Sentence-Normalized Conditional Narrowing Modulo in Rewriting Logic and Maude[⋆]

Luis Aguirre, Narciso Martí-Oliet, Miguel Palomino, and Isabel Pita

Facultad de Informática, Universidad Complutense de Madrid, Spain
`{luisagui, narciso, miguelpt, ipandreu}@ucm.es`

Abstract. This work studies the relationship between verifiable and computable answers for reachability problems in rewrite theories with an underlying membership equational logic. A new definition for R, A-rewriting that allows us to solve a bigger class of reachability problems, and a calculus that solves this class of problems always working with *canonical terms and normalized substitutions* has been developed. Given a reachability problem in a rewrite theory, this calculus can compute any normalized answer that can be checked by rewriting, or a more general one that can be instantiated to that answer.

Keywords: Maude, narrowing, reachability, rewriting logic, membership equational logic, unification

1 Introduction

Rewriting logic is a computational logic that has been around for more than twenty years [Mes90]. The semantics of rewriting logic [BM06] has a precise mathematical meaning, allowing mathematical reasoning for proving properties, providing a flexible framework for the specification of concurrent systems; moreover, it can express both concurrent computation and logical deduction, allowing its application in many areas such as automated deduction, software and hardware specification and verification, security, etc [MM02, Mes12].

A deductive system is specified in rewriting logic as a rewrite theory $\mathcal{R} = (\Sigma, \mathcal{E}, R)$, with $(\Sigma, \mathcal{E})$ an underlying equational theory (in this paper we will consider *membership equational logic*), where terms are given an algebraic data type, allowing us to identify as semantically equal two syntactically different terms, and R a set of rules that specify how the deductive system can derive one term from another. *Order-sorted*, *many-sorted*, and *unsorted* theories can be formulated as special cases of membership equational logic (Mel) theories.

Reachability problems in rewriting logic have the form

$$(\exists \bar{x}) t(\bar{x}) \to^* t'(\bar{x})$$

with t, t' terms with variables in $\bar{x}$, or a conjunction of several of these subgoals. Reachability problems can be solved by model checking methods for finite state spaces. When the initial term t has no variables, i.e., it is a ground term, and under certain admissibility conditions, rewriting can be used in a breadth-first way to traverse the state space, trying to find a suitable matching of $t'(\bar{x})$ in each traversed node. In the general case where $t(\bar{x})$ is not a ground term, a technique known as *narrowing* [Fay78] that was first proposed as a method for solving equational goals (*unification*), has been extended to cover also reachability goals [MT07], leaving equational goals as a special case. The strength of narrowing can be found in that it enables us to manage

[⋆] Partially supported by MINECO Spanish project StrongSoft (TIN2012–39391–C04–04) and Comunidad de Madrid program N-GREENS Software (S2013/ICE-2731).

complex concurrent and deductive systems that cannot be handled by faster, but more limited, specialized methods. Under the admissibility conditions for rewrite theories, which allow for conditional rules and equations with extra variables in the conditions under some requirements, and the assumption of the existence of an $\mathcal{E}$-unification algorithm, we can use narrowing modulo $\mathcal{E}$ to perform symbolic analysis of the possibly infinite set of initial states $t(\bar{x})$ in the state space and determine the actual values of $\bar{x}$ that allow us to derive $t'(\bar{x})$ from $t(\bar{x})$.

Such $\mathcal{E}$-unification algorithm can itself make use of narrowing at another level for finding the solution to its equational goals. Specific $\mathcal{E}$-unification algorithms exist for a small number of equational theories, but if the equational theory $(\Sigma, \mathcal{E})$ can be decomposed as $E \cup A$, where A is a set of axioms having a unification algorithm, and the equations E can be turned into a set of rules $\overrightarrow{E}$, by orienting them, such that the rewrite theory $\overrightarrow{\mathcal{E}} = (\Sigma, A, \overrightarrow{E})$ is admissible in the above sense, then narrowing can be used on $\overrightarrow{\mathcal{E}}$ to solve the $\mathcal{E}$-unification goals generated by performing narrowing on $\mathcal{R}$. For these equational goals the idea of *variants of a term* has been applied in recent years to narrowing. A strategy known as *folding variant narrowing* [ESM12], which computes a complete set of variants of any term, has been developed by Escobar, Sasse, and Meseguer, allowing unification modulo a set of unconditional equations and axioms. The strategy terminates on any input term on those systems enjoying the *finite variant property*, and it is optimally terminating. It is being used for cryptographic protocol analysis [MT07], with tools like Maude-NPA [EMM09], termination algorithms modulo axioms [DLM$^+$08a], algorithms for checking confluence and coherence of rewrite theories modulo axioms [DM12a], and infinite-state model checking [BM14]. Recent development in conditional narrowing have been made for order-sorted equational theories [CEM14] and also for narrowing with constraint solvers [RMM14].

Conditional narrowing without axioms for rewrite theories with an order-sorted type structure has been thoroughly studied for increasingly complex categories of term rewriting systems. A wide survey can be found in [MH94]. The literature is scarce when we allow for extra variables in conditions (e.g., [GM86] [Ham00]), conditional narrowing modulo axioms (e.g., [CEM14]), or conditional narrowing modulo a set of equations(e.g., [Boc93]). Conditional narrowing modulo axioms for Mel theories has not been addressed, to the best of our knowledge, one of the main reasons being the lack of fast and effective unification algorithms modulo axioms for Mel theories. Nonetheless, there are plenty of algebraic data types, including all types that imply some kind of order between subterms, that are better expressed inside a Mel theory, so there is a need to give an answer to these cases. In this paper we focus on conditional narrowing modulo axioms for rewrite theories with an underlying equational Mel theory.

The natural tool to work with when dealing with Mel theories is Maude, a high-level language and high-performance system supporting both equational and rewriting computation [CDE$^+$02], whose underlying equational logic is membership equational logic. Maude makes a systematic and efficient use of reflection through the META-LEVEL module. The META-LEVEL module has several built-in functions (`upModule, metaReduce, metaUnify, glbSorts, leastSort, ...`) that are used in the implementation of our calculi.

In previous work [AMPP14] we developed a narrowing calculus, implemented using Maude's reflection capabilities, that we felt could be enhanced if we could prove that it was sound and complete to use the normal forms of the remaining goals in the computation before each narrowing step with an oriented equation or a rule. From standard definitions for Mel deduction and rewriting modulo, we have developed in this work new concepts as intermediate tools for our improved narrowing calculus, so we had to prove that these intermediate tools were valid as replacement for the original ones. The concept of "sentence normalized rewriting" is the link between rewriting and our new narrowing calculus with normalization, so we had to prove that it was safe to use it. In fact this normalization can be considered as a "strategy" for the narrowing calculi for unification and for reachability. This strategy is independent of the chosen equations and positions were reduction is applied before each reduction step. Moreover, as our implementations work using Maude's metalevel, we get normalization for free before each narrowing step just by invoking the built-in `metaReduce` function with the metalevel versions of the theory (which is generated only once at the beginning of the computation) and of the current unification or reachability goal (which is always available), so we cut off the search space and speed up the computation at the same time. We already had these Maude's metalevel capabilities in mind when we decided to improve our previous calculus. As an unexpected bonus, we could also prove that a narrowing step on any subterm of a given term t with some substitution σ could be skipped if the whole instatiation $t\sigma$ was not a normal form, which again can be very easily checked using the `metaReduce` function.

Our main contributions in this work are a new definition of $\rightarrow^1_{R,A}$ and $R \cup E, A$-rewriting, a definition of a new concept of *narrowable rewrite theory*, and the development of two new narrowing calculi for $\mathcal{E}$-unification and reachability, with the following characteristics:

- a larger class of rewrite theories is accepted by the calculus with respect to previous work, admitting extra variables with no restrictions in equational, membership or rewrite conditions,
- also a larger class of reachability goals is admitted for solving, compared to previous work,
- both calculi use a leftmost strategy,
- both calculi follow a strategy, consisting in applying a calculus rule only if the composition of all computed substitutions remains normalized with respect to all extra variables and all the variables in the initial problem,
- both calculi follow a strategy consisting in normalizing all terms before each narrowing step,
- the calculus for reachability follows a strategy consisting in applying narrowing to a subterm with some substitution only if the whole term remains normalized when instantiated with the same substitution,
- the calculus for reachability follows a strategy consisting in keeping a list of reachability problems. Initially the list holds the original problem. Each new reachability problem generated by the calculus is checked against the current list. If the problem is a renaming and/or reordering of any element in the list, it gets discarded,
- both calculi are sound and weakly complete, i.e., complete with respect to idempotent normalized answers.

The work is structured as follows: in Section 2 basic definitions and properties for rewriting are introduced. Section 3 presents definitions and properties that are specific to this work. In Section 4 we present a version of rewriting that only generates normalized substitutions on extra variables, and prove that the solutions for a unification or a reachability problem can be checked using this restricted rewriting. Section 5 introduces the narrowing calculus for unification. Section 6 introduces the narrowing calculus for reachability. Section 7 shows the calculi at work. In Section 8, related work, conclusions, and future lines of investigation for this work are presented.

As already metioned, this paper is a continuation of a previous one [AMPP14], where non-normalized terms were allowed in both calculi, and extra variables in rules had the same restrictions as in equations.

2 Preliminaries

We assume familiarity with term rewriting and rewriting logic [BM06]. Rewriting logic is always parameterized by an underlying equational logic. This work is focused in membership equational logic [Mes97], an equational logic that generalizes both many-sorted and order-sorted equational theories and that can also handle partial functions [BJ87]. There are several language implementations of rewriting logic, one of them being Maude [CDE$^+$07], a language whose underlying logic is membership equational logic.

2.1 Running example

Example 1. A concurrency specification will be used as running example to explain the definitions in a less abstract way. We review the needed terms. There are `Users` (abbreviated to `u`) `u1`, `u2`, `u3`, and `Tools` (`t`) `t1`, `t2`, `t3`. Several `Users`, separated by commas, are a `UserSet` (`us`) if all the `Users` are different. `emptyU` is the empty `UserSet`. Several `Tools`, separated by semicolons, are a `ToolBox` (`tb`). There will be two `ToolBoxes`, the second one can be seen as a workbench. `emptyT` is the empty `ToolBox`. Each `User` needs two different `Tools` to work which can only be grabbed from the workbench: `u1` needs `t2`; `t3`, `u2` needs `t1`; `t3`, `u3` needs `t1`; `t2`. We also have natural numbers, called `Nat` (`n`), with constant `0` and function successor `s`. We can `count` the number of elements in a `ToolBox`, obtaining a `Nat`, and compare two `Nats` with the function < obtaining a `Boolean` (`b`) value of `ok` when the comparison holds. A `State` (`s`) is composed of two `UserSets`, and two `ToolBoxes`, separated by | symbols. The first `UserSet` holds the `Users` that are not working; the second `UserSet` holds the `Users` that are working. The first `ToolBox` is the main one, while the second `ToolBox` is the workbench. There are two conditions that a `State` must verify: first, the union of both `UserSets` must be a `UserSet`, i.e., each `User` appears only once in the `State`; second, due to the size of the `ToolBox`, the total number of `Tools`, including those being used, cannot exceed four. The initial `State` is called `init`.

