Gutiérrez Gil, R.; Meseguer, J.; Rocha, C. (2015). Order-Sorted Equality Enrichments Modulo Axioms. Science of Computer Programming. 99:235-261. doi:10.1016/j.scico.2014.07.003.

Order-Sorted Equality Enrichments Modulo Axioms[☆]

Raúl Gutiérrez[a,1], José Meseguer[a], Camilo Rocha[b]

[a] *Department of Computer Science, University of Illinois at Urbana-Champaign, 201 N. Goodwin Ave., Urbana, IL 61801-2302, USA*

[b] *Decanatura de Ingeniería de Sistemas, Escuela Colombiana de Ingeniería, AK 45 205-49, Bogotá, D.C., Colombia*

Abstract

Built-in equality and inequality predicates based on comparison of canonical forms in algebraic specifications are frequently used because they are handy and efficient. However, their use places algebraic specifications with initial algebra semantics beyond the pale of theorem proving tools based, for example, on explicit or induction-less induction techniques, and of other formal tools for checking key properties such as confluence, termination, and sufficient completeness. Such specifications would instead be amenable to formal analysis if an equationally-defined equality predicate enriching the algebraic data types were to be added to them. Furthermore, having an equationally-defined equality predicate is very useful in its own right, particularly in inductive theorem proving. Is it possible to *effectively* define a theory transformation $\mathcal{E} \mapsto \mathcal{E}^{\sim}$ that extends an algebraic specification $\mathcal{E}$ to a specification $\mathcal{E}^{\sim}$ having an equationally-defined equality predicate? This paper answers this question in the affirmative for a broad class of order-sorted conditional specifications $\mathcal{E}$ that are sort-decreasing, ground confluent, and operationally terminating modulo axioms B and have a subsignature of constructors. The axioms B can consist of associativity, or commutativity, or associativity-commutativity axioms, so that the constructors are *free modulo* B. We prove that the transformation $\mathcal{E} \mapsto \mathcal{E}^{\sim}$ preserves all the just-mentioned properties of $\mathcal{E}$. The transformation has been automated in Maude using reflection and is used as a component in many Maude formal tools.

Keywords:
equality predicate, order-sorted equational logic, modulo axioms, algebraic specifications, initial algebra semantics, inductive theorem proving, Maude

[☆]This work has been supported in part by NSF Grants CCF 09-05584 and CNS 13-19109, the EU (FEDER), the Spanish MINECO under Grants TIN2010-21062-C02 and TIN 2013-45732-C4-1-P, and by the Generalitat Valenciana, ref. PROMETEO/2011/052. Raúl Gutiérrez is also partially supported by a Juan de la Cierva Fellowship from the Spanish MINECO, ref. JCI-2012-13528.

Email addresses: rgutierrez@dsic.upv.es (Raúl Gutiérrez), meseguer@cs.illinois.edu (José Meseguer), camilo.rocha@escuelaing.edu.co (Camilo Rocha)

[1]Current address: Departament de Sistemes Informàtics i Computació, Universitat Politècnica de València, Camino de Vera s/n, 46022, Valencia, Spain

1. Introduction

It can be extremely useful, when reasoning about equational specifications with initial semantics, to have an explicit equational specification of the *equality predicate* as a binary Boolean-valued operator '$\sim$'. For example, in *theorem proving*, where the logic of universal quantifier-free formulas is automatically reduced to unconditional equational logic, the formula

$$(u \neq v \vee w = r) \wedge q = t$$

becomes equivalent to the single equation

$$(not(u \sim v)\ or\ w \sim r)\ and\ q \sim t = true.$$

Also in *inductionless induction* [1] where inductive proofs are reduced to proofs by consistency because any equation not holding inductively makes *true* and *false* equal.

An equationally-defined predicate can also be useful in the *elimination of built-in equalities and inequalities* that often are introduced in algebraic specifications through built-in operators. Such built-in equalities and inequalities are not defined logically but operationally, for both expressiveness and efficiency reasons, by comparison of canonical forms. However, their non-logical character renders any formal reasoning about specifications using them impossible.

In particular, the *use of formal tools* such as those checking termination, local confluence, or sufficient completeness of an algebraic specification is impossible[2] with built-in equalities and inequalities, but becomes possible when they are replaced by an equationally axiomatized equality predicate '$\sim$'. That is, the equality between t and t' is now expressed as $t \sim t' = true$, and their inequality as $t \sim t' = false$. Furthermore, the equality predicate $t \sim t'$ will *still be evaluated correctly* when t and t' are *terms with variables*, whereas an *operationally defined* built-in equality predicate will often give the *wrong* answer when evaluating such terms, even when the equations are confluent and terminating. For instance, for natural number addition '+', defined by equations $x + 0 = x$ and $x + s(y) = s(x + y)$, the terms $x + y$ and $y + x$ are *already* in canonical form and an operationally defined built-in equality predicate '==' will evaluate $x + y == y + x$ to *false*. Instead, using an equationally defined equality predicate '$\sim$', the term $x + y \sim y + x$ will remain in canonical form, and the equality $x+y \sim y+x = true$ can then be inductively proved using the equations defining '+' and '$\sim$'.

In principle, the meta-theorem of Bergstra and Tucker [2] ensures that any computable data type can be axiomatized as an initial algebra defined by a finite number of Church-Rosser and terminating equations. This also means that such a computable data type *plus* its equality predicate is also finitely axiomatizable by a finite set of Church-Rosser and terminating equations. However, the Bergstra-Tucker result is *non-*

[2]Since built-in equality and inequality predicates do not have a logical definition, they are outside the scope of formal tools whose inputs are equational theories. Here "built in" does *not* mean that there is a theory (say, that of the natural numbers) that has been built in: built-in equality and inequality predicates have no theory at all. The whole point of this paper is to provide a *logical* definition of equality and inequality predicates for an initial algebra in equational logic under the most general conditions possible.

constructive, in the sense that it does not give an algorithm to actually obtain the equational specification of the data type with its equality predicate. Therefore, what would be highly desirable in practice is a general *constructive theory transformation* $\mathcal{E} \mapsto \mathcal{E}^{\sim}$ that adds equationally-axiomatized equality predicates to an algebraic data type specification $\mathcal{E}$.

Such a transformation should be *as general as possible* if it is to be useful in practice. For example, a transformation applicable only to "vanilla-flavored" specifications without support for types and subtypes, or that excludes conditional equations and rewriting modulo axioms would be extremely limited. The transformation should also come with *strong preservation properties*. For example, if $\mathcal{E}$ is ground confluent, ground operationally terminating, and sufficiently complete, then $\mathcal{E}^{\sim}$ should also enjoy these same properties that are often essential both for executability and for applying a variety of formal reasoning methods.

These generality and property-preservation requirements on the transformation $\mathcal{E} \mapsto \mathcal{E}^{\sim}$ are a tall order. For instance, if f is a free constructor symbol, then the equations

$$f(x_1, \ldots, x_n) \sim f(y_1, \ldots, y_n) = x_1 \sim y_1 \textit{ and } \ldots \textit{ and } x_n \sim y_n$$
$$f(x_1, \ldots, x_n) \sim g(y_1, \ldots, y_m) = \textit{false}$$

give a perfectly good and straightforward axiomatization of equality for f, for each constructor $g \neq f$ of same type. But how can the equality predicate be defined when f satisfies, e.g., associativity and commutativity axioms? Also, how should sorts and subsorts, and subsort overloaded function symbols be dealt with? An even harder issue is the preservation of properties such as ground confluence, operational termination, and sufficient completeness. The difficulty is that for any given specification there are tools that can be used to prove such properties, but we need a proof that will work for *all* specifications in a very wide class. What we actually need are *metatheorems* ensuring that these properties are preserved under the transformation for *any* equational specification in the input class.

This paper presents an effective theory transformation $\mathcal{E} \mapsto \mathcal{E}^{\sim}$ that satisfies the above-mentioned preservation properties. The class of equational theories $\mathcal{E}$ accepted as inputs to the transformation is quite general. Modulo mild syntactic requirements, it consists of all order-sorted theories $\mathcal{E}$ of the form $(\Sigma, E \cup B)$ having a subsignature Ω of constructors and such that:

(i) B is a (possibly empty) set of axioms which may declare some operators associative, and/or commutative, and/or associative-commutative.

(ii) the equations E can be conditional and are sort-decreasing, ground confluent, and ground operationally terminating, and

(iii) the constructors Ω are *free modulo* B, i.e., there is an isomorphism of initial algebras $\mathcal{T}_{\Sigma/E \cup B|\Omega} \simeq \mathcal{T}_{\Omega/B}$.

Identity axioms are excluded from the transformation. However, by using the transformation described in [3] and subsort-overloaded operators, the transformation presented in this paper can often be extended to specifications that also include identity axioms.

The transformation $\mathcal{E} \mapsto \mathcal{E}^{\sim}$ is *constructive* and has been automated in Maude using its reflective features: it takes the meta-representation of $\mathcal{E}$ in Maude as input

and constructs a meta-representation of $\mathcal{E}^{\sim}$ as output. This automated transformation is used as a component in many Maude formal tools. In general, the contributions presented in this work open up many useful applications to improve the state of the art in formal verification of algebraic specifications. In particular, these ideas and their automation have already substantially improved the Maude formal environment.

This paper is also an extended version of the conference paper [4] with many substantial improvements. First of all a newer, simpler, and more generally applicable transformation $\mathcal{E} \mapsto \mathcal{E}^{\sim}$ than the one given in [4] is given here. The greater simplicity has to do with the fact that fewer conditional equations are now used in the definition of the equality predicate '$\sim$'. The greater generality resides in the support for several maximal subsort-overloaded versions of a constructor, whereas in [4] a maximal subsort-overloaded constructor was implicitly assumed. Full proofs of all results are given for this newer transformation, which is the one currently implemented in Maude. To make the main ideas of the paper more easily accessible while keeping the paper self-contained, some technical details and some lengthy proofs have been relegated to an electronic appendix fully available with this paper.

1.1. Related Work

In [1], Goguen generalizes and simplifies the technique given by Musser in [5] for proving induction hypothesis without induction (so-called *inductionless induction*) using enriched theories with equality. The notion of *s-taut* related to a sort *s* can be seen as a initial approximation of what we called in this paper an equality enrichment, i.e., conditions of *tautness* are present in equality enrichments. The result obtained in Goguen's paper [1] is stated in Corollary 2.

In [6], Meseguer and Goguen define the notion of *equality enrichment* (without axioms) as an explicit representation of an *equational equality presentation*. Our work extends this notion of equality enrichment with subsorts and axioms and also presents an automatic way to generate this equality enrichment modulo axioms.

In [7], Nakamura and Futatsugi propose an equality predicate for algebraic specifications. Unlike our work, their work does not consider axioms and sufficient completeness in their theories, hence they have to manage terms with defined symbols. In the positive cases, their equality predicate is equivalent to ours, but in the negative cases, a *false* answer in [7] does not mean that both terms are distinct for any possible instantiation (as we state in our work), because the negative rules are based on a check of convergence between terms. The goal of this behavior is to avoid false positives instead of capturing negative cases.

Paper outline. This paper is organized as follows. Preliminaries are gathered in Section 2. General properties satisfied by an equality enrichment are defined and described in Section 3. The constructive transformation $\mathcal{E} \mapsto \mathcal{E}^{\sim}$ is given in detail in Section 4. The meta-theoretical proofs about the transformation $\mathcal{E} \mapsto \mathcal{E}^{\sim}$ preserving all the desirable executability properties, including operational termination, confluence, and sufficient completeness, can be found in Section 5. This section also presents the proof that the equality enrichment transformation indeed outputs an equality enrichment of the input theory. Section 6 describes how the transformation $\mathcal{E} \mapsto \mathcal{E}^{\sim}$ has been automated

in Maude and presents a case study on how it can be used by other formal verification tools. Some concluding remarks can be found in Section 7.

2. Preliminaries

This paper uses standard notation and terminology about terms, term algebras, and order-sorted equational theories as employed, for example, by [8] and [9].

2.1. Order-Sorted Equational Theories

An *order-sorted signature* Σ is a tuple $\Sigma = (S, \leq, F)$ with a finite poset of sorts $(S, \leq)$ and set of function symbols F. The equivalence relation $\equiv_\leq$ is defined for any $s, s' \in S$ by $s \equiv_\leq s'$ iff $s(\leq \cup \geq)^+ s'$ (where R^+ denotes the transitive closure of R). The function symbols in F satisfy the condition that, for $(w, s), (w', s') \in S^* \times S$ and $w, w' \in S^*$ having the same length, if $f \in F_{w,s} \cap F_{w',s'}$, then $w \equiv_\leq w'$ implies $s \equiv_\leq s'$. A *top sort* in Σ is a sort $s \in S$ such that if $s' \in S$ and $s \equiv_\leq s'$, then $s' \leq s$. We denote by $\lfloor s \rfloor = \{s' \in S \mid s' \leq s\}$ the *ideal* of a sort $s \in S$. For any sort $s \in S$, the expression $[s]$ denotes the connected component of s, that is, $[s] = [s]_{\equiv_\leq}$, called the *kind* of the connected component. We say that $f : w \to s$ and $f : w' \to s'$ are *subsort-overloaded* iff $w \equiv_\leq w'$ and $s \equiv_\leq s'$. A function symbol $f : s_1 \cdots s_n \to s \in F$ is a *maximal typing* of f in Σ if there is no other subsort-overloaded $f : s'_1 \cdots s'_n \to s' \in F$ such that $s_i \leq s'_i$, $1 \leq i \leq n$, and $s \leq s'$. We say that $f : s_1 \cdots s_n \to s' \in F$ is a *maximal typing of f with respect to s* if it is a maximal typing of f in the subsignature of Σ obtained by keeping only the function symbols whose return type s'' satisfies $s'' \leq s$. A pair of sorts $s_1, s_2 \in S$ are called *disjoint* iff there is no $s_3 \in S$ such that $s_3 \leq s_1$ and $s_3 \leq s_2$. We say that s_1, s_2 are *maximally disjoint* iff s_1, s_2 are disjoint and there is no $s_3 \in S$ such that either $s_3 > s_1$ and s_3, s_2 are disjoint, or $s_3 > s_2$ and s_3, s_1 are disjoint. We always assume that if a constructor symbol $f \in F$ is commutative then its two arguments have the same sort, and if it is associative or associative-commutative then its two arguments and its return sort are all the same.

We assume a collection of *variables* X, where X is an S-indexed family $X = \{X_s\}_{s \in S}$ of mutually disjoint variable sets with each X_s countably infinite. The *set of terms of sort s* is denoted $T_\Sigma(X)_s$ and the *set of ground terms of sort s* is denoted $T_{\Sigma,s}$. The expressions $\mathcal{T}_\Sigma(X)$ and $\mathcal{T}_\Sigma$ denote the corresponding order-sorted Σ-term algebras. We assume that all order-sorted signatures are *preregular* [9], i.e., that each Σ-term has a *least sort* $ls(t) \in S$ such that $t \in T_\Sigma(X)_{ls(t)}$. The *set of variables* of a term t is written $var(t)$ and is extended to sets of terms in the natural way.

A *substitution* is an S-indexed mapping θ that maps variables $x \in X_s$ to terms $\theta(x) \in T_\Sigma(X)_s$, for $s \in S$, and is different from the identity for a finite subset of X. The identity substitution is denoted by id and the expression $\theta|_Y$ denotes the restriction of a substitution θ to a set of variables $Y \subseteq X$. The expression $dom(\theta)$ denotes the *domain* of θ, i.e., $dom(\theta) = \{x \in X \mid \theta(x) \neq x\}$, and the expression $ran(\theta)$ denotes the set of variables introduced by θ, i.e., $ran(\theta) = \bigcup\{var(\theta(x)) \mid x \in dom(\theta)\}$. Substitutions extend homomorphically to terms in the natural way. A substitution θ is called *ground* iff $ran(\theta) = \varnothing$. The application of a substitution θ to a term t is denoted by $t\theta$ and the composition of two substitutions θ_1 and θ_2 is denoted by $\theta_1\theta_2$.

A Σ-*equation* is an unordered pair $t = u$ with $t \in T_\Sigma(X)_{s_t}$, $u \in T_\Sigma(X)_{s_u}$, and $s_t \equiv_\leq s_u$. A *conditional* Σ-*equation* is a Horn clause $t = u$ **if** C with $t = u$ a Σ-equation and C a finite conjunction $\bigwedge_i u_i = v_i$ of Σ-equations. An *equational theory (presentation)* is a pair $\mathcal{E} = (\Sigma, E)$ with Σ an order-sorted signature and E a finite set of conditional Σ-equations. Throughout this paper we assume that $T_{\Sigma,s} \neq \varnothing$ for each $s \in S$, because this affords a simpler deduction system. For φ a conditional Σ-equation, $(\Sigma, E) \vdash \varphi$ iff φ can be proved from (Σ, E) by the deduction rules in [10] iff φ is valid in all models of (Σ, E) [10].

An equational theory $\mathcal{E} = (\Sigma, E)$ induces a congruence relation $=_E$ on $T_\Sigma(X)$ defined for any $t, u \in T_\Sigma(X)$ by $t =_E u$ iff $(\Sigma, E) \vdash t = u$. $\mathcal{T}_{\Sigma/E}(X)$ and $\mathcal{T}_{\Sigma/E}$ denote the quotient algebras induced by $=_E$ on the algebras $\mathcal{T}_\Sigma(X)$ and $\mathcal{T}_\Sigma$, respectively. The quotient algebra $\mathcal{T}_{\Sigma/E}$, also denoted $\mathcal{T}_\mathcal{E}$, is called the *initial algebra* of (Σ, E). A conditional Σ-equation φ is an *inductive consequence* of (Σ, E) iff $\mathcal{T}_{\Sigma/E} \models \varphi$, i.e., iff $(\forall \theta : X \longrightarrow T_\Sigma)(\Sigma, E) \vdash \varphi\theta$. A theory inclusion $(\Sigma, E) \subseteq (\Sigma', E')$, with $\Sigma \subseteq \Sigma'$ and $E \subseteq E'$, is called *protecting* iff the unique Σ-homomorphism $\mathcal{T}_{\Sigma/E} \longrightarrow \mathcal{T}_{\Sigma'/E'}|_\Sigma$ of the Σ-reduct of the initial algebra $\mathcal{T}_{\Sigma'/E'}$ is a Σ-isomorphism, written $\mathcal{T}_{\Sigma/E} \simeq \mathcal{T}_{\Sigma'/E'}|_\Sigma$.

