability: from theoretical hardness to practical success. *Communications of the ACM*, 52(8):76–82, 2009.
20. João P. Marques Silva, Inês Lynce, and Sharad Malik. Conflict-driven clause learning SAT solvers. In Armin Biere, Marjin Heule, Hans Van Maaren, and Toby Walsh, editors, *Handbook of Satisfiability*, volume 185 of *Frontiers in Artificial Intelligence and Applications*, pages 131–153. IOS Press, Amsterdam, 2009.
21. Greg Nelson and Derek C. Oppen. Simplification by cooperating decision procedures. *ACM Transactions on Programming Languages and Systems*, 1(2):245–257, 1979.
22. Robert Nieuwenhuis, Alberto Oliveras, and Cesare Tinelli. Solving SAT and SAT modulo theories: from an abstract Davis-Putnam-Logemann-Loveland procedure to DPLL(T). *Journal of the ACM*, 53(6):937–977, 2006.
23. Aleksandar Zeljić, Christoph M. Wintersteiger, and Philipp Rümmer. Deciding bit-vector formulas with `mcSAT`. In Nadia Creignou and Daniel Le Berre, editors, *Proceedings of the Nineteenth International Conference on Theory and Applications of Satisfiability Testing (SAT)*, volume 9710 of *Lecture Notes in Computer Science*, pages 249–266. Springer, 2016.