The rules that a `State` must follow are:

1. In the initial `State` nobody is working, and there are no `Tools` in the workbench.
2. When the workbench is empty, any two `Tools` that may be in the first `ToolBox` can be put in the workbench.
3. When there are two `Tools` in the workbench and a `User` who is not working needs those tools, he can grab them and work.
4. When a `User` finishes working, he puts the two `Tools` that he was using in the first `ToolBox`.

2.2 Membership equational logic

We present here a sugared version of membership equational logic, similar to the one that it is used for defining Maude specifications. Let $(S, \leq)$ be a partially ordered set of *sorts*, whose *connected components* are the equivalence classes corresponding to the least equivalence relation $\equiv_{\leq}$ containing $\leq$.

Definition 1 (Mel signature). *A* membership equational logic *(*Mel*) signature [BM06] is defined by a* kind-complete *tuple $\Sigma = (K, \Omega, S, \leq)$ meaning that:*

- *K is a set of kinds, where $K \cap S = \emptyset$.*
- *S is split into a K-kinded family of disjoint sets of sorts S_k, i.e., $S = \bigcup_{k \in K} S_k$, such that if $s_i \leq s_j$ and $s_i \in S_k$ then $s_j \in S_k$. We write $[s_i] = k$ and say that the kind of s_i is k, i.e., each sort in a connected component of $(S, \leq)$ has the same kind. $\leq$ is extended so that $s_i \leq k$ iff $s_i \in S_k$, i.e., k is the top sort of its connected component (we also define $[k] = k$ if $k \in K$ for simplicity of notation).*
- *$\Omega = \{\Sigma_{\bar{\kappa},\kappa}\}_{(\bar{\kappa},\kappa) \in (K \cup S)^* \times (K \cup S)}$ is an algebraic signature of* function symbols*, where for each symbol $f \in \Sigma_{\kappa_1 \ldots \kappa_n, \kappa}$ if at least one of the subindices is not a kind, then there exists another function symbol $f \in \Sigma_{[\kappa_1] \ldots [\kappa_n], [\kappa]}$.*

When $f \in \Sigma_{\epsilon,\kappa}$ (ϵ is the empty word), we say that f is a *constant* with *type* (meaning sort or kind) κ. We write $f \in \Sigma_{\kappa}$ instead of $f \in \Sigma_{\epsilon,\kappa}$.

If $f \in \Sigma_{\kappa_1 \ldots \kappa_n,\kappa}$, then we display f as $f : \kappa_1 \ldots \kappa_n \to \kappa$, and say that f has *arity n*. We call this a *rank* declaration for symbol f. Constant symbols have only one rank declaration $f : \to \kappa$ (plus the mandatory $f : \to [\kappa]$ if κ is not a kind). We extend the order $\leq$ on $K \cup S$ to $(K \cup S)^*$, component-wise, where we use the letters w, w' as a synonym for the elements $\kappa_1 \ldots \kappa_n, \kappa'_1 \ldots \kappa'_n \in (K \cup S)^*$ respectively. Then Ω must also satisfy a *monotonicity condition*: $f \in \Sigma_{w,\kappa} \bigcap \Sigma_{w',\kappa'}$ and $w \leq w'$ imply $\kappa \leq \kappa'$. If $f \in \Sigma_{w,\kappa}$ and $t_1, \ldots, t_n$ have type $\kappa_1, \ldots, \kappa_n$ respectively, then the term $f(t_1, \ldots, t_n)$ has type κ. If $\kappa \leq \kappa'$ and the term t has type κ, then t has also type κ'. This means that a term may have several types. In fact, as for every sort s we have that $s \leq [s]$, if a term has only one type then it must be a kind.

In membership equational logic the elements in a sort are well-defined, while the elements in a kind that don't belong to any sort are usually meant to refer to error or undefined elements. Kinds also provide a general way of dealing with partial functions in equational specifications. For instance, in the concurrency specification example a term with sort `State` must not have more than four `Tools`. Otherwise we have an error term with kind `[State]`. The constructor function for sort `State` is partial or total on `State`, we have mentioned one of several limitations, but total on `[State]`.

We allow *mix-fix* notation in Ω, where the symbol _ is used to identify the position of each $\kappa_i \in \bar{\kappa}$. If omitted, we assume the usual functional notation $f(\kappa_1, \ldots, \kappa_n)$, which is an alternative notation admitted for all functions. We assume a family $\mathcal{X} = \{\mathcal{X}_\kappa\}_{\kappa \in (K \cup S)}$ of infinite sets of variables, such that $\kappa \neq \kappa'$ implies $\mathcal{X}_\kappa \cap \mathcal{X}_{\kappa'} = \emptyset$. If κ is a sort then x^i_κ has sort κ (and kind $[\kappa]$), otherwise x^i_κ has kind κ but no sort (we say that x^i_κ is an *unsorted* variable). The set of variables is infinite, but any finite computation will only require a finite number of variables. A term that has no variables in it is said to be *ground*. A term where each variable occurs only once is said to be *linear* (ground terms are linear).

The sets $T_{\Sigma,\kappa}$, $T_\Sigma(\mathcal{X})_\kappa$ denote, respectively, the set of ground Σ-terms with sort or kind κ and the set of Σ-terms with sort or kind κ over $\mathcal{X}$. We use the notation T_Σ as a shortcut for $\bigcup_{\kappa \in (K \cup S)} T_{\Sigma,\kappa}$. We use the notation $T_\Sigma(\mathcal{X})$ as a shortcut for $\bigcup_{\kappa \in (K \cup S)} T_\Sigma(\mathcal{X})_\kappa$. $\mathit{Var}(t) \subseteq \mathcal{X}$ denotes the set of variables in $t \in T_\Sigma(\mathcal{X})$, and we extend this definition in the usual way for other structures in this paper. Σ is assumed to be *sensible* meaning that if $f \in \Sigma_{\kappa_1 \ldots \kappa_n,\kappa}$, $f \in \Sigma_{\kappa'_1 \ldots \kappa'_n,\kappa'}$ and $[\kappa_i] = [\kappa'_i]$ for $i = 1, \ldots, n$ then $[\kappa] = [\kappa']$. We also assume that Σ has non-empty sorts, i.e., $T_{\Sigma,s} \neq \emptyset$ for all $s \in S$.

Example 2. In the concurrency specification example we have, omitting the implied kinded definition for each function in Ω, that $\Sigma = (K, \Omega, S, \leq)$ is:
$K = \{[\mathtt{us}], [\mathtt{tb}], [\mathtt{n}], [\mathtt{b}], [\mathtt{s}]\}, S = \{\mathtt{u}, \mathtt{us}, \mathtt{t}, \mathtt{tb}, \mathtt{n}, \mathtt{b}, \mathtt{s}\}$,
$S_{[\mathtt{us}]} = \{\mathtt{u}, \mathtt{us}\}, S_{[\mathtt{tb}]} = \{\mathtt{t}, \mathtt{tb}\}, S_{[\mathtt{n}]} = \{\mathtt{n}\}, S_{[\mathtt{b}]} = \{\mathtt{b}\}, S_{[\mathtt{s}]} = \{\mathtt{s}\}$,
$\Omega = \{\{\_ \mid \_ \mid \_ \mid \_\}_{\mathtt{us\ us\ tb\ tb},[\mathtt{s}]}, \{\_,\_\}_{[\mathtt{us}]\ [\mathtt{us}],[\mathtt{us}]}, \{\_;\_\}_{\mathtt{tb\ tb},\mathtt{tb}}, \{\mathtt{count}\}_{\mathtt{tb},\mathtt{n}}, \{\_<\_\}_{\mathtt{n\ n},\mathtt{b}}$,
$\{\mathtt{s}\}_{\mathtt{n},\mathtt{n}}, \{\mathtt{u1}, \mathtt{u2}, \mathtt{u3}\}_{\mathtt{u}}, \{\mathtt{emptyU}\}_{\mathtt{us}}, \{\mathtt{t1}, \mathtt{t2}, \mathtt{t3}\}_{\mathtt{t}}, \{\mathtt{emptyT}\}_{\mathtt{tb}}, \{\mathtt{0}\}_{\mathtt{n}}, \{\mathtt{ok}\}_{\mathtt{b}}, \{\mathtt{init}\}_{\mathtt{s}}\}$.

We explain the notation used in Ω: $\{\_,\_\}_{[\mathtt{us}]\ [\mathtt{us}],[\mathtt{us}]}$ means that there is a mix-fix function symbol _,_ such that if u_1 and u_2 are terms with kind `[UserSet]` then u_1, u_2 is a term with kind `[UserSet]`, but it doesn't have to have any sort (for instance, `a` is a `UserSet`, but `a, a` is not a `UserSet`). We will later define the terms that truly are `UserSets`. It is possible to use functional notation for all function symbols, but this notation usually turns term interpretation into a burden for the reader, so we prefer to use mix-fix notation for some of our function symbols.

Positions in a term t: as previously said, a term t can be always expressed in functional notation as $f(t_1, \ldots, t_n)$. Then we can picture t as a tree with root f and children t_1 at position $1, \ldots, t_n$ at position n. We refer to the root position of t as ϵ and to the other positions of t as lists of nonzero natural numbers separated by dots, $i_1.i_2 \ldots i_m$, meaning the position $i_2 \ldots i_m$ of t_{i_1}. The set of positions of a term is written $\mathit{Pos}(t)$. The set of nonvariable positions of a term is written $\mathit{Pos}_\Sigma(t)$. $t|_p$ is the subtree of t below position p. $t[u]_p$ is the replacement in t of the subterm at position p with a term u. As an example, if t is $f(g(a,b),c)$, then $t|_1$ is $g(a,b)$, $t|_{1.2}$ is b, and $t[d]_{1.2}$ is $f(g(a,d),c)$.