Reasonable admissibility requirements are needed to make a theory *executable* (see [11], Sections 4.6 and 6.3). We assume that the set of Σ-equations of an equational theory $\mathcal{E}$ can be decomposed into a disjoint union $E \cup B$, with B a collection of structural axioms for which there exists a *matching algorithm modulo* B producing a finite number of B-matching solutions, or failing otherwise. Throughout this paper the structural axioms B will always be a combination of commutativity, associativity, and associativity-commutativity axioms for some (or none) of the function symbols in Σ. Furthermore, the signature Σ should be not only preregular, so that each term t has a least sort $ls(t)$, but also B-*preregular*, in the sense that if $t =_B t'$, then $ls(t) = ls(t')$. For B any combination of commutativity, associativity, and associativity-commutativity axioms, B preregularity is easily checkable; the check is indeed automated in the Maude language [11, 22.2.5]. We also assume that the equations E can be oriented into a set of sort-decreasing, operationally terminating, and confluent conditional rewrite rules $E^\bullet$ modulo B, where the rewrite relation modulo B, $\rightarrow_{E^\bullet/B}$ is defined[3] as the relation composition $=_B; \rightarrow_{E^\bullet}; =_B$, with $\rightarrow_{E^\bullet}$ the standard rewrite relation for rules $E^\bullet$. Finally we assume that the rules $E^\bullet$ are *coherent* modulo B [12], which for B any combination of associativity and/or commutativity axioms can always be ensured by adding a few "extension rules" if necessary (see [11, 4.8] for an informal explanation with examples of extension rules). This ensures that the rewrite relation $\rightarrow_{E^\bullet/B}$ can be (bi)simulated by the simpler rewrite relation $\rightarrow_{E^\bullet,B}$ that uses a B-matching algorithm, where, by definition, $t \rightarrow_{E^\bullet,B} t'$ iff t has a subterm decomposition $t[u]$ such that there is a rule $(l \rightarrow r) \in E^\bullet$ and a substitution θ such that $u =_B l\theta$, and $t' = t[r\theta]$.

By definition, $E^\bullet$ is *sort decreasing* modulo B iff for each $t \rightarrow u$ **if** $C \in E^\bullet$ and substitution θ, $ls(t\theta) \geq ls(u\theta)$ if $(\Sigma, E \cup B) \vdash C\theta$. The set $E^\bullet$ is *operationally terminating* modulo B iff there is no infinite well-formed proof tree modulo B in $E^\bullet$ [13]. The set $E^\bullet$ is *confluent* modulo B iff for all $t, t_1, t_2 \in T_\Sigma(X)$, if $t \rightarrow^*_{E^\bullet/B} t_1$ and $t \rightarrow^*_{E^\bullet/B} t_2$,

[3]For simplicity we define here $\rightarrow_{E^\bullet/B}$ for the unconditional case. The precise definition of $\rightarrow_{E^\bullet/B}$ when the rules are conditional is given in Figure 5 in Section 5.2.

then there exist $u_1, u_2 \in T_\Sigma(X)$ such that $t_1 \rightarrow^*_{E^\bullet/B} u_1$, $t_2 \rightarrow^*_{E^\bullet/B} u_2$, and $u_1 =_B u_2$. The set of rewrite rules $E^\bullet$ modulo B is *ground sort-decreasing*, *ground operationally terminating*, and *ground confluent* iff it is, respectively, sort-decreasing, operationally terminating, and confluent for ground terms. The term $t\downarrow_{E/B} \in T_\Sigma(X)$ denotes the *E-canonical form* of t modulo B (or *E/B-canonical form*) so that $t \rightarrow^*_{E^\bullet/B} t\downarrow_{E/B}$ and $t\downarrow_{E/B}$ is $\rightarrow_{E^\bullet/B}$*-irreducible*, i.e., it cannot be further reduced by $\rightarrow_{E^\bullet/B}$. Under the above assumptions $t\downarrow_{E/B}$ is unique up to B-equality. The expression $Can_{\Sigma,E/B}$ denotes the S-indexed set of E/B-canonical forms of T_Σ understood as B-equivalence classes.

Given $\mathcal{E} = (\Sigma, E \cup B)$ ground sort-decreasing, ground operationally terminating, and ground confluent modulo B, an order-sorted signature $\Omega \subseteq \Sigma$ is called a subsignature of *constructors* iff Ω has the same poset of sorts as Σ, and for each sort s in Σ and ground term $t \in T_{\Sigma,s}$ there is a $u \in T_{\Omega,s}$ satisfying $t \rightarrow^*_{E/B} u$. Furthermore, $\Omega \subseteq \Sigma$ is called a subsignature of *free constructors* modulo[4] B iff Ω is a signature of constructors modulo B and $Can_{\Sigma,E/B} = T_{\Omega/B_\Omega}$, with B_Ω as defined in Footnote 4.

3. Equality Enrichments

An *equality enrichment* of an equational theory is another equational theory that extends the former with the definition of a Boolean-valued equality function symbol that characterizes equality of ground terms in the original theory. One advantage of the newly defined equality predicate is that –in contrast to an *operationally defined* built-in equality predicate '=='– it is sound for symbolic deductive reasoning about equational inductive properties.

This section assumes an order-sorted signature $\Sigma = (S, \leq, F)$ and an order-sorted equational theory $\mathcal{E} = (\Sigma, E)$ with initial algebra $\mathcal{T}_\mathcal{E}$. Definition 1 formally introduces the notion of an equality enrichment $\mathcal{E}^\sim$ of $\mathcal{E}$, with equality predicate '$\sim$' that coincides with '=' on $\mathcal{T}_\mathcal{E}$.

Definition 1 (Equality Enrichment, generalizes [6, Definition 68])**.** *An equational theory $\mathcal{E}^\sim = (\Sigma^\sim, E^\sim)$ is called an* equality enrichment *of $\mathcal{E}$, with $\Sigma^\sim = (S^\sim, \leq^\sim, F^\sim)$, iff the following conditions are satisfied:*

- *$\mathcal{E}^\sim$ is a protecting extension of $\mathcal{E}$;*
- *the poset of sorts of $\Sigma^\sim$ extends $(S, \leq)$ by adding a new sort Bool that belongs to a new connected component, with constants $\top$ and $\bot$ such that $\mathcal{T}_{\mathcal{E}^\sim,Bool} = \{[\top], [\bot]\}$ and $\top \neq_{E^\sim} \bot$; and*
- *for each connected component $[s]$ in $(S, \leq)$ if there is not a top sort $k \in [s]$, then a new top sort $k \in S^\sim$ is added; in either case, a binary commutative operator*

$$\_\sim\_ : k\ k \longrightarrow Bool \in F^\sim$$

[4] In general the axioms B may also involve defined function symbols not present in Ω. In such case, the terminology "free constructors modulo B" involves some abuse of notation. Properly speaking, one should talk of *free constructors modulo B_Ω* where, by definition, $B_\Omega = \{u = v \in B \mid u, v \in T_\Omega(X)\}$. In what follows we will ignore these details and will always use the simpler description "free constructors modulo B".

is added to $\Sigma^\sim$ *such that the following equivalences hold for any* ground terms $t, u \in T_{\Sigma,k}$*:*

$$\mathcal{E} \vdash t = u \qquad \Longleftrightarrow \qquad \mathcal{E}^\sim \vdash (t \sim u) = \top, \tag{1}$$

$$\mathcal{E} \nvdash t = u \qquad \Longleftrightarrow \qquad \mathcal{E}^\sim \vdash (t \sim u) = \bot. \tag{2}$$

An equality enrichment $\mathcal{E}^\sim$ *of* $\mathcal{E}$ *is called* Boolean *iff it contains all the function symbols and equations making the elements of* $\mathcal{T}_{\mathcal{E}^\sim,Bool}$ *a two-element Boolean algebra.*

The equality predicate '$\sim$' in $\mathcal{E}^\sim$ is sound for inferring equalities and inequalities in the initial algebra $\mathcal{T}_\mathcal{E}$, even for terms with variables. The precise meaning of this claim is given by Proposition 1.

Proposition 1 (Equality Enrichment Properties)**.** *Let* $\mathcal{E}^\sim = (\Sigma^\sim, E^\sim)$ *be an equality enrichment of* $\mathcal{E}$*. If* $t, u \in T_\Sigma(X)$*, then the following equivalences hold:*

$$\mathcal{T}_\mathcal{E} \models (\forall X)\ t = u \qquad \Longleftrightarrow \qquad \mathcal{T}_{\mathcal{E}^\sim} \models (\forall X)\ (t \sim u) = \top, \tag{3}$$

$$\mathcal{T}_\mathcal{E} \models (\exists X)\ t = u \qquad \Longleftrightarrow \qquad \mathcal{T}_{\mathcal{E}^\sim} \models (\exists X)\ (t \sim u) = \top, \tag{4}$$

$$\mathcal{T}_\mathcal{E} \models (\forall X)\ \neg(t = u) \qquad \Longleftrightarrow \qquad \mathcal{T}_{\mathcal{E}^\sim} \models (\forall X)\ (t \sim u) = \bot, \tag{5}$$

$$\mathcal{T}_\mathcal{E} \models (\exists X)\ \neg(t = u) \qquad \Longleftrightarrow \qquad \mathcal{T}_{\mathcal{E}^\sim} \models (\exists X)\ (t \sim u) = \bot. \tag{6}$$

Proof. The following is a proof of Statement (3); a proof of statements (4), (5), and (6) can be obtained in a similar way.

$\mathcal{T}_\mathcal{E} \models (\forall X)\ t = u$
$\Longleftrightarrow$ { by definition of satisfaction in $\mathcal{T}_\mathcal{E}$ }
$(\forall \sigma : X \longrightarrow T_\Sigma)\ \mathcal{E} \vdash t\sigma = u\sigma$
$\Longleftrightarrow$ { by (1) }
$(\forall \sigma : X \longrightarrow T_\Sigma)\ \mathcal{E}^\sim \vdash (t\sigma \sim u\sigma) = \top$
$\Longleftrightarrow$ { by $\mathcal{E}$ being protected by $\mathcal{E}^\sim$ and the sorts of t, u in Σ }
$(\forall \sigma : X \longrightarrow T_{\Sigma^\sim})\ \mathcal{E}^\sim \vdash (t\sigma \sim u\sigma) = \top$
$\Longleftrightarrow$ { by definition of satisfaction in $\mathcal{T}_{\mathcal{E}^\sim}$ }
$\mathcal{T}_{\mathcal{E}^\sim} \models (\forall X)\ (t \sim u) = \top.$

□

By using an equality enrichment $\mathcal{E}^\sim$ of $\mathcal{E}$, the problem of reasoning in $\mathcal{T}_\mathcal{E}$ about a universally quantified inequality $\neg(t = u)$ (abbreviated $t \neq u$) can be reduced to reasoning in $\mathcal{T}_{\mathcal{E}^\sim}$ about the universally quantified equality $(t \sim u) = \bot$. A considerably more general reduction, not just for inequalities but for *arbitrary quantifier-free first-order formulae*, can be obtained with Boolean equality enrichments.

Corollary 1. *Let* $\mathcal{E}^\sim = (\Sigma^\sim, E^\sim)$ *be a Boolean equality enrichment of* $\mathcal{E}$*. Let* $\varphi = \varphi(t_1 = u_1, \ldots, t_n = u_n)$ *be a quantifier-free Boolean formula whose atoms are the* Σ*-equations* $t_i = u_i$ *with variables in* X*, for* $1 \leq i \leq n$*, and with Boolean connectives in* $\{\neg, \vee, \wedge\}$*.*

Then, the following equivalence holds:

$$\mathcal{T}_{\mathcal{E}} \models (\forall X)\varphi \quad \Longleftrightarrow \quad \mathcal{T}_{\mathcal{E}^{\sim}} \models (\forall X)\, \widehat{\varphi}(t_1 \sim u_1, \ldots, t_n \sim u_n) = \top, \tag{7}$$

where $\widehat{\varphi}(t_1 \sim u_1, \ldots, t_n \sim u_n)$ is the $\Sigma^{\sim}$-term of sort Bool obtained from φ by replacing each occurrence of the logical connectives $\neg$, $\vee$, and $\wedge$ by, respectively, the function symbols $\lrcorner$, $\sqcup$, and $\sqcap$ in $\mathcal{E}^{Bool}$, and each occurrence of an atom $t_i = u_i$ by the Bool term $t_i \sim u_i$, for $1 \leq i \leq n$.

Proof. By structural induction on φ using Proposition 1 for base cases. □

A key property of an equality enrichment $\mathcal{E}^{\sim}$ of $\mathcal{E}$ is that, if $\mathcal{E}^{\sim}$ is extended with any set E' of Σ-equations that are *not* satisfiable in $\mathcal{T}_{\mathcal{E}}$, then the resulting extension is inconsistent in the sense that the *contradiction* $\top = \bot$ can be derived. Conversely, if the set E' of Σ-equations extending $\mathcal{E}^{\sim}$ is satisfied by $\mathcal{T}_{\mathcal{E}}$, then the resulting extension is consistent and therefore cannot yield a proof of contradiction. Statements (9) and (10) in Corollary 2 account for these two facts.

Lemma 1. *If E' is a set of Σ-equations, then the following equivalence holds:*

$$\mathcal{T}_{\Sigma/E} \models E' \qquad \Longleftrightarrow \qquad \mathcal{T}_{\Sigma/E} \simeq \mathcal{T}_{\Sigma/E \cup E'}. \tag{8}$$

Proof. The ($\Longleftarrow$) direction is clear, since it is always true that $\mathcal{T}_{\Sigma/E \cup E'} \models E'$, and therefore $\mathcal{T}_{\Sigma/E} \models E'$ because satisfaction is preserved by isomorphisms. To see the ($\Longrightarrow$) direction, note that, since $\mathcal{T}_{\Sigma/E} \models E$, $\mathcal{T}_{\Sigma/E} \models E'$ implies $\mathcal{T}_{\Sigma/E} \models E \cup E'$. The initiality of $\mathcal{T}_{\Sigma/E \cup E'}$ then forces the existence of a unique Σ-homomorphism $h : \mathcal{T}_{\Sigma/E \cup E'} \longrightarrow \mathcal{T}_{\Sigma/E}$, and the initiality of $\mathcal{T}_{\Sigma/E}$ and the fact that $\mathcal{T}_{\Sigma/E \cup E'} \models E$ forces likewise a unique Σ-homomorphism $q : \mathcal{T}_{\Sigma/E} \longrightarrow \mathcal{T}_{\Sigma/E \cup E'}$. But then the initiality of $\mathcal{T}_{\Sigma/E}$ forces $q; h = 1_{\mathcal{T}_{\Sigma/E}}$, and the initiality of $\mathcal{T}_{\Sigma/E \cup E'}$ forces $h; q = 1_{\mathcal{T}_{\Sigma/E \cup E'}}$. Therefore, $\mathcal{T}_{\Sigma/E} \simeq \mathcal{T}_{\Sigma/E \cup E'}$, as desired. □

Corollary 2 (Generalizes [6, Theorem 74])**.** *Let $\mathcal{E}^{\sim} = (\Sigma^{\sim}, E^{\sim})$ be an equality enrichment of $\mathcal{E}$ and let E' be a collection of Σ-equations. Then the following equivalences hold:*

$$\mathcal{T}_{\mathcal{E}} \not\models E' \qquad \Longleftrightarrow \qquad (\Sigma^{\sim}, E^{\sim} \cup E') \vdash \top = \bot, \tag{9}$$

$$\mathcal{T}_{\mathcal{E}} \models E' \qquad \Longleftrightarrow \qquad (\Sigma^{\sim}, E^{\sim} \cup E') \nvdash \top = \bot. \tag{10}$$

Proof. Note that statements (9) and (10) are logically equivalent. The following is a proof of (10). Without loss of generality assume that the Σ-equations in E' are unconditional. Otherwise, if the equations in E' are conditional, they can be replaced by the set $E'' = \{t = u \mid E \cup E' \vdash t = u\}$. It is then easy to prove that $\mathcal{T}_{\mathcal{E}} \models E'$ if and only if $\mathcal{T}_{\mathcal{E}} \models E''$. The forward direction is first proved:

$\mathcal{T}_{\Sigma/E} \models E'$
$\Longleftrightarrow$ { by $\mathcal{E}^{\sim}$ being a protecting theory extension of $\mathcal{E}$ }
$\mathcal{T}_{\Sigma^{\sim}/E^{\sim}} \models E'$
$\Longleftrightarrow$ { by Lemma 1 }
$\mathcal{T}_{\Sigma^{\sim}/E^{\sim} \cup E'} \simeq \mathcal{T}_{\Sigma^{\sim}/E^{\sim}}$.

Therefore $\mathcal{T}_{\Sigma^{\sim}/E^{\sim}\cup E',Bool} = \{[\top],[\bot]\}$, and hence $(\Sigma^{\sim}, E^{\sim} \cup E') \not\vdash \top = \bot$.

The proof in the other direction is obtained by contrapositive. If $\mathcal{T}_{\mathcal{E}} \not\models E'$, then there are ground terms $t, u \in T_{\Sigma}$ such that (i) $(\Sigma, E) \not\vdash t = u$ and (ii) $(\Sigma, E \cup E') \vdash t = u$. By Definition 1, (i) is equivalent to $(\Sigma^{\sim}, E^{\sim}) \vdash (t \sim u) = \bot$ and thus, by monotonicity of equational deduction, $(\Sigma^{\sim}, E^{\sim} \cup E') \vdash (t \sim u) = \bot$. By (ii) and the monotonicity of equational deduction, it follows that $(\Sigma^{\sim}, E^{\sim} \cup E') \vdash t = u$. Then, from Definition 1, it follows that $(\Sigma^{\sim}, E^{\sim} \cup E') \vdash (t \sim u) = \top$. Therefore $(\Sigma^{\sim}, E^{\sim} \cup E') \vdash \top = \bot$. □

4. An Effective Transformation with Free Constructors Modulo

This section presents an effective theory transformation $\mathcal{E} \mapsto \mathcal{E}^{\sim}$ for enriching with an equality predicate order-sorted equational theories $\mathcal{E}$ that are executable in the sense of satisfying the admissibility conditions in Section 2, and such that they have a subsignature of free constructors modulo structural axioms. In the development to follow, $\mathcal{E} = (\Sigma, E \cup B)$ is an admissible order-sorted equational theory with signature $\Sigma = (S, \leq, F)$ and subsignature Ω of free constructors modulo B. The axioms B are any union of associative (A), commutative (C), and associative-commutative (AC) axioms. Furthermore, the following convention is adopted: for x a variable and s a sort, the expression x_s indicates that x has sort s, i.e., $x \in X_s$.

The theory transformation $\mathcal{E} \mapsto \mathcal{E}^{\sim}$ consists of two main tasks or subtransformations. On input $\mathcal{E}$, it first extends $\mathcal{E}$ by adding new sorts, the equational theory $\mathcal{E}^{Bool}$ of Booleans with constructors $\top$ and $\bot$ (and with the other usual Boolean connectives equationally defined), some auxiliary functions, the equality predicate '$\sim$' for each top sort in the input theory $\mathcal{E}$, and some equations defining '$\sim$' that solely depend on sort information in Σ. Second, it generates a set of equations defining '$\sim$' that *do* depend on the constructor symbols in Ω and their structural axioms.

Transformation 1. Extends the input theory $\mathcal{E}$ by adding:

(i) a fresh top sort for each connected component in Σ that does not have it,
(ii) the theory $\mathcal{E}^{Bool}$ with sort *Bool*,
(iii) for each top sort k in each connected component of $\mathcal{E}$, the Boolean-valued (binary) commutative operator '$\sim$',
(iv) some equations partially defining '$\sim$' for top sorts, and
(v) for each $f \in F_{\Omega}$ with AC structural axioms the Boolean-valued binary operator 'in_f^k'.

Transformation 2. For each $f \in F_{\Omega}$, and depending on the structural axioms of f, generates a suitable set of equations defining '$\sim$' and 'in_f^k', where the auxiliary Boolean-valued operator 'in_f^k' is used for checking if a term rooted by an AC constructor symbol f contains or not a term not rooted by f but by another constructor g under the constructor symbol f.