A Mel signature Σ is said to be *preregular* iff for each n, for every function symbol f with arity n, and for every $\kappa_1 \ldots \kappa_n \in (K \cup S)^n$, if the set S_f containing all the sorts s' that appear in rank declarations in Σ of the form $f : \kappa'_1 \ldots \kappa'_n \to \kappa'$ such that $\kappa_i \leq \kappa'_i$, for $1 \leq i \leq n$, is not empty (so a term $f(t_1, \ldots, t_n)$ where t_i has type κ_i for $1 \leq i \leq n$ would be a Σ-term), then S_f has a least sort. Preregularity guarantees that every Σ-term t has a *least sort*, denoted $\mathit{ls}(t)$, among all the sorts that t has because of the different rank declarations that can be applied to t, which is the most accurate classification for t, i.e., for any rank declaration $f : \kappa_1 \ldots \kappa_n \to \kappa$ that can be applied to t it is true that $\mathit{ls}(t) \leq \kappa$.

A *substitution* $\sigma : \mathcal{X} \to T_\Sigma(\mathcal{X})$ is a function that matches the identity function in all $\mathcal{X}$ except for a finite set of variables $\mathcal{Y} \subseteq \mathcal{X}$. A substitution is *well-formed* if for each variable $y_\kappa \in \mathcal{Y}$ we have that $\mathit{ls}(y_\kappa\sigma) \leq \kappa$. In this text we assume that all substitutions are well-formed unless stated otherwise. Substitutions are written as $\sigma = \{y^1_{\kappa_1} \mapsto t_1, \ldots, y^n_{\kappa_n} \mapsto t_n\}$ where $\mathit{Dom}(\sigma) = \{y^1_{\kappa_1}, \ldots, y^n_{\kappa_n}\}$ and $\mathit{Ran}(\sigma) = \bigcup_{i=1}^n \mathit{Var}(t_i)$. The identity substitution is displayed as id. Substitutions are homomorphically extended to terms in $T_\Sigma(\mathcal{X})$ (and also to the rest of syntactic structures introduced along this paper, such as equations, goals, etc.). The restriction $\sigma|_\mathcal{V}$ of σ to a set of variables $\mathcal{V}$ is defined as $x\sigma|_\mathcal{V} = x\sigma$ if $x \in \mathcal{V}$ and $x\sigma|_\mathcal{V} = x$ otherwise. Composition of two substitutions σ and σ' is denoted by $\sigma\sigma'$, with $x(\sigma\sigma') = (x\sigma)\sigma'$. For a substitution σ, if $\sigma\sigma = \sigma$ we say that σ is *idempotent.* For substitutions σ and σ', where $\mathit{Dom}(\sigma) \cap \mathit{Dom}(\sigma') = \emptyset$, we denote their union by $\sigma \cup \sigma'$.

A Σ-*equation* is an expression of the form $t = t'$. A Σ-*equation* $t = t'$ is said to be:

- *Regular* iff $\mathit{Var}(t) = \mathit{Var}(t')$.
- *Sort-preserving* iff for each substitution σ, we have $t\sigma \in T_\Sigma(\mathcal{X})_\kappa$ $(\kappa \in K \cup S)$ implies $t'\sigma \in T_\Sigma(\mathcal{X})_\kappa$ and vice versa.
- *Left (or right) linear* iff t (resp., t') is linear.
- *Linear* iff it is both left and right linear.

A set of equations E is said to be regular, or sort-preserving, or (left or right) linear, if each equation in it is so.

Definition 2 (Mel theory). *A* Mel *theory [BM06] is a pair $(\Sigma, \mathcal{E})$, where Σ is a* Mel *signature and $\mathcal{E}$ is a finite set of (possibly labeled)* Mel *sentences, either conditional equations or conditional memberships of the forms:*

$$t = t' \ \mathit{if} \ A_1 \wedge \ldots \wedge A_n, \qquad t : s \ \mathit{if} \ A_1 \wedge \ldots \wedge A_n,$$

where $t = t'$ is a Σ-equation, $t : s$, $s \in S$, is a unary membership predicate stating that t is a term with sort s, provided that the condition holds, and each A_i can be of the form $t = t'$, $t : s$ or $t := t'$ (a matching equation).

Matching equations are treated as ordinary Σ-equations. They are a warning that new *extra* variables appear in t, in a concrete way, imposing a limitation in the syntax of the equation. We also admit unconditional sentences in $\mathcal{E}$. $x_{s_1} : s_2$ is an unconditional membership expressing $s_1 \leq s_2$. For each variable $x_s \in \mathcal{X}_s$, where $s \in S$, we have that $x_s : s \in \mathcal{E}$. As an exception, there are two types of unconditional memberships over kinds, instead of sorts, that are implied by the MEL signature: if $f \in \Sigma_{\kappa_1 \ldots \kappa_n, k}$, $k \in K$ then $f(x_{\kappa_1}, \ldots, x_{\kappa_n}) : k \in \mathcal{E}$; also for each variable $x_\kappa \in \mathcal{X}_\kappa$ such that $[\kappa] = k$, $x_\kappa : k \in \mathcal{E}$. Throughout this paper we will assume that all signatures are preregular and all their equations and memberships $t = t'$, $t := t'$ and $t : s$, satisfy the conditions $[ls(t)] = [ls(t')]$ and $[ls(t)] = [s]$, that is, they are well-formed.

A MEL signature Σ imposes an associated set of memberships to any MEL theory $(\Sigma, \mathcal{E})$: for each $s_1, s_2 \in S$ such that $s_1 < s_2$, there is an associated unconditional membership $x_{s_1} : s_2$ in $\mathcal{E}$; each constant definition $c \in \Sigma_\kappa$ has an associated unconditional membership $c : \kappa$ in $\mathcal{E}$; each non constant definition $f \in \Sigma_{\kappa_1 \ldots \kappa_n, s}$, so $n \geq 1$, has an associated conditional membership $f(x_{[\kappa_1]}, \ldots, x_{[\kappa_n]}) : s$ *if* $x_{[\kappa_1]} : \kappa_1 \wedge \ldots \wedge x_{[\kappa_n]} : \kappa_n$ in $\mathcal{E}$; each definition $f \in \Sigma_{k_1 \ldots k_n, k}$, with $n \geq 0$, has an associated unconditional membership $f(x_{k_1}, \ldots, x_{k_n}) : k$ in $\mathcal{E}$. A MEL theory whose only memberships are the associated ones is an *order-sorted* theory, or a *many-sorted* theory if $<$ is the empty relation, where we can use all known results for these equational theories.

Given a MEL sentence ϕ, we denote by $\mathcal{E} \vdash \phi$ the fact that ϕ can be deduced from $\mathcal{E}$ using the rules in Figure 1 [BM06, BM12]; for an equation $t = t'$, $\mathcal{E} \vdash t = t'$ is also written $t =_{\mathcal{E}} t'$, for a membership $t : s$, $\mathcal{E} \vdash t : s$ is also written $t :_{\mathcal{E}} s$. These rules, where the symbol $=$ stands for $=$ or $:=$ indistinctly, specify a sound and complete calculus.

$$\frac{t \in T_\Sigma(\mathcal{X})}{t = t}\ \texttt{Reflexivity} \qquad \frac{t = t'}{t' = t}\ \texttt{Symmetry}$$

$$\frac{t'{:}s \quad t = t'}{t{:}s}\ \texttt{Membership} \qquad \frac{t_1 = t_2 \quad t_2 = t_3}{t_1 = t_3}\ \texttt{Transitivity}$$

$$\frac{f \in \Sigma_{k_1 \cdots k_n, k}\ t_i = t'_i \quad t_i, t'_i {\in} T_\Sigma(X)_{k_i}, 1{\leq}i{\leq}n}{f(t_1, \ldots, t_n) = f(t'_1, \ldots, t'_n)}\ \texttt{Congruence}$$

$$\frac{(\ A_0\ \mathit{if}\ \bigwedge_{i=1}^n A_i) {\in} E \quad \sigma{:}X {\to} T_\Sigma(Y) \quad A_1\sigma \ldots A_n\sigma}{A_0\sigma}\ \texttt{Replacement}$$

Fig. 1. Deduction rules for membership equational logic.

Example 3. The MEL theory for the concurrency specification example has $\Sigma = (K, \Omega, S, \leq)$ and $\mathcal{E}$ is the set of MEL sentences in Table 1, where the first row of MEL sentences represents the subsort ordering in S. We omit the implicit subsorts for each kind, and the implicit memberships for each variable and kinded function. For executability requirements of the theory, that will be later defined, associativity, commutativity, and identity axioms are defined over kinds:

The conditional membership sentences for `State` (`s`) take into account that when checking the total number of `Tools`, any working `User` is holding two `Tools`. When two `Users` are working, both `ToolBoxes` must be empty. If necessary it is also checked that the union of the working `UserSet` and the non working `UserSet` is also a `UserSet`.

2.3 Unification

Given a MEL theory $(\Sigma, \mathcal{E})$, the $\mathcal{E}$*-subsumption* preorder $\ll_{\mathcal{E}}$ on $T_\Sigma(\mathcal{X})_k$ is defined by $t \ll_{\mathcal{E}} t'$ if there is a substitution σ such that $t =_{\mathcal{E}} t'\sigma$. For substitutions σ, ρ and a set of variables $\mathcal{V}$ we define $\sigma|_{\mathcal{V}} \ll_{\mathcal{E}} \rho|_{\mathcal{V}}$ if there is a substitution η such that $\sigma|_{\mathcal{V}} =_{\mathcal{E}} (\rho\eta)|_{\mathcal{V}}$. Then we say that ρ is more general than σ with respect to $\mathcal{V}$. When $\mathcal{V}$ is not specified, we assume that $Dom(\rho) \subseteq Dom(\sigma)$ and say that ρ is more general than σ.

$x_{\mathtt{u}} : \mathtt{us}$	$x_{\mathtt{t}} : \mathtt{tb}$	(subsorts)
$(x_{[\mathtt{us}]}, y_{[\mathtt{us}]}), z_{[\mathtt{us}]} = x_{[\mathtt{us}]}, (y_{[\mathtt{us}]}, z_{[\mathtt{us}]})$	$(x_{[\mathtt{tb}]}; y_{[\mathtt{tb}]}); z_{[\mathtt{tb}]} = x_{[\mathtt{tb}]}; (y_{[\mathtt{tb}]}; z_{[\mathtt{tb}]})$	(associativity)
$x_{[\mathtt{us}]}, y_{[\mathtt{us}]} = y_{[\mathtt{us}]}, x_{[\mathtt{us}]}$	$x_{[\mathtt{tb}]}; y_{[\mathtt{tb}]} = y_{[\mathtt{tb}]}; x_{[\mathtt{tb}]}$	(commutativity)
$x_{[\mathtt{us}]}, \mathtt{emptyU} = x_{[\mathtt{us}]}$	$x_{[\mathtt{tb}]}; \mathtt{emptyT} = x_{[\mathtt{tb}]}$	(identity)
$x_{\mathtt{us}} \mid \mathtt{emptyU} \mid z_{\mathtt{tb}} \mid w_{\mathtt{tb}} : \mathtt{s}\ \mathit{if}\ \mathtt{count}(z_{\mathtt{tb}}; w_{\mathtt{tb}}) < \mathtt{s}(\mathtt{s}(\mathtt{s}(\mathtt{s}(\mathtt{s}(0))))) = \mathtt{ok}$		
$x_{\mathtt{us}} \mid y_{\mathtt{u}} \mid z_{\mathtt{tb}} \mid w_{\mathtt{tb}} : \mathtt{s}\ \mathit{if}\ y_{\mathtt{u}}, x_{\mathtt{us}} : \mathtt{us} \wedge \mathtt{count}(z_{\mathtt{tb}}; w_{\mathtt{tb}}) < \mathtt{s}(\mathtt{s}(\mathtt{s}(0))) = \mathtt{ok}$		
$x_{\mathtt{us}} \mid y_{\mathtt{u}}, y'_{\mathtt{u}} \mid \mathtt{emptyT} \mid \mathtt{emptyT} : \mathtt{s}\ \mathit{if}\ y_{\mathtt{u}}, y'_{\mathtt{u}}, x_{\mathtt{us}} : \mathtt{us}$		
$\mathtt{u1}, \mathtt{u2} : \mathtt{us}$ $\quad \mathtt{u1}, \mathtt{u3} : \mathtt{us}$ $\quad \mathtt{u2}, \mathtt{u3} : \mathtt{us}$ $\quad \mathtt{u1}, \mathtt{u2}, \mathtt{u3} : \mathtt{us}$		
$\mathtt{count}(\mathtt{emptyT}) = 0$	$\mathtt{count}(x_{\mathtt{t}}; y_{\mathtt{tb}}) = \mathtt{s}(\mathtt{count}(y_{\mathtt{tb}}))$	
$0 < \mathtt{s}(x_{\mathtt{n}}) = \mathtt{ok}$	$\mathtt{s}(x_{\mathtt{n}}) < \mathtt{s}(y_{\mathtt{n}}) = x_{\mathtt{n}} < y_{\mathtt{n}}$	

Table 1. MEL sentences for the concurrency specification example

Given a MEL theory $(\Sigma, \mathcal{E})$, a *system of sentences* F is a conjunction of the form $u_1 = v_1 \wedge \ldots \wedge u_n = v_n \wedge t_1 : s_1 \wedge \ldots \wedge t_m : s_m$ (= standing for = or :=) where for $1 \leq i \leq n$, $u_i = v_i$ is a well-formed Σ-equation, and for $1 \leq j \leq m$, $t_j : s_j$ is a well-formed membership. We define $\mathit{Var}(F) = \bigcup_{i=1}^{n}(\mathit{Var}(u_i) \cup \mathit{Var}(v_i)) \cup \bigcup_{j=1}^{m} \mathit{Var}(t_j)$. An *$\mathcal{E}$-solution* for F is a substitution σ such that $u_i\sigma =_{\mathcal{E}} v_i\sigma$ for $1 \leq i \leq n$ and $t_j\sigma :_{\mathcal{E}} s_j$ for $1 \leq j \leq m$. Note that the condition in a conditional MEL sentence is a system of sentences. When F is a conjunction of Σ-equations we say that F is a *system of equations*. An $\mathcal{E}$-solution for a system of equations is called an *$\mathcal{E}$-unifier*. For F a system of equations and $\mathcal{V} = \mathit{Var}(F) \subseteq \mathcal{W}$, a set of substitutions $\mathit{CSU}_{\mathcal{E}}^{\mathcal{W}}(F)$ is said to be a *complete set of $\mathcal{E}$-unifiers* of F away from $\mathcal{W}$ iff:

- each substitution σ in $\mathit{CSU}_{\mathcal{E}}^{\mathcal{W}}(F)$ is an $\mathcal{E}$-unifier of F;
- for any $\mathcal{E}$-unifier ρ of F there is a substitution σ in $\mathit{CSU}_{\mathcal{E}}^{\mathcal{W}}(F)$ such that $\rho|_{\mathcal{W}} \ll_{\mathcal{E}} \sigma|_{\mathcal{W}}$;
- for each substitution σ in $\mathit{CSU}_{\mathcal{E}}^{\mathcal{W}}(F)$, $\mathit{Dom}(\sigma) \subseteq \mathcal{V}$ and $\mathit{Ran}(\sigma) \cap \mathcal{W} = \emptyset$.

We will usually write $\mathit{CSU}_{\mathcal{E}}$ in the understanding that $\mathcal{W}$ is the set of all the variables that have already appeared in the current calculation.

This notion was introduced by Plotkin [Plo72]. An $\mathcal{E}$-unification algorithm is *complete* if for any given system of equations it generates a complete set of $\mathcal{E}$-unifiers, which may not be finite. An $\mathcal{E}$-unification algorithm is said to be *finitary* and complete if it terminates after generating a finite and complete set of solutions.

2.4 Rewriting logic

A rewrite theory $\mathcal{R} = (\Sigma, \mathcal{E}, R)$ consists of a MEL theory $(\Sigma, \mathcal{E})$ together with a finite set R of *conditional rewrite rules* each of which has the form

$$l \to r\ \mathit{if}\ \bigwedge_h p_h = q_h \wedge \bigwedge_i u_i := v_i \wedge \bigwedge_j w_j : s_j \wedge \bigwedge_k l_k \to r_k,$$

where l, r, and also each pair l_k, r_k, are Σ-terms of the same kind, and the rest of conditions fulfill the same requirements pointed out for MEL sentences. We will sometimes write $l \to r\ \mathit{if}\ C$ as a shortcut. Rewrite rules can also be unconditional. Equational and membership conditions are intended to be solved within the MEL theory $(\Sigma, \mathcal{E})$, i.e., no rewriting with rules from R is allowed on those conditions, whereas a *reachability condition* $l_k \to r_k$ means that r_k is *reachable* from l_k as defined below. We define $\mathit{Var}(l \to r) = \mathit{Var}(l) \cup \mathit{Var}(r)$.

Definition 3 ($\to^1_{\mathcal{R}}$ **relation).** *Given a rewrite theory $\mathcal{R} = (\Sigma, \mathcal{E}, R)$, a term $t \in T_\Sigma(\mathcal{X})$, a position $p \in Pos(t)$, and a substitution σ, a rewrite rule $l \to r$ if C specifies a* one-step transition *$t[l\sigma]_p \to^1_{\mathcal{R}} t[r\sigma]_p$ iff $t \equiv t[l\sigma]_p$ ($\equiv$ standing for a syntactic equality), and the instantiated condition $C\sigma$ holds. We sometimes write $(t, t') \in \to^1_{\mathcal{R}}$ when $t \to^1_{\mathcal{R}} t'$. The transitive (resp., transitive and reflexive) closure of any relation $\to^1_R$ is denoted $\to^+_R$ (resp., $\to^*_R$). For any relation $\to^1_R$, if $(t, t') \in \to^*_R$ we say that t' is reachable from t in $\to^1_R$.*

Example 4. In the concurrency specification example, R has as elements the conditional rewrite rules in Table 2.

$$\mathtt{init} \;\to\; x_{\mathtt{us}} \mid \mathtt{emptyU} \mid z_{\mathtt{tb}} \mid \mathtt{emptyT} \;\mathit{if}\; x_{\mathtt{us}} \mid \mathtt{emptyU} \mid z_{\mathtt{tb}} \mid \mathtt{emptyT} : \mathtt{s}$$

$$x_{\mathtt{us}} \mid y_{\mathtt{us}} \mid u_{\mathtt{t}}; v_{\mathtt{t}}; z_{\mathtt{tb}} \mid \mathtt{emptyT} \;\to\; x_{\mathtt{us}} \mid y_{\mathtt{us}} \mid z_{\mathtt{tb}} \mid u_{\mathtt{t}}; v_{\mathtt{t}} \;\mathit{if}\; x_{\mathtt{us}} \mid y_{\mathtt{us}} \mid z_{\mathtt{tb}} \mid u_{\mathtt{t}}; v_{\mathtt{t}} : \mathtt{s}$$

$$\mathtt{u1}, x_{\mathtt{us}} \mid y_{\mathtt{us}} \mid z_{\mathtt{tb}} \mid \mathtt{t2}; \mathtt{t3} \;\to\; x_{\mathtt{us}} \mid \mathtt{u1}, y_{\mathtt{us}} \mid z_{\mathtt{tb}} \mid \mathtt{emptyT} \;\mathit{if}\; x_{\mathtt{us}} \mid \mathtt{u1}, y_{\mathtt{us}} \mid z_{\mathtt{tb}} \mid \mathtt{emptyT} : \mathtt{s}$$

$$\mathtt{u2}, x_{\mathtt{us}} \mid y_{\mathtt{us}} \mid z_{\mathtt{tb}} \mid \mathtt{t1}; \mathtt{t3} \;\to\; x_{\mathtt{us}} \mid \mathtt{u2}, y_{\mathtt{us}} \mid z_{\mathtt{tb}} \mid \mathtt{emptyT} \;\mathit{if}\; x_{\mathtt{us}} \mid \mathtt{u2}, y_{\mathtt{us}} \mid z_{\mathtt{tb}} \mid \mathtt{emptyT} : \mathtt{s}$$

$$\mathtt{u3}, x_{\mathtt{us}} \mid y_{\mathtt{us}} \mid z_{\mathtt{tb}} \mid \mathtt{t1}; \mathtt{t2} \;\to\; x_{\mathtt{us}} \mid \mathtt{u3}, y_{\mathtt{us}} \mid z_{\mathtt{tb}} \mid \mathtt{emptyT} \;\mathit{if}\; x_{\mathtt{us}} \mid \mathtt{u3}, y_{\mathtt{us}} \mid z_{\mathtt{tb}} \mid \mathtt{emptyT} : \mathtt{s}$$