We assume a theory $\mathcal{E}^{Bool}$ specifying the Booleans as its initial algebra that is confluent, and terminating modulo axioms B^{Bool} (which may include any combination of associativity and/or commutativity axioms), and that has a signature of free

constructors $\Omega^{Bool} = \{\top, \bot\}$ modulo B^{Bool}, set of defined symbols $\Sigma^{Bool} \setminus \Omega^{Bool} = \{ \lrcorner, \sqcap, \sqcup, \oplus, \supset \}$, (alternatively, $\Sigma^{Bool} \setminus \Omega^{Bool} = \{ \lrcorner, \sqcap, \sqcup, \equiv, \supset \}$), and satisfies $\mathcal{T}_{\mathcal{E}^{Bool}} \models \top \neq \bot$. The choice of $\mathcal{E}^{Bool}$ is quite general: we can for example choose any equational theory implementing a rewriting-based Boolean decision procedure, such as Hsiang's Boolean ring theory $\mathcal{E}^{Bool}$ (see, e.g., [11, Section 9.1]), or the Dijkstra-Shoelten specification $\mathcal{E}^{Bool}$ in [14] (for two other alternative rewriting-based Boolean decision procedures see [14]). Even a more basic theory, such as the theory $\mathcal{E}^{BBool}$ presented below, which specifies the initial algebra of the Booleans but falls short of being a decision procedure, will suffice for our purposes and will make some confluence proofs easier. In Section 5.3 we show how this theory can be easily extended to obtain an actual decision procedure.

Example 1. *In the following theory $\mathcal{E}^{BBool}$, equations (11)–(16), denoted E_{BBool}, can be oriented as rewrite rules from left to right* modulo *the associativity and commutativity axioms for $\sqcup$ and $\sqcap$, denoted B_{BBool}.*

$$x \sqcup \top = \top \tag{11}$$
$$x \sqcup \bot = x \tag{12}$$
$$x \sqcap \bot = \bot \tag{13}$$
$$x \sqcap \top = x \tag{14}$$
$$\lrcorner \top = \bot \tag{15}$$
$$\lrcorner \bot = \top \tag{16}$$

Note that, since they decrease term size, while associativity and commutativity axioms preserve it, equations (11)–(16) are terminating. Furthermore, equations (11)–(16) are locally confluent modulo the associativity and commutativity axioms and therefore confluent. Also, rules (11)–(16) are closed under AC extensions and are therefore coherent modulo B_{BBool}; therefore, the rewrite relation $\rightarrow_{E_{BBool}/B_{BBool}}$ can be (bi)simulated by the rewrite relation $\rightarrow_{E_{BBool},B_{BBool}}$. Sufficient completeness of $\{\top, \bot\}$ as a subsignature of free constructors is easily checkable by showing that all unary and binary operators, when suppied with any combination of $\top$ or $\bot$ arguments, reduce to either $\top$ or $\bot$, plus the fact that $\top$ and $\bot$ are irreducible. Note that the other Boolean operators can be specified in terms of $\sqcup$, $\sqcap$ and $\lrcorner$ by definitional extensions that do not alter the confluence, termination and sufficient completeness of $\mathcal{E}^{BBool}$. For example, $\supset$ can be defined by the equation, $x \supset y = (\lrcorner x) \sqcup y$.

Defining the equality predicate '$\sim$', for the positive case is simple. Indeed, this case is defined in Transformation 1 and is captured by the equation:

$$x_k \sim x_k = \top,$$

for each top sort k in each connected component of $\mathcal{E}$. Subsort overloading of constructor symbols in conjunction with maximal typings and structural axioms introduce some subtleties for correctly defining the *negative* cases of '$\sim$'. Of course, since constructors are free modulo B by assumption, two terms $t = f(\cdots)$ and $u = g(\cdots)$ with the same sort but with *syntactically* different constructors f and g in their roots, always yield $\bot$ when compared with '$\sim$'.

To better explain how subsort overloading of constructor symbols and disjoint sorts are dealt with by the equations added in Transformation 1, we can consider scenarios (a), (b), and (c) depicted in Figure 1. In these three scenarios, there are two connected components in the sort order, namely, $[s_1]$ and $[s]$. In Scenario (a) there is exactly one unary constructor symbol f. In scenarios (b) and (c) there are two, resp. three, overloaded versions of the constructor f.

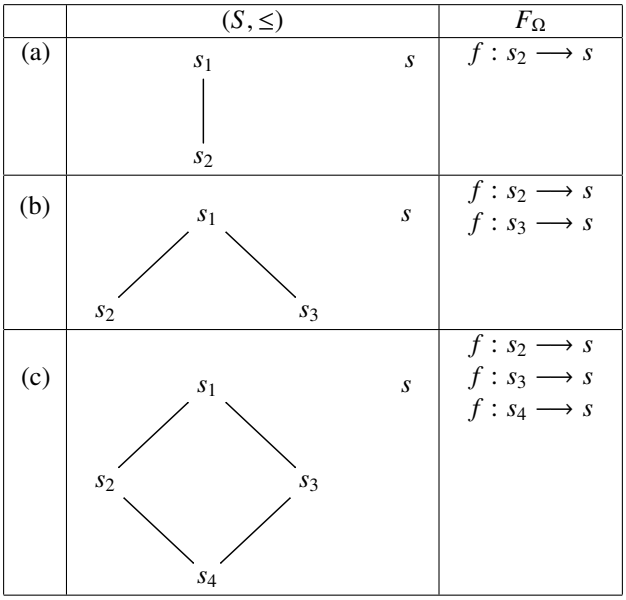

Figure 1: Some cases of constructor overloading.

In Scenario (a), the simplest of the three scenarios, there are three sorts s_1, s_2, s and two connected components. Function symbol $f : s_2 \longrightarrow s$ is the only constructor symbol. Note that, as defined in this scenario, it is perfectly fine if f is not a constructor at the level of s_1. In this case, there is exactly one maximal typing for the constructor f and the definition of '$\sim$' is straightforward by recursion on the argument of f:

$$f(x_{s_2}) \sim f(y_{s_2}) = x_{s_2} \sim y_{s_2}$$

In Scenario (b), there are four sorts s_1, s_2, s_3, s and two connected components. The situation is more interesting than in Scenario (a) because there are two maximal typings for the constructor f, namely, $f : s_2 \longrightarrow s$ and $f : s_3 \longrightarrow s$. First note that $T_\Omega(X)_{s_2} \cap T_\Omega(X)_{s_3} = \varnothing$ because the sorts s_2 and s_3 have no sort below them and, by preregularity, sort-decreasingness and confluence (admissibility requirements from Section 2), every term has a lowest sort; therefore, the comparison using '$\sim$' of *any* two terms, with corresponding sorts s_2 and s_3, should always yield $\bot$. Note also that if a term $g(\cdots) \in T_\Omega(X)_{s_1}$, then either $g(\cdots) \in T_\Omega(X)_{s_2}$ or $g(\cdots) \in T_\Omega(X)_{s_3}$ because of the preregularity assumption on Σ. Therefore, in addition to the equation capturing the above-mentioned positive case, the following equations define '$\sim$' for f in Scenario

(b):

$$
\begin{aligned}
x_{s_2} \sim y_{s_3} &= \bot \\
f(x_{s_2}) \sim f(y_{s_2}) &= x_{s_2} \sim y_{s_2} \\
f(x_{s_3}) \sim f(y_{s_3}) &= x_{s_3} \sim y_{s_3} \\
f(x_{s_2}) \sim f(y_{s_3}) &= x_{s_2} \sim y_{s_3}
\end{aligned}
$$

Note that in practice, the last equation can be replaced by $f(x_{s_2}) \sim f(y_{s_3}) = \bot$ if we are in Scenario (b). Scenario (c) is similar to Scenario (b), but there is a new sort s_4 below s_2 and s_3. In this case, the emptiness $T_\Omega(X)_{s_2} \cap T_\Omega(X)_{s_3} = \varnothing$ does not hold if all sorts are inhabited. Indeed, by our general assumption that $T_{\Sigma,s} \neq \varnothing$ for any sort s in Σ, we have $T_\Omega(X)_{s_2} \cap T_\Omega(X)_{s_3} \neq \varnothing$, so that the first equation given for Scenario (b) does not hold in Scenario (c). Note, finally, that, by f being an overloaded constructor with argument sort s_2 or s_3, f can also be applied to arguments of sort s_4, since $s_4 \leq s_2$ and $s_4 \leq s_3$. Indeed, the equations defining '$\sim$' in Scenario (c) are all but the first equation in Scenario (b).

Definition 2 spells out in detail Transformation 1 and prepares the ground for Transformation 2.

Definition 2 (Enrich)**.** *The transformation $\mathcal{E} \mapsto \mathcal{E}^\sim$ generates the smallest equational theory $\mathcal{E}^\sim = (\Sigma^\sim, E^\sim \cup B^\sim)$ satisfying:*

- $\mathcal{E} \cup \mathcal{E}^{Bool} \subseteq \mathcal{E}^\sim$*;*
- *the poset of sorts of $\mathcal{E}^\sim$ extends that of $\mathcal{E}$ by adding a new connected component $\{Bool\}$, and by adding a fresh top sort to any connected component of the poset of sorts of $\mathcal{E}$ lacking a top sort;*
- *for each top sort k in $\Sigma^\sim$ of a connected component of Σ, $\Sigma^\sim$ contains a commutative operator:*

  $$(\_ \sim \_) : k\ k \to Bool,$$

  $B^\sim$ contains the commutative structural axiom:

  $$x_k \sim y_k = y_k \sim x_k,$$

  and $E^\sim$ contains the equation:

  $$x_k \sim x_k = \top; \tag{17}$$

- *for each top sort k and each* maximal disjoint *pair of sorts s_1, s_2 such that $s_1, s_2 \leq k$, then $E^\sim$ contains the equation:*

  $$x_{s_1} \sim y_{s_2} = \bot; \tag{18}$$

- *for each top sort k in $\Sigma^\sim$ of a connected component of Σ, if there is a function symbol $f : s\ s \to s \in \Omega$, with $s \leq k$ and f satisfies the AC axiom, then $\Sigma^\sim$*

contains the symbol:

$$in^k_f : k\ k \to Bool;$$

- *for each function symbol $f \in F_\Omega$, $E^\sim$ contains the equations* $\mathsf{enrich}_{\mathcal{E}}(f)$ *(see the upcoming definitions in this section).*

As explained earlier, in order-sorted specifications it is possible for a function symbol to have more than one maximal typing. The notion of *subsort-overloading* is necessary to deal with maximal typings of overloaded function symbols. We assume that if two typings are subsort-overloaded they always satisfy the same axioms. The definition of $\mathsf{enrich}_{\mathcal{E}}$ is first given for the case of constructor symbols that do not satisfy any structural axioms; such a function symbol is called *absolutely free*.

Definition 3 (Absolutely Free Enrich)**.** *Assume $f \in \Omega$ is an absolutely free symbol. Then, for each maximal typing $f : s_1 \cdots s_n \to s$ of $f \in \Omega$,* $\mathsf{enrich}_{\mathcal{E}}(f)$ *adds the following equations:*

- *for each $g : s'_1 \cdots s'_m \to s' \in \Omega$ a maximal typing of g such that $s \equiv_\leq s'$, s, s' are not disjoint*[5] *and $f : s_1 \cdots s_n \to s$ is not subsort-overloaded* [6] *with $g : s'_1 \ldots s'_m \to s'$:*

$$f(x^1_{s_1}, \ldots, x^n_{s_n}) \sim g(y^1_{s'_1}, \ldots, y^m_{s'_m}) = \bot; \tag{19}$$

- *for each $f : s'_1 \cdots s'_n \to s' \in \Omega$ a maximal typing of f subsort-overloaded with $f : s_1 \cdots s_n \to s$ and such that s, s' are not disjoint:*

$$f(x^1_{s_1}, \ldots, x^n_{s_n}) \sim f(y^1_{s'_1}, \ldots, y^n_{s'_n}) = \prod_{i=1}^{n} x^i_{s_i} \sim y^i_{s'_i}; \tag{20}$$

- *for each $1 \leq i \leq n$ such that $s_i \equiv_\leq s$:*

$$f(x^1_{s_1}, \ldots, x^n_{s_n}) \sim x^i_{s_i} = \bot. \tag{21}$$

Note that in the second item of Definition 3 any typing is subsort-overloaded with itself. In Definition 3, some equations use the Boolean operator $\sqcap$ in $\mathcal{E}^{Bool}$ to obtain a recursive definition of '$\sim$'. Example 2 shows the transformed theory $\mathcal{E}^\sim$ after applying Definitions 2 and 3 to a concrete specification $\mathcal{E}$.

Example 2. *Consider the following equational theory $\mathcal{E}^{\texttt{NATURAL}}$ specified in Maude [11], that represents the natural numbers in Peano notation, where* 0 *and* s *have been declared as constructors using the* `ctor` *declaration.*

[5] The case when s, s' are disjoint is taken care of by Equation (18).

[6] It could be subsort-overloaded if $f = g$.

```
fmod NATURAL is
  sort Nat .
  op 0 : -> Nat [ctor] .
  op s : Nat -> Nat [ctor] .
endfm
```

An equality enrichment consists of $\mathcal{E}^{\texttt{NATURAL}}$ extended with the equational theory $\mathcal{E}^{Bool}$ and an equational definition of '$\sim$'. The following equational theory $\mathcal{E}^{\texttt{EQ-NATURAL}}$ is an equational enrichment of $\mathcal{E}^{\texttt{NATURAL}}$. The last equation is not essential for getting the proof of Theorem 14, but it is useful in practice for detecting a greater number of inequalities between terms with variables.

```
fmod EQ-NATURAL is
  protecting NATURAL .
  protecting BOOL .
  op _~_ : Nat Nat -> Bool [comm] .
  vars N M : Nat .
  eq N ~ N   = true .
  eq 0 ~ s(N) = false .
  eq s(N) ~ s(M) = N ~ M .
  eq s(N) ~ N  = false .
endfm
```

Definition 4 presents the definition or $\mathsf{enrich}_{\mathcal{E}}$ for the case in which the input symbol is commutative.

Definition 4 (C-Enrich)**.** *Assume $f \in \Omega$ is commutative and non-associative. Then, for each maximal typing $f : s\ s \to s'$ of $f \in \Omega$, $\mathsf{enrich}_{\mathcal{E}}(f)$ adds the following equations:*

- *for each $g : s'_1 \cdots s'_m \to s'' \in \Omega$ a maximal typing of g such that $s' \equiv_{\leq} s''$, s', s'' are not disjoint and $f : s\ s \to s'$ is not subsort-overloaded with $g : s'_1 \cdots s'_m \to s''$:*

$$f(x^1_s, x^2_s) \sim g(y^1_{s'_1}, \ldots, y^m_{s'_m}) = \bot; \tag{22}$$

- *for each $f : s''\ s'' \to s''' \in \Omega$ a maximal typing of f subsort-overloaded with $f : s\ s \to s'$ and such that s', s''' are not disjoint:*

$$f(x^1_s, x^2_s) \sim f(y^1_{s''}, y^2_{s''}) = (x^1_s \sim y^1_{s''} \sqcap x^2_s \sim y^2_{s''}) \sqcup (x^1_s \sim y^2_{s''} \sqcap x^2_s \sim y^1_{s''}); \tag{23}$$

- *if $s \equiv_{\leq} s'$ we add the equation:*

$$f(x^1_s, x^2_s) \sim x^1_s = \bot. \tag{24}$$

Example 3. *We can easily define* unordered pairs *of natural numbers in the following module:*

```
fmod NATURAL-PAIR is
  protecting NATURAL .
  sort Pair .
  op {_,_} : Nat Nat -> Pair [ctor comm] .
endfm
```

we can then define its equality enrichment `EQ-NATURAL-PAIR` *as the following module (we import the already defined equality for naturals in* `EQ-NATURAL`*):*

```
fmod EQ-NATURAL-PAIR is
  protecting NATURAL-PAIR .
  protecting EQ-NATURAL .
  op _~_ : Pair Pair -> Bool [comm] .
  vars N M N' M' : Nat .
  var P : Pair .
  eq P ~ P = true .
  eq {N, M} ~ {N',M'} = (N ~ N' and M ~ M') or
                        (N ~ M' and M ~ N') .
endfm
```

Note that the module `EQ-NATURAL-PAIR` *can be quite useful not only for deciding equality between two unordered pairs of natural numbers, but also to define other useful functions. For example, we could extend* `EQ-NATURAL-PAIR` *with a predicate:*

$$singleton : Pair \to Bool$$

defined by the equation:

$$singleton(\{n, m\}) = n \sim m.$$

For the definition of $\mathsf{enrich}_{\mathcal{E}}$ in the case of an associative function symbol f with maximal typing of sort s, a *top typing* in Σ for such an f is also assumed, i.e., a typing $f : s' \; s' \to s'$ satisfying that if $f : s \; s \to s$ is another typing with $s \equiv_< s'$, then $s' \geq s$ (note that a top typing of f need not belong to Ω, as in Example 4 below).

The notion of *maximal typing with respect to a sort* allows us to avoid the use of conditional equations in the associative case.

Definition 5 (A-Enrich)**.** *Assume $f \in \Omega$ is associative and non-commutative. Then for each maximal typing $f : s \; s \to s$ of $f \in \Omega$,* $\mathsf{enrich}_{\mathcal{E}}(f)$ *adds the following equations:*

- *for each $g : s'_1 \cdots s'_m \to s'$ a maximal typing of $g \in \Omega$ such that $s \equiv_\leq s'$, s, s' are not disjoint and $f : s \; s \to s$ is not subsort-overloaded with $g : s'_1 \cdots s'_m \to s''$:*

$$f(x^1_s, x^2_s) \sim g(y_{s'_1}, \ldots, y^m_{s'_m}) = \bot; \tag{25}$$

- *for each $f : s''' \; s''' \to s''' \in \Omega$ a maximal typing of f subsort-overloaded with $f : s \; s \to s$ and such that s, s''' are not disjoint:*
  - *for each $g : s'_1 \cdots s'_m \to s'$ a maximal typing with respect to s of $g \in \Omega$ and $g : s''_1 \cdots s''_m \to s''$ a maximal typing with respect to s''' of $g \in \Omega$ with*

  * $f : s\ s \to s$ *not subsort-overloaded with* $g : s'_1 \cdots s'_m \to s'$, *and*
  * $g : s'_1 \cdots s'_m \to s'$ *subsort-overloaded with* $g : s''_1 \cdots s''_m \to s''$:

$$f(g(y^1_{s'_1}, \ldots, y^m_{s'_m}), x^1_s) \sim f(g(z^1_{s''_1}, \ldots, z^m_{s''_m}), x^2_{s'''}) = g(y^1_{s'_1}, \ldots, y^m_{s'_m}) \sim g(z^1_{s''_1}, \ldots, z^m_{s''_m}) \sqcap x^1_s \sim x^2_{s'''}; \tag{26}$$

$$f(x^1_s, g(y^1_{s'_1}, \ldots, y^m_{s'_m})) \sim f(x^2_{s'''}, g(z^1_{s''_1}, \ldots, z^m_{s''_m})) = g(y^1_{s'_1}, \ldots, y^m_{s'_m}) \sim g(z^1_{s''_1}, \ldots, z^m_{s''_m}) \sqcap x^1_s \sim x^2_{s'''}; \tag{27}$$

- *for each* $g : s'_1 \cdots s'_m \to s'$ *a maximal typing with respect to* s *of* $g \in \Omega$ *and* $h : s''_1 \cdots s''_l \to s''$ *a maximal typing with respect to* s''' *of* $h \in \Omega$ *with*
  * $f : s\ s \to s$ *not subsort-overloaded with* $g : s'_1 \cdots s'_m \to s'$,
  * $f : s\ s \to s$ *not subsort-overloaded with* $h : s''_1 \cdots s''_l \to s''$, *and*
  * $g : s'_1 \cdots s'_m \to s'$ *not subsort-overloaded with* $h : s''_1 \cdots s''_l \to s''$:

$$f(g(y^1_{s'_1}, \ldots, y^m_{s'_m}), x^1_s) \sim f(h(z^1_{s''_1}, \ldots, z^l_{s''_l}), x^2_{s'''}) = \bot; \tag{28}$$

$$f(x^1_s, g(y^1_{s'_1}, \ldots, y^m_{s'_m})) \sim f(x^2_{s'''}, h(z^1_{s''_1}, \ldots, z^l_{s''_l})) = \bot; \tag{29}$$

- *for any* maximal sort s' *in* $(\lfloor s \rfloor \cap \lfloor s''' \rfloor, \leq)$:

$$f(x^1_{s'}, x^2_s) \sim f(x^1_{s'}, y^2_{s'''}) = x^2_s \sim y^2_{s'''}; \tag{30}$$

$$f(x^1_s, x^2_{s'}) \sim f(y^1_{s'''}, x^2_{s'}) = x^1_s \sim y^1_{s'''}; \tag{31}$$

- *for each* $1 \leq i \leq 2$:

$$f(x^1_s, x^2_s) \sim x^i_s = \bot. \tag{32}$$

Note that subsort-overloading is a transitive property. The above equations mean that equality of two associative terms is performed by recursively comparing their "head elements" (or their "tail elements") using pattern matching.