$$x_{\mathtt{us}} \mid \mathtt{u1}, y_{\mathtt{us}} \mid z_{\mathtt{tb}} \mid w_{\mathtt{tb}} \;\to\; \mathtt{u1}, x_{\mathtt{us}} \mid y_{\mathtt{us}} \mid \mathtt{t2}; \mathtt{t3}; z_{\mathtt{tb}} \mid w_{\mathtt{tb}} \;\mathit{if}\; \mathtt{u1}, x_{\mathtt{us}} \mid y_{\mathtt{us}} \mid \mathtt{t2}; \mathtt{t3}; z_{\mathtt{tb}} \mid w_{\mathtt{tb}} : \mathtt{s}$$

$$x_{\mathtt{us}} \mid \mathtt{u2}, y_{\mathtt{us}} \mid z_{\mathtt{tb}} \mid w_{\mathtt{tb}} \;\to\; \mathtt{u2}, x_{\mathtt{us}} \mid y_{\mathtt{us}} \mid \mathtt{t1}; \mathtt{t3}; z_{\mathtt{tb}} \mid w_{\mathtt{tb}} \;\mathit{if}\; \mathtt{u2}, x_{\mathtt{us}} \mid y_{\mathtt{us}} \mid \mathtt{t1}; \mathtt{t3}; z_{\mathtt{tb}} \mid w_{\mathtt{tb}} : \mathtt{s}$$

$$x_{\mathtt{us}} \mid \mathtt{u3}, y_{\mathtt{us}} \mid z_{\mathtt{tb}} \mid w_{\mathtt{tb}} \;\to\; \mathtt{u3}, x_{\mathtt{us}} \mid y_{\mathtt{us}} \mid \mathtt{t1}; \mathtt{t2}; z_{\mathtt{tb}} \mid w_{\mathtt{tb}} \;\mathit{if}\; \mathtt{u3}, x_{\mathtt{us}} \mid y_{\mathtt{us}} \mid \mathtt{t1}; \mathtt{t2}; z_{\mathtt{tb}} \mid w_{\mathtt{tb}} : \mathtt{s}$$

Table 2. Rewrite rules for the concurrency example

Without losing generality, we will always assume that we use instances of the rules where all the variables that appear on them are fresh.

The relation $\to^1_{R/\mathcal{E}}$ on $T_\Sigma(\mathcal{X})$ is defined as $=_\mathcal{E}; \to^1_R; =_\mathcal{E}$. This relation $\to^1_{R/\mathcal{E}}$ on $T_\Sigma(\mathcal{X})$ induces a relation $\to^1_{R/\mathcal{E}}$ on $T_{\Sigma/\mathcal{E}}(\mathcal{X})$, the equivalence relation modulo $\mathcal{E}$, by $[t]_\mathcal{E} \to^1_{R/\mathcal{E}} [t']_\mathcal{E}$ iff $t \to^1_{R/\mathcal{E}} t'$.

A rewrite rule $l \to r$ *if* C is *sort-decreasing* if for each substitution σ we have that $l\sigma \in T_\Sigma(\mathcal{X})_\kappa$ ($\kappa \in K \cup S$) and $C\sigma$ is verified implies $r\sigma \in T_\Sigma(\mathcal{X})_\kappa$.

For any relation $\to^1_R$ we say that a term t is $\to_R$*-irreducible* (or just R-irreducible) if there is no term t' such that $t \to^1_R t'$ and we say that a substitution is *R-normalized* (or normalized if R can be deduced from the context) if $x\sigma$ is R-irreducible for all $x \in Dom(\sigma)$. We also say that a term t is *strongly* R-irreducible if for every R-normalized substitution σ the term $t\sigma$ is R-irreducible.

The relation $\to^1_R$ is *terminating* if there are no infinite rewriting sequences in $\to^1_R$. The relation $\to^1_R$ is *confluent* if whenever $t \to^*_R t_1$ and $t \to^*_R t_2$, there exists a term t_3 such that $t_1 \to^*_R t_3$ and $t_2 \to^*_R t_3$. In a confluent, terminating, sort-decreasing, membership rewrite theory, for each term $t \in T_\Sigma(\mathcal{X})$, there is a unique (up to $\mathcal{E}$-equivalence) $R/\mathcal{E}$-irreducible term t' obtained by rewriting to *canonical* form, denoted by $t \to^!_{R/\mathcal{E}} t'$, or $t \downarrow_{R/\mathcal{E}}$ when t' is not relevant.

One problem that can arise when trying to decide $t \to^1_R t'$ in a rewrite theory is that although $\to^1_R$ is terminating, an attempt to prove a condition in a rule, building a so-called *well-formed proof tree* [LM09], may generate a recursive infinite check of conditions, and a corresponding infinite well-formed proof tree. This leads us to the notion of *operational termination.*

Definition 4. *The relation $\to^1_R$ is* operationally terminating *if there are no infinite well-formed proof trees.*

This notion of operational termination was presented by Lucas, Marché and Meseguer [LMM05] in an attempt to exclude those conditional term rewriting systems like the one consisting of the single conditional rule:

$$a \to b \;\; \mathit{if} \;\; f(a) \to b$$

The absence of unconditional rules makes the relation $\to^1$ trivially empty, hence terminating. Nevertheless, when trying to reduce the term a, most implementations will loop because of the following infinite derivation tree:

$$\cfrac{\cfrac{\cfrac{\cdots}{a \to b}}{f(a) \to b}}{a \to b}$$

The condition of operational termination states that such derivation trees don't exist.

3 Narrowing and narrowable rewrite theories

3.1 Associated rewrite theory

For a rewrite theory $\mathcal{R} = (\Sigma, \mathcal{E}, R)$, whether a one step rewrite $t \to^1_{R/\mathcal{E}} t'$ holds is undecidable in general, because it involves searching a potentially infinite, and even non computable, set $[t]_\mathcal{E}$ and checking if for any of its elements $u \in [t]_\mathcal{E}$ we have that $u \to^1_R t''$ and $t'' =_\mathcal{E} t'$. The approach taken to solve this problem is to decompose $\mathcal{E}$ into a disjoint union $E \cup A$, with A a set of equational axioms (such as associativity, and/or commutativity, and/or identity) which must be regular, linear, sort-preserving, and where any variable appearing in an axiom must be a kinded variable. Then we define the relations $\to^1_{E,A}$ and $\to^1_{R,A}$ on $T_\Sigma(\mathcal{X})$ which, under certain assumptions on $\mathcal{R}$, will make $t \to^1_{R/\mathcal{E}} t'$ semi-decidable (or decidable under more restricted assumptions). From now on we will use the notations $\mathcal{E}$ and $E \cup A$ as synonyms, assuming this decomposition.

Any Mel theory $(\Sigma, \mathcal{E})$ has a corresponding rewrite theory $\mathcal{R}_E = (\Sigma', A, R_E)$ associated to it [DLM+08a], where A has the properties listed before, that under certain assumptions allows us to check $\mathcal{E}$-solutions for systems of sentences using rewriting. Under some additional assumptions this will be a finite process. The associated rewrite theory is constructed in the following way: we add a new connected component with sort *Truth*, a new constant *tt* of this sort to Σ, for each sort $s \in S$ a new function symbol $\_{:}s \; : \; [s] \to \mathit{Truth}$, and for each kind $k \in K$ a new function symbol $\mathit{eq} : \; k \; k \to \mathit{Truth}$. There are rules $\mathit{eq}(x_k, x_k) \to \mathit{tt}$ in R_E for each kind $k \in K$. For each equation or membership in E

$$t = t' \;\; \mathit{if} \;\; A_1 \wedge \ldots \wedge A_n \qquad t : s \;\; \mathit{if} \;\; A_1 \wedge \ldots \wedge A_n,$$

R_E has a conditional rule of the form

$$t \to t' \;\; \mathit{if} \;\; A'_1 \wedge \ldots \wedge A'_n \qquad t{:}s \to \mathit{tt} \;\; \mathit{if} \;\; A'_1 \wedge \ldots \wedge A'_n$$

where if A_i is $t_i : s_i$ then A'_i is $t_i{:}s_i \to \mathit{tt}$, if A_i is $t_i := t'_i$ then A'_i is $t'_i \to t_i$, and if A_i is $t_i = t'_i$ then A'_i is $\mathit{eq}(t_i, t'_i) \to \mathit{tt}$.

3.2 E, A-rewriting. R, A-rewriting. Closure under A-extensions

Now we define a set of relations where rules are applied not by strict matching, like in $\to^1_R$, or by matching modulo the equational theory, like in $\to^1_{R/\mathcal{E}}$, which may be intractable, but in an intermediate way: by matching modulo axioms A, for which we have fast dedicated algorithms.

The relation $\to_{E,A}$ is defined as $(\to^*_{E,A}; =_A)$, $\to^1_{R\cup E,A}$ as $(\to^1_{R,A} \cup \to^1_{E,A})$, $\to_{R\cup E,A}$ as $(\to^*_{R\cup E,A}; =_\mathcal{E})$, and $\to_{R/\mathcal{E}}$ as $(\to^*_{R/\mathcal{E}}; =_\mathcal{E})$, where the relations $\to^1_{E,A}$ and $\to^1_{R,A}$ are defined below.

Definition 5 (*E,A*-rewriting). *Given a rewrite theory $\mathcal{R}_E = (\Sigma', A, R_E)$, associated to a* Mel *theory $(\Sigma, \mathcal{E})$, and terms $t, t' \in T_\Sigma(\mathcal{X})$, $t \rightarrow^1_{E,A} t'$ if there is a rule $l \rightarrow r$ if $\bigwedge_{i \in I} A'_i$ in R_E, a position $p \in Pos(t)$, and a substitution σ such that $t|_p =_A l\sigma$ (A-matching), $t' = t[r\sigma]_p$, and for all $i \in I$ $t_i\sigma \rightarrow_{E,A} t'_i\sigma$.*

It is important to point out that not only $t \rightarrow_{E,A} t$, for all $t \in T_\Sigma(\mathcal{X})$, without applying any rewrite rule from R_E, but we also have that if $t =_A t'$ then $t \rightarrow_{E,A} t'$, again without applying any rewrite rule from R_E.