Example 4. *Consider the following Maude specification of the equational theory* $\mathcal{E}^{\texttt{LIST}}$ *that specifies lists of natural numbers in Peano notation:*

```
fmod LIST is
  protecting NATURAL .
  sorts NeNatList NatList .
  subsorts Nat < NeNatList < NatList .
  op nil : -> NatList [ctor] .
  op _;_ : NeNatList NeNatList -> NeNatList [ctor assoc] .
  op _;_ : NatList NatList -> NatList [assoc] .
  var L : NatList .
  eq L ; nil = L .
  eq nil ; L = L .
endfm
```

Note that ';' is a constructor symbol only when its arguments are non-empty lists. Therefore, the signature of free constructors modulo B of the theory $\mathcal{E}^{\texttt{LIST}}$ *contains the constructors* 0 *and s for* `NATURAL` *plus the operators:*

```
{nil : -> NatList, _;_ : NeNatList NeNatList -> NeNatList}.
```

The equality enrichment of `LIST` *is the following equational theory* $\mathcal{E}^{\texttt{EQ-LIST}}$*:*

```
fmod EQ-LIST is
  protecting LIST .
  protecting EQ-NATURAL .
  op _~_ : NatList NatList -> Bool [comm] .
  vars P Q R S : NeNatList .
  var N M : Nat .
  eq P ~ P = true .
  eq 0 ~ nil = false .
  eq s(N) ~ nil = false .
  eq (P ; Q) ~ 0 = false .
  eq (P ; Q) ~ s(N) = false .
  eq (P ; Q) ~ nil = false .
  eq (P ; Q) ~ P = false .
  eq (P ; Q) ~ Q = false .
  eq (P ; Q) ~ (P ; R) = Q ~ R .
  eq (P ; Q) ~ (R ; Q) = P ~ R .
  eq (0 ; P) ~ (0 ; Q) = P ~ Q .
  eq (P ; 0) ~ (Q ; 0) = P ~ Q .
  eq (s(N) ; P) ~ (s(M) ; Q) = s(N) ~ s(M) and P ~ Q .
  eq (P ; s(N)) ~ (Q ; s(M)) = s(N) ~ s(M) and P ~ Q .
  eq (0 ; P) ~ (s(N) ; Q) = false .
  eq (P ; 0) ~ (Q ; s(N)) = false .
endfm
```

In order to illustrate the intuition behind the definition of $\mathcal{E}^{\texttt{EQ-LIST}}$*, consider the following two examples:*

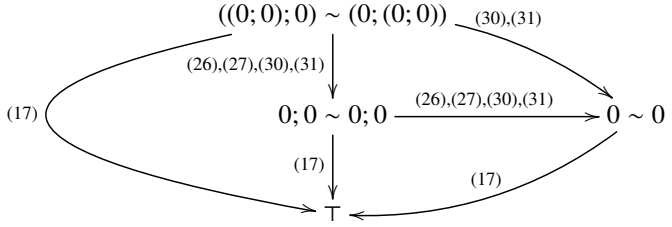

Figure 2: Associative reduction example.

- *for* $((0;0);0) \sim (0;(0;0))$*, Figure 2 illustrates the possible reductions; and*
- *for* $((0;s(0));0) \sim (s(0);0)$ *the only applicable equation modulo associativity*

is $(0;P) \sim (s(N);Q) = \bot$ *under substitution* $\sigma(P) = s(0);0$, $\sigma(Q) = 0$ *and* $\sigma(N) = 0$, *thus obtaining* $((0;s(0));0) \sim (s(0);0) = \bot$.

In the associative-commutative case we assume that there is a *top typing* for f in Σ, as in the associative case.

Definition 6 (AC-Enrich)**.** *Assume* $f \in \Omega$ *is associative-commutative. Then for each maximal typing* $f : s\ s \to s$ *of* $f \in \Omega$, $\mathsf{enrich}_{\mathcal{E}}(f)$ *adds the following equations:*

- *for each* $g : s'_1 \cdots s'_m \to s'$ *a maximal typing of* $g \in \Omega$ *such that* $s \equiv_{\leq} s'$, s, s' *are not disjoint and* $f : s\ s \to s$ *is not subsort-overloaded with* $g : s'_1 \cdots s'_m \to s''$*:*

$$f(x^1_s, x^2_s) \sim g(y^1_{s'_1}, \ldots, y^m_{s'_m}) = \bot; \tag{33}$$

- *for each* $f : s'''\ s''' \to s''' \in \Omega$ *a maximal typing of* f *subsort-overloaded with* $f : s\ s \to s$ *and such that* s, s''' *are not disjoint:*

  - *for each* $g : s'_1 \cdots s'_m \to s'$ *a maximal typing with respect to* s *of* $g \in \Omega$ *and* $f : s\ s \to s$ *is not subsort-overloaded with* $g : s'_1 \cdots s'_m \to s''$*:*

$$\begin{aligned} f(g(x^1_{s'_1}, \ldots, x^m_{s'_m}), z^2_s) \sim f(y^1_{s'''}, y^2_{s'''}) = \bot \\ \textbf{if } in^k_f(g(x^1_{s'_1}, \ldots, x^m_{s'_m}), y^1_{s'''}) \\ \sqcup\, in^k_f(g(x^1_{s'_1}, \ldots, x^m_{s'_m}), y^2_{s'''}) = \bot \end{aligned} \tag{34}$$

  - *for any* maximal sort $s' \in (\lfloor s \rfloor \cap \lfloor s''' \rfloor)$*:*

$$f(x^1_{s'}, x^2_s) \sim f(x^1_{s'}, y^2_{s'''}) = x^2_s \sim y^2_{s'''}; \tag{35}$$

- *for each* $g : s'_1 \cdots s'_m \to s'$ *a maximal typing of* $g \in \Omega$ *such that* $s \equiv_{\leq} s'$ *and* $f : s\ s \to s$ *is not subsort-overloaded with* $g : s'_1 \cdots s'_m \to s''$*:*

$$\begin{aligned} in^k_f(g(x^1_{s'_1}, \ldots, x^m_{s'_m}), f(z^1_s, z^2_s)) = in^k_f(g(x^1_{s'_1}, \ldots, x^m_{s'_m}), z^1_s) \\ \sqcup\, in^k_f(g(x^1_{s'_1}, \ldots, x^m_{s'_m}), z^2_s); \end{aligned} \tag{36}$$

- *for each* $g : s'_1 \cdots s'_m \to s'$ *a maximal typing of* $g \in \Omega$ *and* $g : s''_1 \cdots s''_m \to s''$ *a maximal typing of* $g \in \Omega$ *such that* $s \equiv_{\leq} s'$, $s \equiv_{\leq} s''$ *where*

  - $f : s\ s \to s$ *is not subsort-overloaded with* $g : s'_1 \cdots s'_m \to s'$*, and*
  - $g : s'_1 \cdots s'_m \to s'$ *is subsort-overloaded with* $g : s''_1 \cdots s''_m \to s''$*:*

$$\begin{aligned} in^k_f(g(x^1_{s'_1}, \ldots, x^m_{s'_m}), g(y^1_{s''_1}, \ldots, y^m_{s''_m})) = \\ g(x^1_{s'_1}, \ldots, x^m_{s'_m}) \sim g(y^1_{s''_1}, \ldots, y^m_{s''_m}); \end{aligned} \tag{37}$$

- *for each* $g : s'_1 \cdots s'_m \to s'$ *a maximal typing of* $g \in \Omega$ *and* $h : s''_1 \cdots s''_l \to s''$ *a maximal typing of* $h \in \Omega$ *such that* $s \equiv_{\leq} s'$, $s \equiv_{\leq} s''$ *where*

– $f : s\ s \to s$ *is not subsort-overloaded with* $g : s'_1 \cdots s'_m \to s'$,
– $f : s\ s \to s$ *is not subsort-overloaded with* $h : s''_1 \cdots s''_l \to s''$, *and*
– $g : s'_1 \cdots s'_m \to s'$ *is not subsort-overloaded with* $h : s''_1 \cdots s''_l \to s''$:

$$in^k_f(g(x^1_{s'_1}, \ldots, x^m_{s'_m}), h(y^1_{s''_1}, \ldots, y^l_{s''_l})) = \bot; \tag{38}$$

- *for* f *itself:*

$$in^k_f(f(x^1_s, x^2_s), y_k) = \bot; \tag{39}$$

$$f(x^1_s, x^2_s) \sim x^1_s = \bot. \tag{40}$$

Intuitively, a constructor term rooted by an associative-commutative symbol f can be viewed as a multiset with union operator f. The function 'in^k_f' in Definition 6 identifies when an element (a term not rooted by f) belongs to such a multiset.

Example 5. *Consider the following Maude specification of the equational theory* $\mathcal{E}^{\texttt{MSET}}$*, which represents multisets of natural numbers in Peano notation:*

```
fmod MSET is
  protecting NATURAL .
  sorts NeNatMSet NatMSet .
  subsort Nat < NeNatMSet < NatMSet .
  op empty : -> NatMSet [ctor] .
  op __  : NeNatMSet NeNatMSet -> NeNatMSet [ctor assoc comm] .
  op __  : NatMSet NatMSet -> NatMSet [assoc comm] .
  var T : NatMSet .
  eq empty T = T .
endfm
```

The membership function 'in^k_f' is used as an auxiliary function to give a recursive definition of equality for constructor terms rooted by AC-symbols. The following equational theory $\mathcal{E}^{\texttt{EQ-MSET}}$ *is the equality enrichment for* $\mathcal{E}^{\texttt{MSET}}$*:*

```
fmod EQ-MSET is
  protecting MSET .
  protecting EQ-NATURAL .
  op in--  : NatMSet NatMSet -> Bool .
  op _~_ : NatMSet NatMSet -> Bool [comm] .
  vars P Q R S : NeNatMSet .
  var N M : Nat .
  vars T U : NatMSet .
  eq in--(P Q, R) = false .
  eq in--(0, 0) = true .
  eq in--(s(N), s(M)) = s(N) ~ s(M) .
  eq in--(empty, empty) = true .
  eq in--(0, s(N)) = false .
```

```
eq in--(s(N), 0) = false .
eq in--(0, empty) = false .
eq in--(empty, 0) = false .
eq in--(s(N), empty) = false .
eq in--(empty, s(N)) = false .
eq in--(0, (P Q)) = in--(0,P) or in--(0, Q) .
eq in--(s(N), (P Q)) = in--(s(N), P) or in--(s(N), Q) .
eq in--(empty, (P Q)) = in--(empty, P) or in--(empty, Q) .
eq P ~ P = true .
eq 0 ~ empty = false .
eq s(N) ~ empty = false .
eq (P Q) ~ 0 = false .
eq (P Q) ~ empty = false .
eq (P Q) ~ s(N) = false .
eq (P Q) ~ P = false .
eq (P Q) ~ (P R) = Q ~ R .
ceq (0 P) ~ (Q R) = false if in--(0, Q) or in--(0, R) = false .
ceq (s(N) P) ~ (Q R) = false if in--(s(N), Q)
                            or in--(s(N), R) = false .
```

endfm

In order to illustrate the intuition behind the definition of $\mathcal{E}^{\texttt{EQ-MSET}}$*, consider the following two examples:*

- *for* $((0\ 0)\ 1) \sim ((0\ 1)\ 0)$*, Figure 3 illustrates the possible reductions; and*

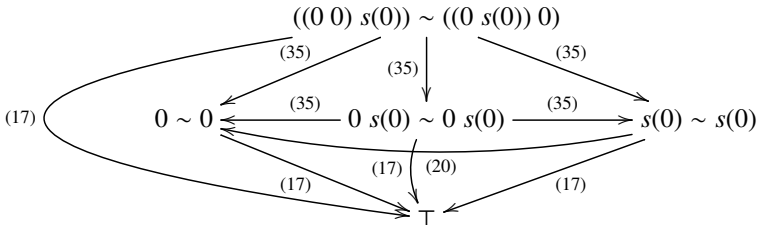

Figure 3: Associative-Commutative reduction example.

- *for* $((0\ s(0))\ 0) \sim (s(0)\ 0)$*, Figure 4 illustrates the possible reductions.*

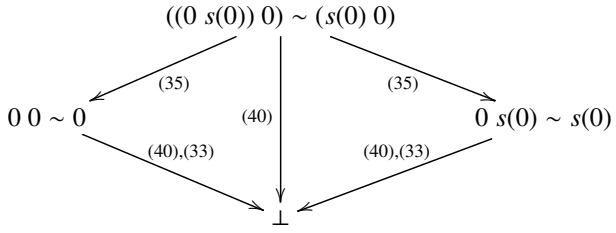

Figure 4: Associative-Commutative reduction example.

5. Executability Properties of $\mathcal{E}^{\sim}$

It would be very useful, both from a theoretical and a practical point of view, that if a theory $\mathcal{E}$ satisfies some executability properties, then the equality enrichment $\mathcal{E}^{\sim}$ of $\mathcal{E}$ obtained from the transformation in Section 4 would inherit these properties. In particular, if the original theory $\mathcal{E}$ is sort-decreasing (resp., ground sort-decreasing), confluent (resp., ground confluent), and operationally terminating (resp., ground operationally terminating), then $\mathcal{E}^{\sim}$ should also be so. Similarly, the subsignature of free constructors of $\mathcal{E}^{\sim}$ should be an extension of the subsignature of free constructors of $\mathcal{E}$ (modulo the structural axioms). In this way, full agreement between mathematical and operational semantics would be preserved in the equality enrichment $\mathcal{E}^{\sim}$.

The domain of the transformation $\mathcal{E} \mapsto \mathcal{E}^{\sim}$ includes exactly those equational theories whose structural axioms are any combination of A and/or C axioms for some of its symbols. However, if the input theory $\mathcal{E}$ has symbols with identity axioms, one could use the results in [3] to avoid these axioms and instead use identity equations, provided that the constructors remain free after the transformation. This is often possible in practice, as illustrated by the `LIST` and `MSET` examples, where identities for lists and multisets are specified as oriented equations and not as structural axioms.

In this section, $\mathcal{E} = (\Sigma, E \cup B)$ is an order-sorted equational theory with $\Sigma = (S, \leq, F)$ and Ω a signature of free constructors modulo B, with $\mathcal{E}^{\sim} = (\Sigma^{\sim}, E^{\sim} \cup B^{\sim})$ the Boolean equality enrichment obtained by the transformation $\mathcal{E} \mapsto \mathcal{E}^{\sim}$, with order-sorted signature $\Sigma^{\sim} = (S^{\sim}, \leq^{\sim}, F^{\sim})$. We assume that the axioms B are any combination of A and/or C axioms for some of the function symbols in Σ. Definitions and results on the termination of Boolean Theories relevant to the main results in Section 5.2 and on the modular decomposition of B-equality relevant to the main results in Section 5.3 can be found in the companion appendix, electronically available with this paper.

5.1. Preservation of (Ground) Sort-Decreasingness

Recall from Section 2 that the equational theory $\mathcal{E}$ is sort-decreasing (resp., ground sort-decreasing) iff $\mathcal{E} \vdash C\theta$ implies $ls(t\theta) \geq ls(u\theta)$ for each $t = u$ **if** $C \in E$ and substitution (resp., ground substitution) θ. The key observation for the proof is that since *Bool* is a fresh sort in a new connected component of $\mathcal{E}^{\sim}$ and all the new equations added in the transformation $\mathcal{E} \mapsto \mathcal{E}^{\sim}$ are of sort *Bool*, it is impossible for the equations added by Definition 2 to apply to terms in $T_{\Sigma}(X)$.

Proposition 2. *If $\mathcal{E}$ is sort-decreasing (resp., ground sort-decreasing), then $\mathcal{E}^{\sim}$ is also sort-decreasing (resp., ground sort-decreasing). Furthermore, since we have assumed throughout that $\mathcal{E}$ is preregular modulo B, $\mathcal{E}^{\sim}$ is also preregular modulo $B^{\sim}$.*

Proof. Consider the following cases on the equations in $\mathcal{E}^{\sim}$:

1. any equation in $\mathcal{E}$ is sort-decreasing (resp., ground sort-decreasing) by assumption,
2. if an equation is in $\mathcal{E}^{\sim}$ but not in $\mathcal{E}$, then the left-hand and right-hand sides of such an equation have least sort *Bool* because *Bool* has no proper subsorts or supersorts.