Definition 6 (*R,A*-rewriting). *Given a rewrite theory $\mathcal{R} = (\Sigma, \mathcal{E}, R)$, and terms $t, t' \in T_\Sigma(\mathcal{X})$, $t \rightarrow^1_{R,A} t'$ if there is a rule $l \rightarrow r$ if $\bigwedge_{i \in I} A_i$ in R, a position $p \in Pos(t)$, and a substitution σ such that $t|_p =_A l\sigma$, $t' = t[r\sigma]_p$, and for all $i \in I$:*

- *If A_i is of the form $t_i \rightarrow t'_i$ then $t_i\sigma \rightarrow_{R \cup E,A} t'_i\sigma$.*
- *If A_i is of the form $u = v$, $u := v$, or $u : s$, and we consider A'_i (as in R_E), which is of the form $t_i \rightarrow t'_i$, then $t_i\sigma \rightarrow_{E,A} t'_i\sigma$.*

Under certain assumptions on the rewrite theories, the task of finding the substitution σ to apply in $\rightarrow^1_{E,A}$ or $\rightarrow^1_{R,A}$ becomes always decidable. We will then speak of an *executable* rewrite theory. A more proper name for the relations would have been $\rightarrow^1_{R_E,A}$ and $\rightarrow_{R(E),A}$, but we will use $\rightarrow_{E,A}$ and $\rightarrow_{R,A}$ for simplicity of notation.

The standard definition for $\rightarrow_{R \cup E,A}$ is $\rightarrow^*_{R \cup E,A}; =_A$, because it allows the instantiation of the new variables that may appear in the right term of the reachability goal by matching them against the left term of the goal if this left term meets the executability requirements. Our setting for narrowing allows us to replace $=_A$ with $=_\mathcal{E}$ and relax the requirements, that transform from being executable to being *narrowable*, a new concept that we formalize later in the work. Using this definition we can use Maude to solve narrowing problems but Maude's rewrite engine cannot be used to verify the solutions obtained for the theories belonging to this wider class of narrowable rewrite theories except when they are also executable.

We plan to replace $=_\mathcal{E}$ and $:_\mathcal{E}$ with $\rightarrow_{E,A}$, and $\rightarrow_{\mathcal{R}/\mathcal{E}}$ with $\rightarrow_{R \cup E,A}$, but there is a problem that must be solved to make these replacements feasible. Consider a rewrite theory $\mathcal{R}$ with only one sort s, and whose only rule is $f(a,b) \rightarrow c$, where f is associative and commutative ($E = \emptyset$). The term $f(f(a,a),b)$ is a normal form in $\rightarrow^1_{R \cup E,A}$, but $f(f(a,a),b) \rightarrow^1_{R/\mathcal{E}} f(a,c)$, because $f(f(a,a),b) =_A f(a, f(a,b))$, so the relations are different. This problem would not happen if $\mathcal{R}$ had another rule $f(x_s, f(a,b)) \rightarrow f(x_s, c)$ that could be applied on top of the term $f(f(a,a),b)$ with matching $x_s \mapsto a$, modulo associativity and commutativity, leading to $f(f(a,a),b) \rightarrow^1_{R \cup E,A} f(a,c)$. Rewrite theories, including those associated to a Mel theory, that have these rules, avoiding such problems, are called *closed under A-extensions* [Mes14].

Definition 7 (Closure under *A*-extensions). *Let $\mathcal{R} = (\Sigma, E \cup A, R)$ be a rewrite theory, and let $l \rightarrow r$ if C be a rule in R. Without loss of generality we asume that $Var(A) \cap Var(l \rightarrow r \text{ if } C) = \emptyset$. If this is not the case,* only the variables of *A will be renamed; the variables of $l \rightarrow r$ if C* will never be *renamed. We then define the set of A*-extensions *as the set:*

$$Ext_A(l \rightarrow r \text{ if } C) = \{u[l]_p \rightarrow u[r]_p \text{ if } C \mid u = v \in A \cup A^{-1} \wedge p \in Pos_\Sigma(u) - \{\epsilon\} \wedge CSU_A(l = u|_p) \neq \emptyset\}$$

where, by definition, $A^{-1} = \{v = u \mid u = v \in A\}$.

Given two rules $l \rightarrow r$ if C and $l' \rightarrow r'$ if C with the same *condition C we say that $l \rightarrow r$ if C A*-subsumes *$l' \rightarrow r'$ if C iff there is a substitution σ such that: (i) $Dom(\sigma) \cap Var(C) = \emptyset$, (ii) $l' =_A l\sigma$, and (iii) $r' =_A r\sigma$.*

We call $\mathcal{R} = (\Sigma, E \cup A, R)$ closed under A-extensions *iff for any rule $l \rightarrow r$ if C in R, each rule $l' \rightarrow r'$ if C in $Ext_A(l \rightarrow r \text{ if } C)$ is subsumed by some rule in R.*

Theorem 2 and Corollary 3 in [Mes14] can be applied in a straightfordward way to $\rightarrow^1_{E,A}$ and $\rightarrow^1_{R,A}$, and we get the following Lemmas.

Lemma 1. *Given a* MEL *theory* $(\Sigma, E \cup A)$ *and its associated rewrite theory* $\mathcal{R}_E$*, if* $\mathcal{R}_E$ *is closed under* A*-extensions then* $\rightarrow^1_{E,A}$ *is strictly coherent, i.e., for all* t_1, t_2, t_3 *if* $t_1 \rightarrow^1_{E,A} t_2$ *and* $t_1 =_A t_3$ *then there exists* t_4 *such that* $t_3 \rightarrow^1_{E,A} t_4$ *and* $t_2 =_A t_4$ *(see Fig. 2).*

Lemma 2. *Given a rewrite theory* $\mathcal{R} = (\Sigma, E \cup A, R)$*, if* R *is closed under* A*-extensions then* $\rightarrow^1_{R,A}$ *is strictly coherent, i.e., for all* t_1, t_2, t_3 *if* $t_1 \rightarrow^1_{R,A} t_2$ *and* $t_1 =_A t_3$ *then there exists* t_4 *such that* $t_3 \rightarrow^1_{R,A} t_4$ *and* $t_2 =_A t_4$ *(see Fig. 2).*

$$\begin{tikzcd}
t_1 \arrow[r, "1", "E,A"'] \arrow[d, equal, "A"'] & t_2 \arrow[d, dotted, dash, "A"] \\
t_3 \arrow[r, dotted, "1", "E,A"'] & t_4
\end{tikzcd}
\qquad
\begin{tikzcd}
t_1 \arrow[r, "1", "R,A"'] \arrow[d, equal, "A"'] & t_2 \arrow[d, dotted, dash, "A"] \\
t_3 \arrow[r, dotted, "1", "R,A"'] & t_4
\end{tikzcd}$$

Fig. 2. strict coherence of $\rightarrow^1_{E,A}$ and $\rightarrow^1_{R,A}$

Strict coherence of $\rightarrow^1_{E,A}$ and $\rightarrow^1_{R,A}$ will be used later in the paper to prove the equivalence of $\rightarrow_{R/\mathcal{E}}$ and $\rightarrow_{R \cup E,A}$ for narrowable rewrite theories. Given a MEL theory $(\Sigma, E \cup A)$ and its associated rewrite theory $\mathcal{R}_E = (\Sigma', A, R_E)$, if $\mathcal{R}_E$ is closed under A-extensions, and $\rightarrow^1_{E,A}$ is sort-decreasing, terminating, and confluent, then for each term $t \in T_\Sigma(\mathcal{X})$ there exists a unique (up to A-equivalence and new variable renaming) term $t{\downarrow}_{E,A}$. When the MEL theory is also *admissible* (defined below) then $t{\downarrow}_{E,A}$ is unique up to A-equivalence. When $\mathcal{E}$ is understood from the context we use the simpler notation $t{\downarrow}$.

3.3 Admissible theories. Executable MEL theory. Narrowable rewrite theory

In this subsection we will first define the MEL theories that we can deal with by rewriting. As our aim is to shrink the state space by computing canonical terms with Maude's metalevel `metaReduce` function, before each narrowing step, the most general MEL theories that we are able to admit will be the admissible MEL theories, the ones that Maude can deal with. Then we will define the concept of narrowable rewrite theory, which has the following assumptions relaxed with respect to executable rewrite theories [CDE$^+$07]: a narrowable rewrite theory doesn't need to be operationally terminating, it admits extra variables anywhere in the conditions, and it has no restrictions on equational, membership or rewrite conditions. Only matching equations have restrictions.

Definition 8 (Σ-pattern). *Given a* MEL *theory* $(\Sigma, E \cup A)$ *we call a term* $t \in T_\Sigma(\mathcal{X})$ *a* Σ-pattern *if* $t \notin \mathcal{X}$*, and for any* E, A*-normalized substitution* σ *with* $Dom(\sigma) \subseteq Var(t) \neq \emptyset$ *if* $x \in Dom(\sigma)$ *then* $t\sigma$ *is* E, A*-irreducible.*

A sufficient condition for t to be a Σ-pattern is the absence of A-unifiers between nonvariable subterms of t and lefthand sides of equations in E.

Definition 9 (Admissible MEL Theory). *A* MEL *theory* $(\Sigma, \mathcal{E})$ *is* admissible *[CDE$^+$07] if:*

- *For each conditional equation* $t = t'$ *if* $\bigwedge_{i=1}^n A_i$ *in* $\mathcal{E}$ *the following requirements are satisfied:*

1.

$$Var(t') \subseteq Var(t) \cup \bigcup_{j=1}^{n} Var(A_j).$$

2. If A_i is an equation $u_i = v_i$ or a membership $u_i : s_i$, then

$$Var(A_i) \subseteq Var(t) \cup \bigcup_{j=1}^{i-1} Var(A_j).$$

3. If A_i is a matching equation $u_i := v_i$, then u_i is a Σ-pattern and

$$Var(v_i) \subseteq Var(t) \cup \bigcup_{j=1}^{i-1} Var(A_j).$$

- *For each conditional membership $t : s$ if $\bigwedge_{i=1}^{n} A_i$ in $\mathcal{E}$ conditions 2 and 3 above are satisfied.*

We want to apply narrowing only to canonical terms, reducing the state space of our narrowing problems. Matching with canonical forms may not be safe in general. The use of *FPP* theories will ensure the completeness of this procedure [CEM14].

Definition 10 (FPP Mel theory). *A* Mel *theory $(\Sigma, \mathcal{E})$ has the* Fresh Pattern Property (FPP) *if for each sentence $t = t'$ if $\bigwedge_{i=1}^{n} A_i$ or $t : s$ if $\bigwedge_{i=1}^{n} A_i$ in $\mathcal{E}$, if A_i has the form $u_i := v_i$ then $(Var(t) \cup \bigcup_{j=1}^{i-1} Var(A_j)) \cap Var(u_i) = \emptyset$.*

A matching equation in an FPP Mel theory is similar to a "let" expression in functional programming, allowing us to define locally some value that is needed later in the condition, or in the right part of a conditional equation. For narrowing purposes the restriction that we put on matching equations will allow us to instantiate the extra variables in u_i (we call them *matching variables*) by A-unification of u_i with the canonical form of some instance of v_i, instead of performing a needless unification by $\mathcal{E}$-narrowing. The main difference with respect to "let" expressions is that this matching is done modulo the axioms A, so we gain expressiveness.