Therefore, $\mathcal{E}^{\sim}$ is sort-decreasing (resp., ground sort-decreasing). Finally, note that all axioms in $B^{\sim} - B$ are of sort *Bool*, which has no subsorts or supersorts, and therefore are trivially sort-preserving. □

5.2. Preservation of (Ground) Operational Termination

The operational (resp., ground operational) termination of $\mathcal{E}^{\sim}$ is obtained from the operational (resp. ground operational) termination of some of its subtheories, including that of the input theory $\mathcal{E}$. In general, the union of two operationally terminating theories may not be operationally terminating. Furthermore, the fact of dealing with arbitrary theories whose rules are unknown makes proving operational termination of the union more involved. However, using a modular argument that exploits sort information to take advantage of the shape of terms in $\mathcal{E}^{\sim}$, it is possible to prove that there is no possible infinite well-formed proof tree in the theory $\mathcal{E}^{\sim}$. The proof of operational termination is based on the conditional order-sorted rewriting logic modulo axioms defined in Figure 5, under the very general assumption that any rewrite system associated to a finite set of conditional equations is a deterministic 3-CTRS [15]. The main result of this section is stated in Theorem 7.

$$(\textit{Refl}) \quad \frac{}{t \rightarrow^* u} \quad \text{if } t =_B u \qquad (\textit{Cong}) \quad \frac{t_i \rightarrow u_i}{f(t_1, \ldots, t_i, \ldots, t_n) \rightarrow f(t_1, \ldots, u_i, \ldots, t_n)} \quad \text{where } f \in \Sigma \text{ and } 1 \leq i \leq ar(f)$$

$$(\textit{Tran}) \quad \frac{t \rightarrow u \quad u \rightarrow^* v}{t \rightarrow^* v} \qquad (\textit{Repl}) \quad \frac{u_1\sigma \rightarrow^* v_1\sigma \cdots u_m\sigma \rightarrow^* v_m\sigma}{t \rightarrow v\sigma}$$

where $u \rightarrow v$ **if** $u_1 \rightarrow v_1 \cdots u_m \rightarrow v_m \in R$ and $t =_B u\sigma$

Figure 5: Conditional Order-Sorted Rewriting Logic Modulo.

Our strategy for proving the operational termination of $\mathcal{E}^{\sim}$ relies on the (assumed) operational termination of $\mathcal{E}$ and on proofs for the operational termination of the following equational subtheories of $\mathcal{E}^{\sim}$:

$$\mathcal{E}_0 = (\Sigma^{\sim}, E_0 \cup B^{\sim}) \text{ and} \tag{41}$$

$$\mathcal{E}_1 = (\Sigma^{\sim}, E_1 \cup B^{\sim}), \tag{42}$$

where $E_1 = (E^{\sim} \setminus E)$ and $E_0 = \{l = r \mid l = r \textbf{ if } C \in E_1\}$. The equational theory $\mathcal{E}_1$ forgets the equations in $\mathcal{E}$, while the equational theory $\mathcal{E}_0$ is the same as $\mathcal{E}_1$ but it also drops the conditions of the conditional equations in E_1.

Lemma 2 states that $\mathcal{E}_0$ is operationally terminating. In this case, since $\mathcal{E}_0$ is unconditional, operational termination of $\mathcal{E}_0$ is a logical consequence of the well-foundedness of $\rightarrow_{E_0/B^{\sim}}$, which is the main concern in the proof of the lemma.

Lemma 2. *The rewrite relation $\rightarrow_{E_0/B^\sim}$ is well-founded. Moreover, $\mathcal{E}_0$ is operationally terminating.*

Proof. It is enough to prove that $\mathcal{E}_0$ is terminating regardless of the sort information. The symmetric, stable, and monotonic relation $\sim$ is witnessed by $=_{B^\sim}$. The $B^\sim$-compatible simplification ordering is witnessed by $\succ$ [16], that can be obtained using an AC-RPO [17] which, depending on the choice of either the basic Boolean theory presented in Example 1, Hsiang's Boolean ring theory or Dijkstra-Schoelten's theory, uses one of the precedence orders $>_{BB}$, $>_{BR}$, or $>_{DS}$ between symbols that can be found in the companion appendix (Appendix A), electronically available with this paper. Simplification orderings imply $\rhd \subseteq \succ$, where $\rhd$ is the subterm ordering. It is routine to check by inspection on the equations E_0 (including the Boolean equations for the Boolean theories we have considered) that $\mathcal{E}_0$ is terminating modulo $B^\sim$. Furthermore, since $\mathcal{E}_0$ is unconditional, well-foundedness of $\rightarrow_{E_0/B^\sim}$ is equivalent to the operational termination of $\mathcal{E}_0$ [18, p. 79]. □

Corollary 3. *The rewrite relation $\rightarrow_{E_1/B^\sim}$ is well-founded.*

Proof. It follows by Lemma 2 and observing that $\rightarrow_{E_1/B^\sim} \subseteq \rightarrow_{E_0/B^\sim}$. □

The next subgoal is to extend the well-foundedness of $\mathcal{E}_1$'s rewrite relation obtained in Corollary 3 to a proof of the operational termination of $\mathcal{E}_1$. For this purpose, it is convenient to reason about the more general binary relations $\Rightarrow_{E_0/B^\sim}$ and $\Rightarrow_{E_1/B^\sim}$ on $T_{\Sigma^\sim/B^\sim}(X)_{Bool}$, which are compatible with the subterm relation, and are defined by

$$\Rightarrow_{E_0/B^\sim} = \rightarrow_{E_0/B^\sim} \cup \rhd_{B^\sim} \tag{43}$$

$$\Rightarrow_{E_1/B^\sim} = \rightarrow_{E_1/B^\sim} \cup \rhd_{B^\sim} . \tag{44}$$

where $\rhd_{B^\sim}$ is defined by:

$$u \rhd_{B^\sim} v \iff \exists w.\, w \in [u]_{B^\sim} \wedge w \rhd v \wedge v \in T_\Sigma(X)_{Bool}$$

where, by definition, $w \rhd v$ holds iff v is a proper subterm of w. Note that $\Rightarrow_{E_1/B^\sim} \subseteq \Rightarrow_{E_0/B^\sim}$. Also, these relations are well-founded.

Lemma 3. *The binary relations $\Rightarrow_{E_0/B^\sim}$ and $\Rightarrow_{E_1/B^\sim}$ are well-founded.*

Proof. Note that $\rhd_{B^\sim}, \rightarrow_{E_0/B^\sim} \subseteq \succ$, where $\succ$ is the $B^\sim$-compatible simplification ordering in the proof of Lemma 2. Therefore, $\Rightarrow_{E_0/B^\sim} \subseteq \succ$ and $\Rightarrow_{E_0/B^\sim}$ is well-founded because $\succ$ is well-founded. Also, since $\Rightarrow_{E_1/B^\sim} \subseteq \Rightarrow_{E_0/B^\sim}$, $\Rightarrow_{E_1/B^\sim}$ is also well-founded. □

Definition 7 introduces a notion of measure μ on $\Sigma^\sim$-terms that takes into account the size of maximal Σ-terms modulo the axioms $B^\sim$.

Definition 7 (Measure)**.** *Let $\hat{\mu} : T_{\Sigma^\sim/B^\sim}(X) \longrightarrow \wp_{fin}(\mathbb{N})$ be the mapping defined for any $[t]_{B^\sim} \in T_{\Sigma^\sim/B^\sim}(X)$ by:*

$$\hat{\mu}\,([t]_{B^\sim}) = \begin{cases} \{\,|t|\,\} & , \text{if } t \notin T_{\Sigma^\sim}(X)_{Bool} \\ \bigcup_{i=1}^{n} \hat{\mu}\,([t_i]_{B^\sim}) & , \text{if } t = f(t_1, \ldots, t_n) \in T_{\Sigma^\sim}(X)_{Bool} \end{cases} \tag{45}$$

where $|\_| : T_{\Sigma^{\sim}}(X) \longrightarrow \mathbb{N}$ *is the usual size function on terms. Note that, since the axioms in* $B^{\sim}$ *are any combination of A and/or C axioms, they preserve the size of terms in* $T_{\Sigma^{\sim}/B^{\sim}}(X)$ *so that* $\hat{\mu}$ *is well-defined. Furthermore, let* $\mu : T_{\Sigma^{\sim}/B^{\sim}}(X) \longrightarrow \mathbb{N}$ *be the mapping defined by*

$$\mu = \hat{\mu}\,; \max, \tag{46}$$

where $\max : \wp_{fin}(\mathbb{N}) \longrightarrow \mathbb{N}$ *is the function denoting the maximum element of a nonempty finite set of natural numbers or 0 otherwise.*

Lemma 4 ($(\Rightarrow_{E_0/B^{\sim}}, \mu)$-compatibility)**.** *Let* $t, u \in T_{\Sigma^{\sim}}(X)$*. If* $[t]_{B^{\sim}} \Rightarrow_{E_0/B^{\sim}} [u]_{B^{\sim}}$*, then* $\mu([t]_{B^{\sim}}) \geq \mu([u]_{B^{\sim}})$*.*

Proof. Since $\Rightarrow_{E_0/B^{\sim}}$ is $B^{\sim}$-compatible, because both $\rightarrow_{E_0/B^{\sim}}$ and $\rhd_{B^{\sim}}$ are, it induces a binary relation on $B^{\sim}$ equivalence classes, denoted $\Rightarrow_{E_0/B^{\sim}} \subseteq T^2_{\Sigma^{\sim}/B^{\sim}}(X)$, by $[t]_{B^{\sim}} \Rightarrow_{E_0/B^{\sim}} [u]_{B^{\sim}} \iff t \Rightarrow_{E_0/B^{\sim}} u$. If $[t]_{B^{\sim}} \Rightarrow_{E_0/B^{\sim}} [u]_{B^{\sim}}$, then either $[t]_{B^{\sim}} \rightarrow_{E_0/B^{\sim}} [u]_{B^{\sim}}$ or $[t]_{B^{\sim}} \rhd_{B^{\sim}} [u]_{B^{\sim}}$. If $[t]_{B^{\sim}} \rightarrow_{E_0/B^{\sim}} [u]_{B^{\sim}}$, then it is routine to check by inspection on the equations E_0 that $\mu([t]_{B^{\sim}}) \geq \mu([u]_{B^{\sim}})$. If $[t]_{B^{\sim}} \rhd_{B^{\sim}} [u]_{B^{\sim}}$, then since $v' \in [v]_{B^{\sim}}$ implies $|v'| = |v|$, clearly $\mu([t]_{B^{\sim}}) \geq \mu([u]_{B^{\sim}})$. □

Let $\Rrightarrow_{E_1/B^{\sim}}$ be the binary relation on $T_{\Sigma^{\sim}/B^{\sim}}(X)$ defined by

$$\Rrightarrow_{E_1/B^{\sim}} = \Rightarrow_{E_1/B^{\sim}} \cup \rightsquigarrow_{E_1/B^{\sim}}, \tag{47}$$

where

$$\rightsquigarrow_{E_1/B^{\sim}} = \Big\{([l\sigma]_{B^{\sim}}, [l_1\sigma]_{B^{\sim}}) \in T_{\Sigma^{\sim}/B^{\sim}}(X)^2 \mid \sigma \in [X \rightarrow T_{\Sigma^{\sim}}(X)] \wedge (l \rightarrow r \textbf{ if } l_1 \rightarrow r_1 \in E_1)\Big\}.$$

Lemma 5. *The binary relation* $\Rrightarrow_{E_1/B^{\sim}}$ *is well-founded.*

Proof. Suppose not. Then we must have an infinite chain of steps of the form:

$$[u_1]_{B^{\sim}} \left\{ \begin{array}{c} \rightarrow_{E_1/B^{\sim}} \\ \rhd_{B^{\sim}} \\ \rightsquigarrow_{E_1/B^{\sim}} \end{array} \right\} [u_2]_{B^{\sim}} \left\{ \begin{array}{c} \rightarrow_{E_1/B^{\sim}} \\ \rhd_{B^{\sim}} \\ \rightsquigarrow_{E_1/B^{\sim}} \end{array} \right\} [u_3]_{B^{\sim}} \left\{ \begin{array}{c} \rightarrow_{E_1/B^{\sim}} \\ \rhd_{B^{\sim}} \\ \rightsquigarrow_{E_1/B^{\sim}} \end{array} \right\} [u_4]_{B^{\sim}} \left\{ \begin{array}{c} \rightarrow_{E_1/B^{\sim}} \\ \rhd_{B^{\sim}} \\ \rightsquigarrow_{E_1/B^{\sim}} \end{array} \right\} \cdots$$

Let $\mathcal{T}_\infty$ be the collection of equivalence classes that generate an infinite sequence on $\Rrightarrow_{E_1/B^{\sim}}$. By definition, the relation $\Rrightarrow_{E_1/B^{\sim}}$ is well-founded iff $\mathcal{T}_\infty = \varnothing$. We assume $\mathcal{T}_\infty \neq \varnothing$ and reason by contradiction. Also, let $\mathcal{T}_\infty^{\Rrightarrow}$ and $\mathcal{T}_\infty^{\Rrightarrow,\mu}$ be the subsets of $\mathcal{T}_\infty$ defined by:

$$\begin{aligned} \mathcal{T}_\infty^{\Rrightarrow} &= \left\{[t]_{B^{\sim}} \in \mathcal{T}_\infty \mid [t]_{B^{\sim}} \text{ is } \left(\Rightarrow^{+}_{E_1/B^{\sim}}\right)\text{-minimal in } \mathcal{T}_\infty\right\} \text{ and} \\ \mathcal{T}_\infty^{\Rrightarrow,\mu} &= \left\{[t]_{B^{\sim}} \in \mathcal{T}_\infty^{\Rrightarrow} \mid (\forall [u]_{B^{\sim}} \in \mathcal{T}_\infty^{\Rrightarrow})\, \mu([t]_{B^{\sim}}) \leq \mu([u]_{B^{\sim}})\right\}. \end{aligned}$$

First note that, by definition, we have $\mathcal{T}_\infty^{\Rrightarrow,\mu} \subseteq \mathcal{T}_\infty^{\Rrightarrow} \subseteq \mathcal{T}_\infty$. Also note that since $\Rightarrow_{E_1/B^{\sim}}$ is well-founded, $\mathcal{T}_\infty \neq \varnothing$ implies $\mathcal{T}_\infty^{\Rrightarrow} \neq \varnothing$. Similarly, since the poset $\langle \mathbb{N}, \leq \rangle$

is well-founded, $\mathcal{T}_\infty^{\Rightarrow} \neq \varnothing$ implies $\mathcal{T}_\infty^{\Rightarrow,\mu} \neq \varnothing$. Hence, we can choose $[t]_{B^\sim} \in T_{\Sigma^\sim/B^\sim}(X)$ such that $[t]_{B^\sim} \in \mathcal{T}_\infty^{\Rightarrow,\mu}$. Reasoning by cases:

1. If $[t]_{B^\sim} \rightarrow_{E_1/B^\sim} [t']_{B^\sim}$ and $[t']_{B^\sim} \in \mathcal{T}_\infty$, then $[t]_{B^\sim}$ is not $\left(\Rightarrow^+_{E_1/B^\sim}\right)$-minimal in $\mathcal{T}_\infty$ because $\rightarrow_{E_1/B^\sim} \subseteq \Rightarrow^+_{E_1/B^\sim}$, a contradiction.
2. If $[t]_{B^\sim} \rhd_{B^\sim} [t']_{B^\sim}$ and $[t']_{B^\sim} \in \mathcal{T}_\infty$, then $[t]_{B^\sim}$ is not $\left(\Rightarrow^+_{E_1/B^\sim}\right)$-minimal in $\mathcal{T}_\infty$ because $\rhd_{B^\sim} \subseteq \Rightarrow^+_{E_1/B^\sim}$, a contradiction.
3. If $[t]_{B^\sim} \rightsquigarrow_{E_1/B^\sim} [t']_{B^\sim}$ and $[t']_{B^\sim} \in \mathcal{T}_\infty$, then a pair of the form:

$$\Big(f(g(x^1_{s'_1},\ldots,x^m_{s'_m}),z^2_s)\sigma \sim f(y^1_{s'''},y^2_{s'''})\sigma \;,\; in^k_f(g(x^1_{s'_1},\ldots,x^m_{s'_m}),y^1_{s'''})\sigma \sqcup in^k_f(g(x^1_{s'_1},\ldots,x^m_{s'_m}),y^2_{s'''})\sigma = \bot\Big)$$

is applied. Therefore,

$$t =_{B^\sim} f(g(x^1_{s'_1},\ldots,x^m_{s'_m}),z^2_s)\sigma \sim f(y^1_{s'''},y^2_{s'''})\sigma$$

with f in Σ, and $t' =_{B^\sim} u_1\sigma$ with

$$u_1 = in^k_f(g(x^1_{s'_1},\ldots,x^m_{s'_m}),y^1_{s'''}) \sqcup in^k_f(g(x^1_{s'_1},\ldots,x^m_{s'_m}),y^2_{s'''}) = \bot$$

Note that $\mu([t]_{B^\sim}) > \mu([u_1\sigma]_{B^\sim})$ because f is in Σ and we must have that $[u_1\sigma]_{B^\sim} \in \mathcal{T}_\infty$. Therefore, there is $[w]_{B^\sim} \in T_\infty^{\Rightarrow}$ satisfying $[u_1\sigma] \Rightarrow^*_{E_1/B^\sim} [w]$ because $\Rightarrow_{E_1/B^\sim}$ is well-founded. By Lemma 4 and since $\Rightarrow_{E_1/B^\sim} \subseteq \Rightarrow_{E_0/B^\sim}$, $\mu([u_1\sigma]_{B^\sim}) \geq \mu([w]_{B^\sim})$ and hence $\mu([t]_{B^\sim}) > \mu([w]_{B^\sim})$, i.e., $[t]_{B^\sim} \notin \mathcal{T}_\infty^{\Rightarrow,\mu}$, a contradiction.

This exhausts all the possible cases showing that $\mathcal{T}_\infty = \varnothing$. □

We assume that the operational termination (resp., ground operational termination) of $(\Sigma, E \cup B)$ is Σ-*extensible*, i.e., if $(\Sigma, E \cup B)$ is operationally terminating (resp., ground operationally terminating) then $(\Sigma \cup \Delta, E \cup (B \cup B^\Delta))$ is so too for any order-sorted signature Δ disjoint from Σ and A, C, and/or AC structural axioms B^Δ for some function symbols in Δ. This is not a strong restriction in practice, since all the actual existing tools for proving termination properties on rewriting theories generate Σ-extensible orderings.

Corollary 4. *If $\mathcal{E}$ is sort-decreasing (resp., ground sort-decreasing), confluent (resp., ground confluent), and operationally terminating (resp., ground operationally terminating) in a Σ-extensible way modulo B, then $(\Sigma^\sim, E \cup B^\sim)$ is operationally terminating (resp., ground operationally terminating) and confluent (resp., ground confluent).*

Proof. Since $(\Sigma, E \cup B)$ is Σ-extensible, $(\Sigma^\sim, E \cup B^\sim)$ is (ground) operationally terminating. Therefore, its (ground) confluence is equivalent to its (ground) local confluence. But, local confluence is straightforward, because no new critical pairs appear and symbols from $\Sigma^\sim \setminus \Sigma$ cannot appear in the existing critical pairs (which are joinable or unfeasible by hypothesis). □

Let $\succ_{E^{\sim}/B^{\sim}}$ be the binary relation on $T_{\Sigma^{\sim}/B^{\sim}}(X)$ defined by

$$\succ_{E^{\sim}/B^{\sim}} = \rightarrow_{E/B^{\sim}} \cup \Rrightarrow_{E_1/B^{\sim}} . \tag{48}$$

Lemma 6. *Let $t, u \in T_{\Sigma^{\sim}}(X)_{Bool}$. If $[t]_{B^{\sim}} \Rrightarrow_{E_1/B^{\sim}} [u]_{B^{\sim}}$ then $[t \downarrow_{E/B^{\sim}}]_{B^{\sim}} \Rrightarrow_{E_1/B^{\sim}} [u \downarrow_{E/B^{\sim}}]_{B^{\sim}}$.*

Proof. Note that, by definition, $\Rrightarrow_{E_1/B^{\sim}} = \rightarrow_{E_1/B^{\sim}} \cup \rhd_{B^{\sim}} \cup \rightsquigarrow_{E_1/B^{\sim}}$ and t, u has the form $C[t_1, \ldots, t_n]$, where $C \in T_{(\Sigma^{\sim}\setminus\Sigma)\cup\Omega\cup\{\Box\}}(X)_{Bool}$ is a non-empty context and either $root(t_i) \in \Sigma \setminus \Omega$ or, only in the non-ground case, $t_i \in X_s$, for some $s \in S$. We reason by cases:

1. If $[t]_{B^{\sim}} \rightarrow_{E_1/B^{\sim}} [u]_{B^{\sim}}$ then
   (a) $[t]_{B^{\sim}} = [C[t_1, \ldots, t_n]]_{B^{\sim}} \rightarrow_{E_1/B^{\sim}} [C'[t'_1, \ldots, t'_m]]_{B^{\sim}} = [u]_{B^{\sim}}$ where the context $C' \in T_{(\Sigma^{\sim}\setminus\Sigma)\cup\Omega\cup\{\Box\}}(X)_{Bool}$ is non-empty, and
   (b) $\{[t'_1]_{B^{\sim}}, \ldots, [t'_m]_{B^{\sim}}\} \subseteq \{[t_1]_{B^{\sim}}, \ldots, [t_n]_{B^{\sim}}\}$.

   Therefore,

$$\begin{array}{ccc} [t \downarrow_{E/B^{\sim}}]_{B^{\sim}} & \rightarrow_{E_1/B^{\sim}} & [u \downarrow_{E/B^{\sim}}]_{B^{\sim}} \\ = & & = \\ [C[t_1 \downarrow_{E/B^{\sim}}, \ldots, t_n \downarrow_{E/B^{\sim}}]]_{B^{\sim}} & \rightarrow_{E_1/B^{\sim}} & [C'[t'_1 \downarrow_{E/B^{\sim}}, \ldots, t'_m \downarrow_{E/B^{\sim}}]]_{B^{\sim}} \end{array}$$

   and the following schema is obtained:

$$\begin{array}{ccc} [t]_{B^{\sim}} & \xrightarrow[E_1/B^{\sim}]{} & [u]_{B^{\sim}} \\ {\scriptstyle E/B^{\sim}}\downarrow{\scriptstyle !} & & \downarrow{\scriptstyle E/B^{\sim}}{\scriptstyle !} \\ [t \downarrow_{E/B^{\sim}}]_{B^{\sim}} & \xrightarrow[E_1/B^{\sim}]{} & [u \downarrow_{E/B^{\sim}}]_{B^{\sim}} \end{array}$$

2. If $[t]_{B^{\sim}} \rhd_{B^{\sim}} [u]_{B^{\sim}}$ then
   (a) $[t]_{B^{\sim}} = [C[t_1, \ldots, t_n]]_{B^{\sim}} \rhd_{B^{\sim}} [C'[t'_1, \ldots, t'_m]]_{B^{\sim}} = [u]_{B^{\sim}}$ where the context $C' \in T_{(\Sigma^{\sim}\setminus\Sigma)\cup\Omega\cup\{\Box\}}(X)_{Bool}$ is non-empty (because u has sort *Bool*), and
   (b) $\{[t'_1]_{B^{\sim}}, \ldots, [t'_m]_{B^{\sim}}\} \subseteq \{[t_1]_{B^{\sim}}, \ldots, [t_n]_{B^{\sim}}\}$.

   Therefore,

$$\begin{array}{ccc} [t \downarrow_{E/B^{\sim}}]_{B^{\sim}} & \rhd_{B^{\sim}} & [u \downarrow_{E/B^{\sim}}]_{B^{\sim}} \\ = & & = \\ [C[t_1 \downarrow_{E/B^{\sim}}, \ldots, t_n \downarrow_{E/B^{\sim}}]]_{B^{\sim}} & \rhd_{B^{\sim}} & [C'[t'_1 \downarrow_{E/B^{\sim}}, \ldots, t'_m \downarrow_{E/B^{\sim}}]]_{B^{\sim}} \end{array}$$

   and the following schema is obtained:

$$\begin{array}{ccc} [t]_{B^{\sim}} & \rhd_{B^{\sim}} & [u]_{B^{\sim}} \\ {\scriptstyle E/B^{\sim}}\downarrow{\scriptstyle !} & & \downarrow{\scriptstyle E/B^{\sim}}{\scriptstyle !} \\ [t \downarrow_{E/B^{\sim}}]_{B^{\sim}} & \rhd_{B^{\sim}} & [u \downarrow_{E/B^{\sim}}]_{B^{\sim}} \end{array}$$

3. If $[t]_{B^{\sim}} \rightsquigarrow_{E_1/B^{\sim}} [u]_{B^{\sim}}$ then
   (a) $[t]_{B^{\sim}} = [C[t_1, \ldots, t_n]]_{B^{\sim}} \rightsquigarrow_{E_1/B^{\sim}} [C'[t'_1, \ldots, t'_m]]_{B^{\sim}} = [u]_{B^{\sim}}$ where $C' \in T_{(\Sigma^{\sim}\setminus\Sigma)\cup\Omega\cup\{\Box\}}(X)_{Bool}$ is a non-empty context, and

(b) $\{[t'_1]_{B^\sim}, \ldots, [t'_m]_{B^\sim}\} \subseteq \{[t_1]_{B^\sim}, \ldots, [t_n]_{B^\sim}\}$.

Therefore,

$$\begin{array}{ccc} [t \downarrow_{E/B^\sim}]_{B^\sim} & \leadsto_{E_1/B^\sim} & [u \downarrow_{E/B^\sim}]_{B^\sim} \\ = & & = \\ [C[t_1 \downarrow_{E/B^\sim}, \ldots, t_n \downarrow_{E/B^\sim}]]_{B^\sim} & \leadsto_{E_1/B^\sim} & [C'[t'_1 \downarrow_{E/B^\sim}, \ldots, t'_m \downarrow_{E/B^\sim}]]_{B^\sim} \end{array}$$

and the following schema is obtained:

$$\begin{array}{ccc} [t]_{B^\sim} & \overset{E_1/B^\sim}{\rightsquigarrow} & [u]_{B^\sim} \\ {\scriptstyle E/B^\sim}\downarrow{\scriptstyle !} & & \downarrow{\scriptstyle !}{\scriptstyle E/B^\sim} \\ [t \downarrow_{E/B^\sim}]_{B^\sim} & \overset{E_1/B^\sim}{\rightsquigarrow} & [u \downarrow_{E/B^\sim}]_{B^\sim} \end{array}$$

This exhausts all the possible cases. □

Proposition 3. *The binary relation $\succ_{E^\sim/B^\sim}$ is well-founded.*

Proof. Suppose otherwise. Then, since $\rightarrow_{E/B^\sim}$ and $\Rrightarrow_{E_1/B^\sim}$ are well-founded, there must exist an infinite chain of steps of the form:

$$[u_1]_{B^\sim} \rightarrow^*_{E/B^\sim} [u'_1]_{B^\sim} \Rrightarrow_{E_1/B^\sim} [u_2]_{B^\sim} \rightarrow^*_{E/B^\sim} [u'_2]_{B^\sim} \Rrightarrow_{E_1/B^\sim} \cdots$$

Note that, by Lemma 6, if $[u'_i]_{B^\sim} \Rrightarrow_{E_1/B^\sim} [u_{i+1}]_{B^\sim}$, then $[u'_i \downarrow_{E/B^\sim}]_{B^\sim} \Rrightarrow_{E_1/B^\sim} [u'_{i+1} \downarrow_{E/B^\sim}]_{B^\sim}$. By $\rightarrow_{E/B^\sim}$ being confluent and operationally terminating, there is then an infinite chain

$$[u_1 \downarrow_{E/B^\sim}]_{B^\sim} \Rrightarrow_{E_1/B^\sim} [u_2 \downarrow_{E/B^\sim}]_{B^\sim} \Rrightarrow_{E_1/B^\sim} \cdots$$

contradicting the well-foundedness of $\Rrightarrow_{E_1/B^\sim}$. □

Theorem 7. *If $\mathcal{E}$ is sort-decreasing (resp., ground sort-decreasing), confluent (resp., ground confluent), and operationally terminating (resp., ground operationally terminating) in a Σ-extensible way, then $\mathcal{E}^\sim$ is operationally terminating (resp., ground operationally terminating).*

Proof. Let $WfPT_{\mathcal{E}^\sim}$ be the collection of well-formed proof trees in $\mathcal{E}^\sim$ and let the mapping $lhead : WfPT_{\mathcal{E}^\sim} \longrightarrow T_{\Sigma^\sim/B^\sim}(X)$ denote the equivalence class $[t]_{B^\sim}$ of the term t occurring in the left-hand side of the goal of the given well-formed proof tree in $\mathcal{E}^\sim$. Let $\mathcal{T}_\infty \subseteq WfPT_{\mathcal{E}^\sim}$ be the collection of *infinite* well-formed proof trees in $\mathcal{E}^\sim$. By definition, the equational theory $\mathcal{E}^\sim$ is operationally terminating iff $\mathcal{T}_\infty = \varnothing$. We assume $\mathcal{T}_\infty \neq \varnothing$ and reason by contradiction. First of all, note that since E is operationally terminating modulo $B^\sim$, if $T \in \mathcal{T}_\infty$ we must have $lhead(T) \in T_{\Sigma^\sim/B^\sim}(X)_{Bool}$.

Let $L_\infty = lhead\,[\mathcal{T}_\infty]$. Obviously $L_\infty \neq \varnothing$. Also, let $L^\succ_\infty$ be the subset of L_∞ defined by:

$$L^\succ_\infty = \left\{ [t]_{B^\sim} \in L_\infty \mid [t]_{B^\sim} \text{ is } \left(\succ^+_{E^\sim/B^\sim}\right)\text{-minimal in } L_\infty \right\}.$$

First note that, by definition, we have $L^{\succ}_{\infty} \subseteq L_{\infty}$. Also note that since $\succ_{E^{\sim}/B^{\sim}}$ is well-founded, then $L_{\infty} \neq \varnothing$ implies $L^{\succ}_{\infty} \neq \varnothing$. Hence, we can choose $T \in \mathcal{T}_{\infty}$ such that $lhead(T) \in L^{\succ}_{\infty}$.

The goal of T must be either of the form $t \to u$ or $t \to^* u$, with $t, u \in T_{\Sigma^{\sim}}(X)$. Let us first show that the goal of T cannot be of the form $t \to u$:

1. If $t = f(t_1, \ldots, t_i, \ldots, t_n)$, $u = f(t_1, \ldots, u_i, \ldots, t_n)$, $n \geq 1$, and T has the form

$$(Cong)\ \frac{T_1}{f(t_1, \ldots, t_i, \ldots, t_n) \to f(t_1, \ldots, u_i, \ldots, t_n)}$$

   where the goal of T_1 is $t_i \to u_i$, then it must be the case that $T_1 \in \mathcal{T}_{\infty}$. Therefore, we must have t_i of sort *Bool*. But then $[t]_{B^{\sim}}$ is not $\left(\succ^{+}_{E^{\sim}/B^{\sim}}\right)$-minimal in L_{∞}, because $f(t_1, \ldots, t_i, \ldots, t_n) \rhd t_i$, and $\rhd_{B^{\sim}} \subseteq \succ^{+}_{E^{\sim}/B^{\sim}}$, i.e., $lhead(T) \notin L^{\succ}_{\infty}$, a contradiction.
2. Otherwise, T must have the form

$$(Repl)\ \frac{T_1}{t \to v\sigma}$$

   for some $\sigma : X \longrightarrow T_{\Sigma^{\sim}}(X)$, where the equation applied is of type (34), for some AC symbol $f \in \Omega$, namely

$$\begin{aligned} f\left(g\left(x^1_{s'_1}, \ldots, x^m_{s'_m}\right), z^2_s\right) \sim f\left(y^1_{s'''}, y^2_{s'''}\right) = \bot \\ \textbf{if } in^k_f\left(g\left(x^1_{s'_1}, \ldots, x^m_{s'_m}\right), y^1_{s'''}\right) \\ \sqcup\, in^k_f\left(g\left(x^1_{s'_1}, \ldots, x^m_{s'_m}\right), y^2_{s'''}\right) = \bot \end{aligned}$$

   which is the only type of conditional equations in $E_1 = E^{\sim} \setminus E$. Therefore,

$$t =_{B^{\sim}} \left(f\left(t^1_1, t^2_1\right) \sim f\left(t^1_2, t^2_2\right)\right)\sigma$$

   with f in Σ, and $lhead(T_1) = [u_1\sigma]_{B^{\sim}}$ with

$$u_1 = in^k_f\left(g\left(x^1_{s'_1}, \ldots, x^m_{s'_m}\right), y^1_{s'''}\right) \sqcup in^k_f\left(g\left(x^1_{s'_1}, \ldots, x^m_{s'_m}\right), y^2_{s'''}\right).$$

   Note that then $[t]_{B^{\sim}} \rightsquigarrow_{E_1/B^{\sim}} [u_1\sigma]_{B^{\sim}}$. But since $\rightsquigarrow_{E_1/B^{\sim}} \subseteq \succ_{E^{\sim}/B^{\sim}}$, then $[t]_{B^{\sim}} \notin L_{\infty}$, a contradiction.

We now show that the goal of T cannot be of the form $t \to^* u$. The well-formed proof tree T cannot be an application of rule (*Refl*), since then T would be finite. Therefore, it must be of the form

$$(Tran)\ \frac{T_1 \qquad T_2}{t \to^* u}$$

where the goal of T_1 has the form $t \to v$ and the goal of T_2 has the form $v \to^* u$. But

then, we have already shown that T_1 must be finite (we have shown that the minimal infinite proof tree, if it exists, does not have a $\rightarrow$-goal and, furthermore, the goal of T and T_1 have the same left-hand term) and (by the definition of well-founded proof trees) closed. Therefore, we must have $T_2 \in \mathcal{T}_\infty$ and $[t]_{B^\sim} \rightarrow_{E^\sim/B^\sim} [v]_{B^\sim}$. But then $[t]_{B^\sim}$ is not $\left(\succ^+_{E^\sim/B^\sim}\right)$-minimal in L_∞ because $\rightarrow_{E^\sim/B^\sim} \subseteq \succ_{E^\sim/B^\sim}$, a contradiction.

This exhausts all the possible cases showing that $\mathcal{T}_\infty = \varnothing$. Thus, $\mathcal{E}^\sim$ is operationally terminating. □

5.3. Preservation of (Ground) Confluence

Since $\mathcal{E}^\sim$ is sort-decreasing (resp.,ground sort-decreasing) by Theorem 2 and operationally terminating (resp.,ground operationally terminating) by Theorem 7, the confluence (resp.,ground confluence) of $\mathcal{E}^\sim$ follows from its local confluence (resp.,ground local confluence) [19].

Local confluence (resp.,ground local confluence) can be established via joinability (resp.,ground joinability) of the so-called conditional critical pairs. Throughout this section the case where $\mathcal{E}$ itself is not only ground confluent but also confluent is also considered, which is indicated by enclosing the term "ground" in parentheses to cover both confluence and the weaker ground confluence case.

Definition 8 (Conditional Critical Pair)**.** *Given* $(\Sigma, E \cup B)$ *with* Σ *preregular, B sort-preserving and with R (rules corresponding to E as oriented equations) B-coherent, and given oriented conditional equations* $l \rightarrow r$ **if** $C, l' \rightarrow r'$ **if** $C' \in R$ *such that* $(\mathcal{V}ar(l) \cup \mathcal{V}ar(r) \cup \mathcal{V}ar(C)) \cap (\mathcal{V}ar(l') \cup \mathcal{V}ar(r') \cup \mathcal{V}ar(C')) = \varnothing$ *and* $l|_p\sigma =_B l'\sigma$*, for some nonvariable position* $p \in \mathcal{P}os(l)$ *and B-unifier* σ *of* $l|_p$ *and* l'*, then the triple*

$$C\sigma \wedge C'\sigma \Rightarrow l\sigma[r'\sigma]_p = r\sigma$$

is called a (conditional) critical pair.

The proof of Theorem 8 below is obtained by case analysis. It considers the conditional critical pairs of E that are joinable by assumption, the critical pairs of the rules E_{BBool} in the theory $\mathcal{E}^{BBool}$ in Example 1, which are also joinable modulo the axioms B_{BBool}, and the conditional critical pairs of $E^\sim \setminus (E \cup E^{Bool})$. Note that, if $B^\sim$ contains associative axioms, $B^\sim$-unification is infinitary in general. Hence, we need to reason about the possible form of any $B^\sim$-unifier that can involve a critical pair between two oriented equations involving A-symbols to conclude the local confluence of $\mathcal{E}^\sim$. Note that the statement and proof of the theorem assumes that the theory $\mathcal{E}^{BBool}$ in Example 1 is used as the specification of the Booleans. We show below how this assumption can be relaxed to allow many other specifications of the Booleans, including decision procedures.

Due to the large number of cases that need to be considered, we state the theorem below and give a detailed proof of it in the companion appendix (Appendix C), electronically available with this paper.

Theorem 8. *Let* $\mathcal{E}$ *be sort-decreasing (resp., ground sort-decreasing), B-coherent, operationally terminating (resp., ground operationally terminating) in a* Σ*-extensible way, and confluent (resp., ground confluent), and with sub-signature* Ω *of free constructors*

modulo B, and let $\mathcal{E}^{\sim}$ be obtained from $\mathcal{E}$ by importing the theory $\mathcal{E}^{BBool}$ as the Boolean theory. Then $\mathcal{E}^{\sim}$ is confluent (resp., ground confluent).