Example 5. Let $(\Sigma, E \cup A)$ be a Mel theory, with sorts *item*(i), *multiset*(m), and *state*(s); subsorts $i \leq m$; constants $a :\to i$, $b :\to i$, and *empty* $:\to i$; functions $\_;\_ \ : \ m \ m \to m$ (with axioms associative, commutative, and identity *empty*), and $[\_] : \ m \to s$.

If $E = \{[x_m] = [y_m] \ \mathit{if} \ a; y_m := x_m\}$ then $(\Sigma, E \cup A)$ is FPP because the equation applies to states, not to multisets, so $a; y_m$ is a Σ-pattern, x_m appears in the left side of the equation, and y_m is a new variable.

What this equation does is to remove any appearance of the constant a in the multiset included within a state, just by matching the multiset with the Σ-pattern (modulo the axioms of the constructor for multisets $\_;\_$), leaving a state holding a multiset whose elements are the remaining b's, or the empty multiset if there were none.

Example 6. Consider the Mel theory $(\Sigma, E \cup A)$:

$$K = \{k\}, S = \{s\}, S_k = \{s\}, \Omega = \{\{a, b, c, d\}_s, \{f, [\_,\_]\}_{ss,s}\},$$

with $A = \emptyset$ and equations:

$$E = \{a = b, \quad c = d, \quad f(x, y) = z \ \mathit{if} \ [x, z] := [x, y]\}$$

Its associated rewrite theory has rules:

$$R_E = \{a \to b, \quad c \to d, \quad f(x,y) \to z \ \mathit{if}\ [x,y] \to [x,z]\}$$

E is admissible; $\to^1_{E,A}$ is confluent, terminating, and sort-decreasing. We have omitted the sort subindex in the variables. Rewriting the term $f(a,c)$ in $\to^1_{E,A}$ generates the condition $[a,c] \to [a,z]$. If we match $[a,c]$ with $[a,z]$ before rewriting $[a,c]$ in $\to^1_{E,A}$ we get the match $z \mapsto c$, so $f(a,c) \to c$. However, if we rewrite $[a,c]$ to its canonical form $[b,d]$ we get the condition $[b,d] \to [a,z]$ that does not match, so $f(a,c)$ cannot be rewritten.

Using FPP theories we can rewrite any term to normal form before matching. An easy transformation allows us to turn any rewrite or Mel theory into an FPP one. We demonstrate it using the previous example.

Example 7. The transformed FPP Mel theory $(\Sigma, E \cup A)$ has equations:

$$E = \{a = b, \quad c = d, \quad f(x,y) = z \ \mathit{if}\ [x',z] := [x,y] \wedge x = x'\}$$

where we have added a new variable x', and a new condition $x = x'$ that forces both variables, x and x', to be instantiated to E, A-equivalent terms. Now, the associated rewrite theory is:

$$R_E = \{a \to b, \quad c \to d, \quad f(x,y) \to z \ \mathit{if}\ [x,y] \to [x',z] \wedge eq_{k,k}(x,x') \to tt\}$$

E is admissible; $\to^1_{E,A}$ is confluent, terminating, and sort-decreasing. Rewriting the term $f(a,c)$ generates the condition $[a,c] \to [x',z] \wedge eq_{k,k}(a,x') \to tt$ now. Using rules $a \to b$ and $c \to d$ we get $[b,d] \to [x',z] \wedge eq_{k,k}(a,x') \to tt$, where $[b,d]$ is a normal form. Substitution $\sigma = \{x' \mapsto b, z \mapsto d\}$ solves the first part of the condition, and the second part of the condition becomes $eq_{k,k}(a,b) \to tt$ which, using the rule $a \to b$, rewrites to $eq_{k,k}(b,b) \to tt$ and, using rule $eq_{k,k}(x_k,x_k) \to tt$ with substitution $x_k \mapsto b$, rewrites to $tt \to tt$, that holds by reflexivity. Then $f(a,c)$ rewrites to d, which is the normal form of c, so the rewritings in both examples are E, A-equivalent.

Definition 11 (Admissible rewrite theory). *A rewrite theory $\mathcal{R} = (\Sigma, \mathcal{E}, R)$ is* admissible *if:*

1. *The* Mel *theory $(\Sigma, \mathcal{E})$ is admissible.*
2. *For each rule $l \to r$ if $\bigwedge_{i=1}^n A_i$ in R:*
   - *$Var(r) \subseteq Var(l) \cup \bigcup_{i=1}^n Var(A_i)$.*
   - *If A_i has the form $u_i := v_i$, then u_i is a Σ-pattern.*

Definition 12 (FPP rewrite theory). *A rewrite theory $\mathcal{R} = (\Sigma, \mathcal{E}, R)$ has the Fresh Pattern Property if the* Mel *theory $(\Sigma, \mathcal{E})$ is FPP, and for each rule $l \to r$ if $\bigwedge_{i=1}^n A_i$ in R, if A_i has the form $u_i := v_i$ then $(Var(l) \cup \bigcup_{j=1}^{i-1} Var(A_j)) \cap Var(u_i) = \emptyset$.*

Note that for the rules in an FPP rewrite theory we admit extra variables anywhere in their conditions, even on the right side of matching equations, but not in the right term r, and again we demand that for matching equations $u_i := v_i$ the variables in u_i haven't appeared before in the rule. We can relax these requirements because we only need the rewrite theories to be *narrowable*, while we need the Mel theories to be *executable*, so we can get the canonical form of any term by E, A-rewriting.

Definition 13 (Narrowable rewrite theory). *An admissible rewrite theory $\mathcal{R} = (\Sigma, E \cup A, R)$ is* narrowable *if Σ is preregular modulo A; E, A, and R are finite; no left term in E and R is a variable; and $\mathcal{R}$ satisfies the following requirements:*

1. *$\mathcal{R}$ is FPP, and both $\mathcal{R}$ and the associated rewrite theory $\mathcal{R}_E = (\Sigma', A, R_E)$ are closed under A-extensions.*
2. *The axioms in A are regular, linear, and sort-preserving. Any variable appearing in an axiom must be a kinded variable. Furthermore, equality modulo A must be decidable and there must exist a finitary matching algorithm modulo A producing a finite number of A-matching substitutions, $Match_A(t_1, t_2) = \{\sigma_i\}_{i=1}^n$ meaning that $t_1 =_A t_2\sigma_i$ for $i = 1, \ldots, n$, or failing otherwise.*
3. *The relation $\rightarrow^1_{E,A}$ is sort-decreasing, terminating, confluent, and operationally terminating.*
4. *$\rightarrow^1_{R,A}$ is $\mathcal{E}$-*coherent *with $\rightarrow^1_{E,A}$ (see Fig. 3), i.e., for all t_1, t_2, t_3 we have $t_1 \rightarrow^1_{R,A} t_2$ and $t_1 \rightarrow^*_{E,A} t_3$ implies that there exist t_4, t_5 such that $t_3 \rightarrow^*_{E,A} t_4$, $t_4 \rightarrow^1_{R,A} t_5$, and $t_2 =_{\mathcal{E}} t_5$. We represent this property by using a diagram with filled lines for universal quantification and dotted lines for existential quantification:*

$$\begin{tikzcd}
t_1 \arrow[rr, "1", "{R,A}"'] \arrow[d, "{E,A}"', "*"] & & t_2 \arrow[d, dotted, dash, "\mathcal{E}"] \\
t_3 \arrow[r, dotted, "*", "{E,A}"'] & t_4 \arrow[r, dotted, "1", "{R,A}"'] & t_5
\end{tikzcd}$$

Fig. 3. $\mathcal{E}$-coherence of $\rightarrow^1_{R,A}$ with $\rightarrow^1_{E,A}$

Example 8. Consider a rewrite theory $\mathcal{R} = (\Sigma, E \cup A, R)$, where $S = \{s\}$, $\Omega = \{\{a, b, c\}_s, \{f\}_{s,s}\}$, with $A = \emptyset$, $E = \{a = b\}$ and $R = \{f(a) \rightarrow c\}$.

In this theory $f(a) \rightarrow^1_{R,A} c$, but $f(a) \rightarrow^1_{E,A} f(b)$ and $f(b)$ cannot be further rewritten in $\rightarrow^1_{R \cup E,A}$, so the theory is not $\mathcal{E}$-coherent. If we add the rule $f(b) \rightarrow c$ to R then $f(a) \rightarrow^1_{E,A} f(b) \rightarrow^1_{R,A} c$, and we have an $\mathcal{E}$-coherent rewrite theory.

Regarding Maude specifications, operational termination of the rewrite theory associated to a Mel theory can be checked using the Maude Termination Tool [DLM08b] and $\mathcal{E}$-coherence of $\rightarrow^1_{R,A}$ with $\rightarrow^1_{E,A}$ can be checked using the Maude Coherence Checker Tool [DM12b]. These tools together with the Church-Rosser Checker (which can be used to check the Churck-Rosser property of equational theories), the Maude Sufficient Completeness Checker (which can be used to check that defined functions have been fully defined in terms of constructors), and the Maude Inductive Theorem Prover (which can be used to verify inductive properties of equational theories), conform the Maude Formal Environment [CDH+07].

Definition 14. *The* Mel *theory associated to a narrowable rewrite theory is an* executable Mel *theory [CDE+07].*

For narrowable rewrite theories we can implement $\rightarrow_{R/\mathcal{E}}$ using $\rightarrow_{R \cup E,A}$. This lemma links $\rightarrow^1_{R/\mathcal{E}}$ with $\rightarrow^1_{E,A}$ and $\rightarrow^1_{R,A}$.