We now show how the dependence of Theorem 8 on the choice of the Boolean theory $\mathcal{E}^{BBool}$ can be greatly relaxed, so that many other specifications of the Booleans, including decision procedures, can be used instead. The key results allowing this extension are the following, much more general lemma and corollary:

Lemma 9. *Let $(\Sigma, E \cup B)$ be a B-preregular ordered-sorted theory, with the (possibly conditional) equations E ground sort-decreasing and ground confluent modulo B when oriented from left to right as rewrite rules; and let $E' \cup B'$ be unconditional Σ-equations such that $\mathcal{T}_{\Sigma/E\cup B} \models E' \cup B'$. Then the theory $(\Sigma, E \cup E' \cup B \cup B')$ with the equations $E \cup E'$ oriented as rewrite rules is ground confluent modulo $B \cup B'$, and $\mathcal{T}_{\Sigma/E\cup B} \cong \mathcal{T}_{\Sigma/E\cup E'\cup B\cup B'}$.*

Proof. The isomorphism $\mathcal{T}_{\Sigma/E\cup B} \cong \mathcal{T}_{\Sigma/E\cup E'\cup B\cup B'}$ follows directly from Lemma 1. Therefore, for any ground Σ-terms t, t' we have $t =_{E\cup E'\cup B\cup B'} t'$ iff $t =_{E\cup B} t'$. To prove that $E \cup E'$ is ground confluent modulo $B \cup B'$, let t, u, v be ground terms such that $t \rightarrow^*_{E\cup E'/B\cup B'} u$ and $t \rightarrow^*_{E\cup E'/B\cup B'} v$. Then we have $u =_{E\cup E'\cup B\cup B'} v$ and therefore $u =_{E\cup B} v$, so that, by the assumptions on $(\Sigma, E \cup B)$, there is a ground term w such that $u \rightarrow^*_{E/B} w$ and $v \rightarrow^*_{E/B} w$, which, since $\rightarrow_{E/B} \subseteq \rightarrow_{E\cup E'/B\cup B'}$, proves the ground confluence of $E \cup E'$ modulo $B \cup B'$, as desired. □

Corollary 5. *Let $(\Sigma, E \cup B)$ and $E' \cup B'$ be as in Lemma 9 above, with Σ $(B \cup B')$-preregular, and $E \cup E'$ sort-decreasing, and operationally terminating modulo $B \cup B'$. Then, for any ground Σ-term t, we have $t{\downarrow_{E/B}} =_B t{\downarrow_{E\cup E'/B\cup B'}}$.*

Proof. Since $\rightarrow_{E/B} \subseteq \rightarrow_{E\cup E'/B\cup B'}$, we must have $(t{\downarrow_{E\cup E'/B\cup B'}}){\downarrow_{E/B}} = t{\downarrow_{E\cup E'/B\cup B'}}$. And by $\mathcal{T}_{\Sigma/E\cup B} \cong \mathcal{T}_{\Sigma/E\cup E'\cup B\cup B'}$, $t{\downarrow_{E\cup E'/B\cup B'}} =_{E\cup B} t$, which by the assumptions on $(\Sigma, E \cup B)$ gives us $t{\downarrow_{E/B}} =_B t{\downarrow_{E\cup E'/B\cup B'}}$, as desired. □

We can now apply the above lemma to show how the dependence of Theorem 8 on the choice of the Boolean theory $\mathcal{E}^{BBool}$ can be greatly relaxed.

Theorem 10. *Let $\mathcal{E}$ be ground sort-decreasing, B-coherent, ground operationally terminating in a Σ-extensible way, and ground confluent, and with sub-signature Ω of free constructors modulo B; and let $(\Sigma^{Bool}, E^{Bool} \cup B^{Bool})$ be a specification of the Booleans with B^{Bool} a collection of associativity and commutativity axioms, and such that the Boolean theory $(\Sigma^{Bool}, E^{Bool} \cup E_{BBool} \cup B^{Bool} \cup B_{BBool})$ obtained by combining $(\Sigma^{Bool}, E^{Bool}, B^{Bool})$ and $\mathcal{E}^{BBool}$ is terminating with one of the AC-RPO orders described in Lemma 2. Let $\mathcal{E}^{\sim}$ be obtained from $\mathcal{E}$ by importing the theory $(\Sigma^{Bool}, E^{Bool} \cup E_{BBool} \cup B^{Bool} \cup B_{BBool})$ as the Boolean theory. Then $\mathcal{E}^{\sim}$ is ground operationally terminating and ground confluent.*

Proof. First of all, by Theorem 7, $\mathcal{E}^{\sim}$ is ground operationally terminating. Furthermore, $\mathcal{E}^{\sim}$ extends the simpler theory $\mathcal{E}^{\sim\prime}$ obtained as an equality enrichment of $\mathcal{E}$ by importing $\mathcal{E}^{BBool}$ as the Boolean theory by just adding the inductive theorems $E^{Bool} \cup B^{Bool}$. Therefore, by Theorems 7 and 8, $\mathcal{E}^{\sim\prime}$, is ground operationally terminating and ground confluent, and then by Lemma 9 $\mathcal{E}^{\sim}$ is ground confluent. □

Example 6. *As a concrete example showing how the above theorem can be applied in particular to a rewriting-based Boolean decision procedure, we can use as our theory* $(\Sigma^{Bool}, E^{Bool} \cup E_{BBool} \cup B^{Bool} \cup B_{BBool})$ *the extension of the theory* $\mathcal{E}^{BBool}$ *obtained by combining it with the Dijkstra-Schoelten rewriting-based decision procedure described in [14]. This combination can be achieved by explicitly adding a binary Boolean equivalence operator* $\equiv$*, together with associativity and commutativity axioms for* $\equiv$*, plus the following equations oriented from left to right as rewrite rules:*

$$\begin{aligned} x \sqcup x &= x, & x \sqcup (y \equiv z) &= (x \sqcup y) \equiv (x \sqcup z), \\ x \equiv x &= \top, & x \sqcap y &= (x \equiv y) \equiv (x \sqcup y), \\ x \equiv \top &= x, & \lrcorner x &= x \equiv \bot. \end{aligned}$$

Furthermore, the theory thus obtained is confluent and terminating modulo the associativity and commutativity axioms for $\sqcup$, $\sqcap$ *and* $\equiv$*, and provides a rewriting-based decision procedure for Boolean logic which is just a definitional extension (plus extra rules that are logical consequences) of the Dijkstra-Schoelten decision procedure in [14].*

5.4. Preservation of Free Constructors Modulo

Theorem 12 below, together with the fact that for any Σ-terms t, t', $t \to_{E/B} t'$ iff $t \to_{E^\sim/B^\sim} t'$, proves that $\Omega \cup \{\top, \bot\}$ is a signature of *constructors* for $(\Sigma^\sim, E^\sim)$, in the sense that each ground Σ-term reduces to an Ω-term, and each ground Boolean term to either $\top$ or $\bot$. Then, using the ground confluence of $(\Sigma^\sim, E^\sim)$ modulo $B^\sim$, we prove that these constructors are *free* modulo $B^\sim$ in Theorem 13.

Lemma 11. *Let* Ω *be a signature of free constructors modulo* B *for* $\mathcal{E}$ *and let terms* $f(t_1, \ldots, t_n)$[7]$, t \in T_\Omega$ *where* $f : s\ s \to s \in \Omega$ *is AC and root*(t_i) *and root*(t) *are not subsort-overloaded with* $f : s\ s \to s$*. If* $\mathcal{E}$ *is ground sort-decreasing, ground confluent, ground operationally terminating modulo* B*, and if all* $t, f(t_1, \ldots, t_n), t_1, \ldots, t_n$ *are in the same connected component, then:*

1. *if* $\exists t_i$, $1 \le i \le n$, $in_f^k(t, t_i) \to^+_{E^\sim/B^\sim} \top$*, then* $in_f^k(t, f(t_1, \cdots, t_n)) \to^+_{E^\sim/B^\sim} \top$,
2. *if* $\forall t_i$, $1 \le i \le n$, $in_f^k(t, t_i) \to^+_{E^\sim/B^\sim} \bot$ *then* $in_f^k(t, f(t_1, \cdots, t_n)) \to^+_{E^\sim/B^\sim} \bot$.

Proof. To prove (1), let $in_f^k(t, t_i) \to^+_{E^\sim/B^\sim} \top$. Since there is an Ω-term u such that $f(t_1, \ldots, t_n) =_B f(t_i, u)$, there is a one-step rewrite $in_f^k(t, f(t_1, \ldots, t_n)) \to_{E^\sim/B^\sim} in_f^k(t, t_i) \sqcup in_f^k(t, u)$ and therefore a rewrite sequence of the form $in_f^k(t, f(t_1, \ldots, t_n)) \to^+_{E^\sim/B^\sim} \top \sqcup in_f^k(t, u) \to^+_{E^\sim/B^\sim} \top$, as desired.

To prove (2), we induct on n to get a rewrite sequence $in_f^k(t, f(t_1, \ldots, t_n)) \to^+_{E^\sim/B^\sim} in_f^k(t, t_1) \sqcup \cdots \sqcup in_f^k(t, t_n)$; therefore, a rewrite sequence $in_f^k(t, f(t_1, \ldots, t_n)) \to^+_{E^\sim/B^\sim} \bot \sqcup \cdots \sqcup \bot \to^+_{E^\sim/B^\sim} \bot$ using the hypothesis, as desired. □

[7] any term t such that $t =_B f(t_1, f(t_2, \ldots, f(t_{n-1}, t_n) \cdots))$ is abbreviated by $f(t_1, \ldots, t_n)$.

Theorem 12. *If $\mathcal{E}$ is ground sort-decreasing, ground confluent, and ground operationally terminating modulo B with a signature of free constructors Ω modulo B, then for all $t, t' \in T_\Omega$ in the same connected component:*

1. *if $t =_B t'$ then $t \sim t' \to^+_{E^\sim/B^\sim} \top$,*
2. *if $t \neq_B t'$ then $t \sim t' \to^+_{E^\sim/B^\sim} \bot$.*

Proof. Suppose the theorem fails for some pair $t, t' \in T_\Omega$ of terms in the same connected component. It *cannot* fail for $t =_B t'$, because applying the equation (17) oriented as a rule, $t \sim t' \to^+_{E^\sim/B^\sim} \top$ is obtained. Therefore, it must fail for some pair t, t' such that $t \neq_B t'$, but $t \sim t' \not\to^+_{E^\sim/B^\sim} \bot$. Let us choose a pair with $|t| + |t'|$ *smallest possible*, where $|t|$ is the size of t as a tree. Note that $root(t)$, $root(t')$ must be subsort-overloaded, since otherwise $t = f(t_1, \ldots, t_n)$, $t' = g(t'_1, \ldots, t'_m)$ and a rewrite $f(t_1, \ldots, t_n) \sim g(t'_1, \ldots, t'_m) \to_{E^\sim/B^\sim} \bot$ is possible. Likewise, t and t' cannot have disjoint sorts, since then $t \sim t'$ could again be rewritten to $\bot$. Reasoning by cases on $f = root(t) = root(t')$:

1. if f is an absolutely free function symbol, then $t = f(t_1, \ldots, t_n)$, $t' = f(t'_1, \ldots, t'_n)$. This means that there is a $i \in \{1, \ldots, n\}$ such that $t_i \neq_B t'_i$. And since $|t_i| + |t'_i| < |t| + |t'|$ we must have $t_i \sim t'_i \to^+_{E^\sim/B^\sim} \bot$. But then this gives us:

$$t \sim t' \to_{E^\sim/B^\sim} \prod_{j=1}^{n} t_j \sim t'_j \to^+_{E^\sim/B^\sim} (\bot \sqcap \prod_{\substack{i \neq j \\ j=1}}^{n} t_i \sim t'_i) \to_{E^\sim/B^\sim} \bot$$

   contradicting $t \sim t' \not\to^+_{E^\sim/B^\sim} \bot$;
2. if f is a C symbol we have $t = f(t_1, t_2)$, $t' = f(t'_1, t'_2)$, and the reduction $t \sim t' \to_{E^\sim/B^\sim} (t_1 \sim t'_1 \sqcap t_2 \sim t'_2) \sqcup (t_1 \sim t'_2 \sqcap t_2 \sim t'_1)$. We will reach a contradiction if we show that both conjunctions reduce to $\bot$. Consider the left conjunction. We must have $t_1 \neq_B t'_1$ or $t_2 \neq_B t'_2$, since otherwise $t =_B t'$. Say $t_1 \neq_B t'_1$ (the other side is similar). Since $|t_1| + |t'_1| < |t| + |t'|$, we must have $t_1 \sim t'_1 \to^+_{E^\sim/B^\sim} \bot$. So that the left conjunction reduces to $\bot$. Reasoning in the same way on the right conjunction, we have that $t_1 \sim t'_2 \to^+_{E^\sim/B^\sim} \bot$ ($t_1 \neq_B t'_2$) or $t_2 \sim t'_1 \to^+_{E^\sim/B^\sim} \bot$ ($t_2 \neq_B t'_1$). This gives us:

$$t \sim t' \to^+_{E^\sim/B^\sim} \bot \sqcup \bot \to^+_{E^\sim/B^\sim} \bot$$

   contradicting $t \sim t' \not\to^+_{E^\sim/B^\sim} \bot$;
3. if f is an A symbol (for the sake of readability, we represent such as f by an infix operator '$\cdot$'), we must have $t =_B t_1 \cdot \ldots \cdot t_n \neq_B t'_1 \cdot \ldots \cdot t'_m =_B t'$, where we ignore parentheses and $root(t_i)$, $root(t'_j)$ are not subsort-overloaded with $\cdot$, $n, m \geq 2$, and without loss of generality we may assume $n \leq m$. We can distinguish two cases:
   (a) $t' = t_1 \cdot \ldots \cdot t_n \cdot t'_{n+1} \cdot \ldots \cdot t'_m$ and $n < m$, in which case we reach a contradiction since we can apply the equation (32) oriented as a rule to get the contradiction $t \sim t' \to^+_{E^\sim/B^\sim} \bot$, or
   (b) $n \leq m$ and there is an i, $1 \leq i \leq n$ such that $t_j =_B t'_j$ for $1 \leq j < i$, but $t_i \neq_B t'_i$. Therefore, by applying the equation (30) oriented as a rule we get

either $t \sim t' \to^*_{E^\sim/B^\sim} t_i \cdot \ldots \cdot t_n \sim t'_i \cdot \ldots t'_m$ for $i < n$ where by $|t_i| + |t'_i| < |t| + |t'|$ we must have $t_i \sim t'_i \to^+_{E^\sim/B^\sim} \bot$ and therefore we can apply either:

- the equation (26) oriented as a rule if $root(t_i)$, $root(t_j)$ are subsort-overloaded and the minimality hypothesis, contradicting $t \sim t' \not\to^+_{E^\sim/B^\sim} \bot$;
- or the equation (28) oriented as a rule if $root(t_i)$, $root(t_j)$ are not subsort-overloaded;

or $i = n$, and we get $t \sim t' \to^+_{E^\sim/B^\sim} t_n \sim t'_n$ if $m = n$, immediately yielding the contradiction $t \sim t' \to^+_{E^\sim/B^\sim} \bot$; or finally $t \sim t' \to^+_{E^\sim/B^\sim} t_n \sim t'_n \cdot \ldots \cdot t'_m$ if $m > n$ yielding again the contradiction $t \sim t' \to^+_{E^\sim/B^\sim} \bot$, because $root(t_n)$, $\cdot$ are not subsort-overloaded.

4. if f is an AC symbol (for the sake of readability, we represent such as f by an infix operator '+') we have again (ignoring parentheses) decompositions $t =_B t_1 + \cdots + t_n \neq_B t'_1 + \cdots + t'_m =_B t'$ where $root(t_i)$ and $root(t'_j)$ are not subsort-overloaded with $+$, $n, m \geq 2$, and without loss of generality we may assume $n \leq m$. We again have two main cases:
   (a) if $t' = t_1 + \cdots + t_n + t'_{j_1} + \cdots + t'_{j_k}$, $k \geq 1$ and $n < m$, we can apply the equation (40) oriented as a rule to get the contradiction $t \sim t' \to^+_{E^\sim/B^\sim} \bot$, or
   (b) if $n \leq m$ and there is a subset $I \subset \{1, \ldots, n\}$, and an injective function $\alpha : I \to \{1, \ldots, m\}$ such that $t_i =_B t'_{\alpha(i)}$, but for all $j \in \{1, \ldots, n\} \setminus I$, $l \in \{1, \ldots, m\} \setminus \alpha(I)$ we have $t_j \neq_B t'_l$. By $|t_j| + |t'_l| < |t| + |t'|$ we must have $t_j \sim t'_l \to^+_{E^\sim/B^\sim} \bot$. We then have two cases:
      i. if $\{1, \ldots, n\} \setminus I = \{j\}$, then, either (1) $m > n$, so that by applying the equation (35) oriented as a rule we get $t \sim t' \to^+_{E^\sim/B^\sim} t_j \sim t_{l_1} + \cdots + t_{l_k}$ where $\{l_1, \ldots, l_k\} = \{1, \ldots, m\} \setminus \alpha(I)$, for $k \geq 2$, yielding immediately a contradiction $t \sim t' \to^+_{E^\sim/B^\sim} \bot$ since $root(t_j)$ is not subsort-overloaded with the AC symbol $+$; or (2) $n = m$, so that applying the same equation oriented as a rule we get $t \sim t' \to^+_{E^\sim/B^\sim} t_j \sim t_{l_1} \to^+_{E^\sim/B^\sim} \bot$, again a contradiction,
      ii. $|\{1, \ldots, n\} \setminus I| \geq 2$, so we get for $j \in \{j_1, \ldots, j_{k_1}\} = \{1, \ldots, n\} \setminus I$, $l \in \{l_1, \ldots, l_{k_2}\} = \{1, \ldots, m\} \setminus \alpha(I)$ that $t_j \neq t'_l$, and a reduction $t \sim t' \to t_{j_1} + \cdots + t_{j_{k_1}} \sim t'_{l_1} + \cdots + t'_{l_{k_2}}$. But then, by applying the equations for in^k_+ oriented as rules we have $in^k_f(t_{j_1}, t'_{l_1} + \cdots t'_{l_{k_2}}) \to^+_{E^\sim/B^\sim} \bot$, so that we can apply to $t_{j_1} + \cdots + t_{j_{k_1}} \sim t'_{l_1} + \cdots + t'_{l_{k_2}}$ the equation (34) oriented as a rule and by Lemma 11 and the minimality hypothesis, a contradiction is again reached.

□

We have already shown that $\Omega^\sim = \Omega \cup \{\top, \bot\} \subseteq \Sigma^\sim$ is a signature of constructors for $\mathcal{E}^\sim$. We now show that, thanks to the preservation of confluence, these constructors are free modulo $B^\sim$. To begin with, from Theorem 12 we obtain the following corollary.

Corollary 6. *If $\mathcal{E}$ is ground sort-decreasing, B-coherent, ground confluent, and operationally terminating modulo B, with Ω a signature of free constructors modulo B, then for all $t, t' \in T_\Omega$ in the same connected component:*

1. $t =_B t'$ *iff* $t \sim t' \to^+_{E^\sim/B^\sim} \top$.
2. $t \neq_B t'$ *iff* $t \sim t' \to^+_{E^\sim/B^\sim} \bot$.

Proof. Both ($\Rightarrow$) implications follow from Theorem 12. Suppose that $t \sim t' \to^+_{E^\sim/B^\sim} \top$ and $t \neq_B t'$. Then, by Theorem 12, we also have $t \sim t' \to^+_{E^\sim/B^\sim} \bot$, which is impossible, since $\top$ and $\bot$ are in canonical form and the equations $E^\sim$ are ground confluent modulo B. Likewise, if $t \sim t' \to^+_{E^\sim/B^\sim} \bot$ and $t =_B t'$ we also get $t \sim t' \to^+_{E^\sim/B^\sim} \bot$, contradicting confluence. □

We have identified $\Omega^\sim = \Omega \cup \{\top, \bot\} \subseteq \Sigma^\sim$ as a signature of constructors for $\mathcal{E}^\sim$. We now prove that the constructors in $\Omega^\sim$ are free modulo $B^\sim$.

Theorem 13. *If $\mathcal{E}$ is ground sort-decreasing, B-coherent, ground confluent, and operationally terminating modulo B in a Σ-extensible way, and Ω is the signature of free constructors modulo B of $\mathcal{E}$, then $\mathcal{E}^\sim$ has $\Omega^\sim = \Omega \cup \{\top, \bot\}$ as a signature of free constructors modulo $B^\sim$.*

Proof. For each $t \in T_{\Sigma^\sim}$, either $t \in T_\Sigma$, in which case, since $t \to_{E/B} t'$ iff $t \to_{E^\sim/B^\sim} t'$, the result follows trivially, or $t \in T_{\Sigma^\sim,Bool}$. Therefore, since $\mathcal{E}^\sim$ is ground confluent and terminating, and both $\bot$, and $\top$ are in canonical form, all we need to prove is that $t \downarrow_{E^\sim/B^\sim}$ is either $\bot$ or $\top$.

Indeed, by Lemma 11, and Theorem 12, $t \downarrow_{E^\sim/B^\sim}$ *cannot* contain any subterm of the form $u \sim v$ or $in^k_f(u, v)$. Therefore $t \downarrow_{E^\sim/B^\sim}$ *must* be a Boolean ground term, which, being in canonical form, must be either $\top$, or $\bot$, as desired. □

5.5. $\mathcal{E}^\sim$ is an Equality Enrichment

From the good properties of $\mathcal{E}^\sim$ we can now prove that this theory is indeed an equality enrichment of $\mathcal{E}$.

Theorem 14. *Let $\mathcal{E} = (\Sigma, E \cup B)$ be an order-sorted equational theory with signature $\Omega \subseteq \Sigma$ of free constructors modulo B and let $\mathcal{E}^\sim = (\Sigma^\sim, E^\sim \cup B^\sim)$ be the equational theory obtained by the transformation $\mathcal{E} \mapsto \mathcal{E}^\sim$. If $\mathcal{E}$ is ground sort-decreasing, B-coherent, ground operationally terminating in a Σ-extensible way, and ground confluent modulo B, then $\mathcal{E}^\sim$ is a Boolean equality enrichment of $\mathcal{E}$.*

Proof. Since for each $t, t' \in T_\Sigma$ we have $t \to_{E/B} t'$ iff $t \to_{E^\sim/B^\sim} t'$, the ground confluence and sort-decreasingness of E modulo B and $E^\sim$ modulo $B^\sim$ ensure that $E \cup B \vdash t = t'$ iff $E^\sim \cup B^\sim \vdash t = t'$, and therefore that $\mathcal{T}_{\mathcal{E}^\sim|_\Sigma} \cong \mathcal{T}_\mathcal{E}$, so that the extension is protecting. Also, since $\Omega^\sim$ is a signature of free constructors we know that $\mathcal{T}_{\mathcal{E}^\sim,Bool} = \{[\top], [\bot]\}$, with $[\top] \neq [\bot]$. We only need to show that the equivalences:

$$\begin{array}{lcl} \mathcal{E} \vdash t = u & \iff & \mathcal{E}^\sim \vdash (t \sim u) = \top, \\ \mathcal{E} \nvdash t = u & \iff & \mathcal{E}^\sim \vdash (t \sim u) = \bot, \end{array}$$

hold. But by our assumptions on $\mathcal{E}$, the result in Corollary 6, and ground confluence, sort-decreasingness, and ground operational termination of $\mathcal{E}^\sim$ we have:

$$\begin{array}{l} \mathcal{E} \vdash t = u \iff t \downarrow_{E/B =_B} u \downarrow_{E/B} \iff (t \sim u) \downarrow_{E^\sim/B^\sim} = \top \iff \mathcal{E}^\sim \vdash (t \sim u) = \top, \\ \mathcal{E} \nvdash t = u \iff t \downarrow_{E/B \neq_B} u \downarrow_{E/B} \iff (t \sim u) \downarrow_{E^\sim/B^\sim} = \bot \iff \mathcal{E}^\sim \vdash (t \sim u) = \bot. \end{array}$$

□

6. Automation and Applications of $\mathcal{E} \mapsto \mathcal{E}^{\sim}$

The transformation $\mathcal{E} \mapsto \mathcal{E}^{\sim}$ is obviously *constructive* and has been automated in Maude using its reflective features: it takes the meta-representation of $\mathcal{E}$ in Maude as input and constructs a meta-representation of $\mathcal{E}^{\sim}$ as output. The transformation itself has already been incorporated into Maude formal tools, including the latest version of the Maude Formal Environment [20], the Maude Church-Rosser and Coherence Checker [19] (CRC-ChC), and the Maude Invariant Analyzer tool [21].

6.1. A Case Study

We present a case study in which the transformation $\mathcal{E} \mapsto \mathcal{E}^{\sim}$ is used in the Maude Invariant Analyzer (InvA) tool [21]. The InvA tool mechanizes an inference system for deductively proving safety properties of rewrite theories: it transforms all formal temporal reasoning about safety properties of concurrent transitions to purely equational inductive reasoning. The InvA tool provides a substantial degree of mechanization and can automatically discharge many proof obligations without user intervention. In this section, we illustrate how equality enrichments can be used to support the deductive verification task in the InvA tool for a mutual exclusion property of processes in the QLOCK protocol.

The mutual exclusion protocol QLOCK uses a global queue as follows:

- each process that participates in the protocol does the following:
  - if the process wants to use the critical resource and its name is not in the global queue, it places its name in the queue;
  - if the process wants to use the critical resource and its name is in the global queue, if its name is at the top of the queue then the process gains access to the critical resource; otherwise it waits; and
  - if the process finishes the critical resource, it removes its name from the top of the global queue;
- the protocol should start from a state where the queue is empty; and
- it is assumed that each process can use the critical resource any number of times.

Consider the following equational theory $\mathcal{E}^{\texttt{QLOCK-STATE}}$, which represents the states of QLOCK with terms of sort *State*. It protects the equational theory $\mathcal{E}^{\texttt{MSET}}$ presented in Section 4. Processes and names of processes are modeled with natural numbers of sort *Nat* in Peano notation. A term *Pi* | *Pw* | *Pc* | *Q* of sort *State* describes the state in which *Pi* is the collection of processes whose name is not in the global queue (or *idle* processes), *Pw* is the collection of processes that are waiting to gain access to the critical resource (or *waiting* processes), *Pc* is the collection of processes that are using the critical resource (or *critical* processes), and *Q* is the global queue of the system. Sorts *MSet* and *Queue* are used to represent collections of processes and queues of processes' names, respectively.

```
fmod QLOCK-STATE is
  protecting MSET .
  sort Queue .
  op nil : -> Queue [ctor] .
  op _@_ : Nat Queue -> Queue [ctor] .
  op _;_ : Queue Queue -> Queue .
  eq nil ; Q:Queue = Q:Queue .
  eq (N:Nat @ Q1:Queue) ; Q2:Queue = N:Nat @ (Q1:Queue ; Q2:Queue) .
  sort State .
  op _|_|_|_ : MSet MSet MSet Queue -> State [ctor] .
endfm
```

The behavior of a concurrent system in rewriting logic is specified by rewrite rules that define how the individual transitions change the state of the system. The specification of all transitions of QLOCK is described by six rewrite rules in the rewrite theory $\mathcal{R}^{\texttt{QLOCK}}$ as follows.

```
mod QLOCK is
  protecting QLOCK-STATE .
  vars Pi Pw Pc : MSet . var Q : Queue . vars N N' N'' : Nat .
  rl [to-wait-1] : N | Pw | Pc | Q => empty | Pw N | Pc
                   | Q ; (N @ nil ) .
  rl [to-wait-2] : N Pi | Pw | Pc | Q => Pi | Pw N | Pc
                   | Q ; (N @ nil ) .
  rl [to-crit-1] : Pi | N | Pc | N @ Q => Pi | empty | Pc N | N @ Q .
  rl [to-crit-2] : Pi | Pw N | Pc | N @ Q => Pi | Pw | Pc N | N @ Q .
  rl [to-idle-1] : Pi | Pw | N | N' @ Q => Pi N | Pw | empty | Q .
  rl [to-idle-2] : Pi | Pw | Pc N | N' @ Q => Pi N | Pw | Pc | Q .
endm
```

Rewrite rules to-idle-1 and to-idle-2 specify the behavior of a process that finishes using the critical resource: it goes to state idle and the name on top of the global queue is removed. Similarly, rewrite rules to-wait-1 and to-wait-2, and to-crit-1 and to-crit-2, specify the behavior of a process that wants to use the critical resource and of a process that is granted access to the critical resource, respectively.

We want to verify that the QLOCK system satisfies the following safety properties. It is key that: (i) it satisfies the mutual exclusion property, namely, that at any point of execution there is at most one process using the critical resource. We also want to verify that: (ii) the name on top of the global queue coincides with the name of the process using the critical resource, if any. Finally, we want to verify that: (iii) the global queue only contains the names of all waiting and critical processes. State predicates *mutex*, *priority*, and *cqueue*, respectively, specify properties (i), (ii), and (iii) in the following equational theory $\mathcal{E}^{\texttt{QLOCK-PREDS}}$. State predicate *init* specifies the set of initial states of QLOCK, with auxiliary function *set*? that characterizes multisets having no repeated elements. State predicate *unique* is a strengthening of *mutex* and *priority*. Auxiliary function *to-soup* on input Q of sort *Queue* computes the multiset of natural numbers appearing in Q.

```
fmod QLOCK-PREDS is
  protecting QLOCK-STATE .
  protecting EQ-MSET .
  vars N N'     : Nat .  var Q   : Queue .
  vars Pi Pw Pc : MSet . var NeS : NeMSet .
  ops init mutex unique priority cqueue : State -> [Bool] .
  eq init( Pi | empty | empty | nil ) = set?(Pi) .
  eq mutex( Pi | Pw | empty | Q ) = true .
  eq mutex( Pi | Pw | N | Q ) = true .
  eq mutex( Pi | Pw | N NeS | Q ) = false .
  eq unique( Pi | Pw | empty | Q ) = set?(Pi Pw) .
  eq unique( Pi | Pw | N | N @ Q ) = set?(Pi Pw N) .
  eq unique( Pi | Pw | N NeS | Q ) = false .
  eq priority( Pi | Pw | empty | Q ) = true .
  eq priority( Pi | Pw | N | N' @ Q ) = N ~ N' .
  eq priority( Pi | Pw | N Pc | N' @ Q )
   = (N ~ N') and (Pc ~ empty) .
  eq cqueue( Pi | Pw | Pc | Q ) = Pw Pc ~ to-soup(Q) .
....
endfm
```

Observe that $\mathcal{E}^{\text{QLOCK-PREDS}}$ protects the equality enrichment $\mathcal{E}^{\text{EQ-MSET}}$, in Section 4, for the connected component of sort *MSet* that defines the equality enrichment for sorts *Nat*, *MSet*, and *NeMSet*. The equality enrichments for these sorts are key in the specification of the state predicates. For instance, predicates *priority* and *cqueue* are directly defined in terms of the equality predicate for sorts *Nat* and *MSet*, and also use the Boolean connective for conjunction *and* that comes with the Boolean equality enrichment. Auxiliary function *set*? also makes use of the equality enrichment for sort *Nat*. Note that, in general, defining from scratch the equality enrichment for an AC-symbol such as the multiset union in $\mathcal{E}^{\text{MSET}}$, can be a daunting task. Instead, in $\mathcal{E}^{\text{QLOCK-PREDS}}$, the definition of the state predicate *cqueue* was straightforward with the help of the equality enrichment for multisets of natural numbers.

By using the InvA tool we are able to automatically prove that predicates *mutex* and *priority* are invariants of $\mathcal{R}^{\text{QLOCK}}$ for any initial state that satisfies predicate *init*. For predicate *cqueue* some proof obligations cannot be automatically discharged. In general terms, 22 out of 26 proof obligations were automatically discharged. However, this is an encouraging result, given that the current version of the InvA tool does not yet have dedicated inference support for Boolean equality enrichments, which could further improve the degree of automation.

7. Conclusion

This paper solves an important open problem: how to make the addition of equationally defined equality predicates effective and automatic for a very wide class of equational specifications with initial algebra semantics. That such a transformation should exists is suggested by the Bergstra-Tucker meta-theorem [2], but such a meta-result is not constructive and gives no insight as to how the transformation could be defined. The *equality enrichment transformation* has been defined for a very wide class

of algebraic specifications with highly expressive features such as order-sorted types, conditional equations, and rewriting modulo commonly occurring axioms. By means of (non-trivial) meta-theorems, it has been shown that all the expected good properties of the input theory $\mathcal{E}$ are inherited by $\mathcal{E}^{\sim}$ in the equality enrichment transformation $\mathcal{E} \mapsto \mathcal{E}^{\sim}$.

Using reflection, the transformation has been implemented in Maude and has already been integrated into the Maude Church-Rosser and Coherence Checker [19] (CRC-ChC), and the Maude Invariant Analyzer tool [21]. The case study in Section 6.1 shows how the addition of equationally-defined equality predicates also makes the specification and verification of safety properties in the InvA tool considerably easier.

In general, the contributions presented in this work open up many useful applications to improve the state of the art in formal verification of algebraic specifications. In particular, they have already provided a host of such applications within the Maude formal environment.

In the near future it should be added to other tools such as the Maude Termination Tool [18] (MTT) and the Maude Sufficient Completeness Checker [22] (SCC). One obvious advantage of these additions is the possibility of systematically transforming specifications making use of built-in equalities and inequalities, which cannot be handled by formal tools, into specifications where such built-in equalities and inequalities are systematically replaced by equationally-defined equalities, so that formal tools can be applied.

Adding an equationally-defined equality to Maude's Inductive Theorem Prover [23] (ITP) would make this tool more effective in many ways, and would also greatly reduce the complexities of dealing with arbitrary universal formulas as goals, since all such formulas could be reduced to unconditional equality goals. It would also be very useful to explore the use of the $\mathcal{E} \mapsto \mathcal{E}^{\sim}$ transformation in *inductionless induction* theorem proving. Yet another very useful field of application would be *early failure detection* in narrowing-based unification. The idea is that E/B-unification goals can be viewed as equality goals, which can be detected to have already *failed* if they can be rewritten to *false* with $E^{\sim}$ modulo $B^{\sim}$.

References

[1] J. A. Goguen, How to Prove Algebraic Inductive Hypotheses Without Induction, in: W. Bibel, R. Kowalski (Eds.), Proc. of the 5th Conference on Automated Deduction, CADE'80, Vol. 87 of LNCS, Springer-Verlag, 1980, pp. 356–373.

[2] J. Bergstra, J. Tucker, Characterization of Computable Data Types by Means of a Finite Equational Specification Method, in: J. W. de Bakker, J. van Leeuwen (Eds.), Proc. of the 7th International Colloquium on Automata, Languages and Programming, ICALP'80, Vol. 81 of LNCS, Springer-Verlag, 1980, pp. 76–90.

[3] F. Durán, S. Lucas, J. Meseguer, Termination Modulo Combinations of Equational Theories, in: S. Ghilardi, R. Sebastiani (Eds.), Proc. of the 7th International Conference on Frontiers of Combining Systems, FroCoS'09, Vol. 5749 of LNCS, Springer-Verlag, 2009, pp. 246–262.

[4] R. Gutiérrez, J. Meseguer, C. Rocha, Order-Sorted Equality Enrichments Modulo Axioms, in: F. Durán (Ed.), Proc. of the 9th International Workshop on Rewriting Logic and its Applications, WRLA'12, Vol. 7571 of LNCS, Springer-Verlag, 2012, pp. 162–181.

[5] D. R. Musser, On Proving Inductive Properties of Abstract Data Types, in: Proc. of the 7th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL'80, ACM Press, 1980, pp. 154–162.

[6] J. Meseguer, J. A. Goguen, Initiality, Induction and Computability, in: M. Nivat, J. C. Reynolds (Eds.), Algebraic Methods in Semantics, Cambridge University Press, 1986, pp. 459–541.

[7] M. Nakamura, K. Futatsugi, On Equality Predicates in Algebraic Specification Languages, in: C. B. Jones, Z. Liu, J. Woodcock (Eds.), Proc. of the 4th International Conference on Theoretical Aspects of Computing, ICTAC'07, Vol. 4711 of LNCS, Springer-Verlag, 2007, pp. 381–395.

[8] F. Baader, T. Nipkow, Term Rewriting and All That, Cambridge University Press, 1998.

[9] J. Goguen, J. Meseguer, Order-Sorted Algebra I: Equational Deduction for Multiple Inheritance, Overloading, Exceptions and Partial Operations, Theoretical Computer Science 105 (1992) 217–273.

[10] J. Meseguer, Membership Algebra as a Logical Framework for Equational Specification, in: F. Parisi-Presicce (Ed.), Recent Trends in Algebraic Development Techniques, Proc. of the 12th International Workshop on Workshop on Algebraic Development Techniques, WADT'97, Vol. 1376 of LNCS, Springer-Verlag, 1997, pp. 18–61.

[11] M. Clavel, F. Durán, S. Eker, P. Lincoln, N. Martí-Oliet, J. Meseguer, C. Talcott, All About Maude – A High-Performance Logical Framework, Vol. 4350 of LNCS, Springer-Verlag, 2007.

[12] F. Durán, J. Meseguer, On the Church-Rosser and coherence properties of conditional order-sorted rewrite theories, Journal of Logic and Algebraic Programming 81 (7-8) (2012) 816–850.

[13] S. Lucas, C. Marché, J. Meseguer, Operational Termination of Conditional Term Rewriting Systems, Information Processing Letters 95 (4) (2005) 446–453. doi:http://dx.doi.org/10.1016/j.ipl.2005.05.002.

[14] C. Rocha, J. Meseguer, Theorem proving modulo based on Boolean equational procedures, in: R. Berghammer, B. Möller, G. Struth (Eds.), RelMiCS, Vol. 4988 of LNCS, Springer, 2008, pp. 337–351.

[15] E. Ohlebusch, Advanced Topics in Term Rewriting, Springer-Verlag, 2002.

[16] L. Bachmair, D. A. Plaisted, Termination Orderings for Associative-Commutative Rewriting Systems, Journal of Symbolic Computation 1 (4) (1985) 329–349.

[17] A. Rubio, A Fully Syntactic AC-RPO, Information and Computation 178 (2) (2002) 515–533.

[18] F. Durán, S. Lucas, C. Marché, J. Meseguer, X. Urbain, Proving Operational Termination of Membership Equational Programs, Higher Order Symbolic Computation 21 (1-2) (2008) 59–88.

[19] F. Durán, J. Meseguer, On the Church-Rosser and Coherence Properties of Conditional Order-Sorted Rewrite Theories, Journal of Logic and Algebraic Programming 81 (7–8) (2012) 816–850.

[20] M. Clavel, F. Durán, J. Hendrix, S. Lucas, J. Meseguer, P. Ölveczky, The Maude Formal Tool Environment, in: T. Mossakowski, U. Montanari, M. Haveraaen (Eds.), Proc. of the 2nd Conference on Algebra and Coalgebra in Computer Science, CALCO'07, Vol. 4624 of LNCS, Springer-Verlag, 2007, pp. 173–178.

[21] C. Rocha, J. Meseguer, Proving safety properties of rewrite theories, in: A. Corradini, B. Klin, C. Cîrstea (Eds.), Proc. of 4th International Conference on Algebra and Coalgebra in Computer Science, CALCO'11, Vol. 6859 of LNCS, Springer-Verlag, 2011, pp. 314–328.

[22] J. Hendrix, M. Clavel, J. Meseguer, A Sufficient Completeness Reasoning Tool for Partial Specifications, in: J. Giesl (Ed.), Proc. of the 16th International Conference on Rewriting Techniques and Applications, RTA'05, Vol. 3467 of LNCS, Springer-Verlag, 2005, pp. 165–174.

[23] J. Hendrix, Decision Procedures for Equationally Based Reasoning, Ph.D. thesis, Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL, USA (2008).