Lemma 3 (Reduction of $\rightarrow^1_{R/\mathcal{E}}$ to $\rightarrow_{R \cup E,A}$ for Narrowable Rewrite Theories). *Let $\mathcal{R} = (\Sigma, \mathcal{E}, R)$ be a narrowable rewrite theory. Then $t_1 \rightarrow^1_{R/\mathcal{E}} t_2$ if and only if $t_1 \rightarrow^!_{E,A} t_1{\downarrow} \rightarrow^1_{R,A} t_3$ for some $t_3 =_{\mathcal{E}} t_2$. This can be verified by checking $t_3{\downarrow} =_A t_2{\downarrow}$.*

$$\begin{tikzcd}
t_1 \arrow[r, equal, "\mathcal{E}"'] \arrow[d, "E{,}A"', "!"] & t \arrow[r, "1", "R"'] \arrow[d, "E{,}A"', "!"] & t' \arrow[r, equal, "\mathcal{E}"'] \arrow[d, equal, dotted, "\mathcal{E}"] & t_2 \arrow[dd, "E{,}A"', "!"] \\
t_1{\downarrow} \arrow[r, equal, "A"'] \arrow[dr, dotted, "1", "R{,}A"'] & t{\downarrow} \arrow[r, dotted, "1", "R{,}A"'] & t_4 \arrow[d, "E{,}A"', "!"] & \\
 & t_3 \arrow[ur, equal, dotted, "\mathcal{E}"'] \arrow[dr, "!", "E{,}A"'] & t_4{\downarrow} \arrow[r, equal, "A"'] \arrow[d, equal, "A"] & t_2{\downarrow} \\
 & & t_3{\downarrow} &
\end{tikzcd}$$

Fig. 4. Reduction of $\rightarrow^{1}_{R/\mathcal{E}}$ to $\rightarrow_{R\cup E,A}$

Proof. The if part is immediate just by noticing that $t_1 =_{\mathcal{E}} t_1{\downarrow}$, so $t_1{\downarrow} \rightarrow^{1}_{R,A} t_3$ implies that there is some t' such that $t_1{\downarrow} =_A t'$ and $t' \rightarrow^{1}_{R} t_3$. Then $t_1 =_{\mathcal{E}} t' \rightarrow^{1}_{R} t_3$ and, as a consequence, for any t_2 such that $t_2 =_{\mathcal{E}} t_3$ we get $t_1 \rightarrow^{1}_{R/\mathcal{E}} t_2$.

The only if part follows from the diagram in Fig. 4: the upper line of the diagram represents our assumption on $t_1 \rightarrow^{1}_{R/\mathcal{E}} t_2$; the upper left square follows from convergence and termination of E modulo A; the upper right square follows from $\mathcal{E}$-coherence of $\rightarrow^{1}_{R,A}$, since $\rightarrow^{1}_{R}\subseteq\rightarrow^{1}_{R,A}$; finally, the upper inverted triangle is a distorted version of the square that represents the strict coherence of $\rightarrow^{1}_{R,A}$. The lower triangle and the right rectangle use the same property applied in the upper left square. As a conclusion we see that $t_1{\downarrow} \rightarrow^{1}_{R,A} t_3$ for some t_3 and $t_3{\downarrow} =_A t_2{\downarrow}$, which is decidable, since the number of rules in R_E is finite, A-matching is decidable and finite, and $\rightarrow^{1}_{E,A}$ is operationally terminating. □

Theorem 1 (Equivalence of $\rightarrow_{R/\mathcal{E}}$ and $\rightarrow_{R\cup E,A}$ for Narrowable Rewrite Theories). *Let $\mathcal{R} = (\Sigma, \mathcal{E}, R)$ be a narrowable rewrite theory. Then $t \rightarrow_{R/\mathcal{E}} t'$ if and only if $t \rightarrow_{R\cup E,A} t'$.*

Proof. The if part is immediate because $\rightarrow_{R\cup E,A}\subseteq\rightarrow_{R/\mathcal{E}}$ by definition.

We prove the only if part by induction on the number of $\rightarrow^{1}_{R/\mathcal{E}}$ steps.

- Base case: zero rewrite steps. Then $t =_{\mathcal{E}} t'$, so $t \rightarrow_{R\cup E,A} t'$ also with zero rewrite steps.
- Induction case: $t \rightarrow^{1}_{R/\mathcal{E}} v \rightarrow_{R/\mathcal{E}} t'$. By Lemma 3, $t \rightarrow^{*}_{R\cup E,A} u =_{\mathcal{E}} v$. Then also $u \rightarrow_{R/\mathcal{E}} t'$, and by induction hypothesis $u \rightarrow_{R\cup E,A} t'$, so $t \rightarrow_{R\cup E,A} t'$.

□

Example 9. The rewrite theory for the concurrency specification example is narrowable if we decompose $\mathcal{E}$ in the following way: the set A contains the associative, commutative, and identity equations in $\mathcal{E}$; the set E contains the rest of equations and all memberships in $\mathcal{E}$.

3.4 Unification goal. Reachability goal

Definition 15 (Unification Goal). *A system of sentences F in $(\Sigma, \mathcal{E})$ of the form*

$$\bigwedge_{i=1}^{n} u_i = u'_i \wedge \bigwedge_{j=1}^{m} v_j := v'_j \wedge \bigwedge_{k=1}^{l} t_k : s_k,$$

has an associated unification *goal G in $\mathcal{R}_E$ of the form*

$$\bigwedge_{i=1}^{n} eq(u_i, u'_i) \to tt \wedge \bigwedge_{j=1}^{m} v'_j \to v_j \wedge \bigwedge_{k=1}^{l} t_k{:}s_k \to tt,$$

where v_j is a Σ-pattern, for $1 \leq j \leq m$. A substitution σ is an $\mathcal{E}$-solution for G if $eq(u_i\sigma, u'_i\sigma) \to_{E,A} tt$ $(1 \leq i \leq n)$, $v'_j\sigma \to_{E,A} v_j\sigma$ $(1 \leq j \leq m)$, and $t_k\sigma{:}s_k \to_{E,A} tt$ $(1 \leq k \leq l)$.

Recall that a unification goal associated to a system of sentences has the same form and restrictions as the conditions of the rules in R_E.

Definition 16 (Reachability Goal). *Given a rewrite theory $\mathcal{R} = (\Sigma, \mathcal{E}, R)$, a* reachability goal *$G$ is a conjunction of the form $u_1 \Rightarrow v_1 \wedge \ldots \wedge u_n \Rightarrow v_n \wedge G'$ where for $1 \leq i \leq n$, $u_i, v_i \in T_\Sigma(\mathcal{X})_{\kappa_i}$ for appropriate κ_i, and G' is a unification goal in $\mathcal{R}_E$ associated to a system of sentences F in the* Mel *theory $(\Sigma, \mathcal{E})$. The subgoals $u_i \Rightarrow v_i$ can be interleaved with the subgoals in G'.*

We define $Var(G) = \bigcup_{i=1}^{n}(Var(u_i) \cup Var(v_i)) \cup Var(G')$. A substitution σ is a *solution* of G if σ is an $\mathcal{E}$-solution for G', and $u_i\sigma \to_{R/\mathcal{E}} v_i\sigma$, for $1 \leq i \leq n$. If the substitution is idempotent we also say that the solution is *idempotent.* We define $\mathrm{E}(G)$ to be the system of sentences $u_1 = v_1 \wedge \ldots \wedge u_n = v_n \wedge F$. We say σ is a *trivial solution* of G if it is an $\mathcal{E}$-solution for $\mathrm{E}(G)$. We say G is trivial if the identity substitution *id* is a trivial solution of G.

Theorem 2 (Equivalence of $\mathcal{E}$-solutions for systems of sentences and unification goals). *Given a* Mel *theory $(\Sigma, E \cup A)$ and its associated rewrite theory $\mathcal{R}_E$, if $\mathcal{R}_E$ is terminating, convergent, sort-decreasing, and closed under A-extensions, then for any system of sentences F, an idempotent E,A-normalized substitution σ is an $\mathcal{E}$-solution for F iff σ is an $\mathcal{E}$-solution for its associated unification goal (G) in $\mathcal{R}_E$.*

Proof. First we prove that if σ is an $\mathcal{E}$-solution of F then σ is an $\mathcal{E}$-solution for G. We prove it by induction on the number of deduction rules for Mel theories applied. We consider each possible type of sentence in F.

1. $t = t'$, and $t\sigma =_{\mathcal{E}} t'\sigma$, so $(t\sigma){\downarrow} =_A (t'\sigma){\downarrow}$. In G we have the subgoal $eq(t, t') \to tt$, and $eq(t\sigma, t'\sigma) \to^{*}_{E,A} eq((t\sigma){\downarrow}, (t'\sigma){\downarrow}) \to^{1}_{E,A} tt =_A tt$, with rule $eq(x_k, x_k) \to tt$, and substitution $\{x_k \mapsto (t\sigma){\downarrow}\}$ so, by definition $eq(t\sigma, t'\sigma) \to_{E,A} tt$. Then σ is an $\mathcal{E}$-solution for $eq(t, t') \to tt$.
2. $t := t'$, and $t\sigma =_{\mathcal{E}} t'\sigma$, so $(t\sigma){\downarrow} =_A (t'\sigma){\downarrow}$. As t is a Σ-pattern and σ is E,A-normalized then $(t\sigma){\downarrow} \equiv t\sigma$ so $t\sigma =_A (t'\sigma){\downarrow}$. In G we have the subgoal $t' \to t$, and $t'\sigma \to^{*}_{E,A} (t'\sigma){\downarrow} =_A t\sigma$. Then, by definition, $t'\sigma \to_{E,A} t\sigma$ so σ is an $\mathcal{E}$-solution for $t' \to t$ in $\mathcal{R}_E$.
3. $t : s$, and $t\sigma :_{\mathcal{E}} s$. We consider two subcases, depending on the deduction rule applied.
   - Rule replacement. Then we infer $t\sigma :_{\mathcal{E}} s$ because there is a sentence $l : s$ *if* $\bigwedge_{i=1}^{n} A_i$ in E, and a substitution ρ such that $t\sigma \equiv l\rho$, and ρ is an $\mathcal{E}$-solution for A_i, $1 \leq i \leq n$. In G we have the subgoal $t{:}s \to tt$, and in R_E we have the rule $l{:}s \to tt$ *if* $\bigwedge_{i=1}^{n} A'_i$ where, by induction hypothesis, ρ is a solution for A'_i, $1 \leq i \leq n$, in $\mathcal{R}_E$, so $l\rho{:}s \to^{1}_{E,A} tt$, that is, $t\sigma{:}s \to^{1}_{E,A} tt =_A tt$ so, by definition, $t\sigma{:}s \to_{E,A} tt$ so σ is an $\mathcal{E}$-solution for $t{:}s \to_{E,A} tt$.
   - Rule membership. Then we infer $t\sigma :_{\mathcal{E}} s$ because $t\sigma =_{\mathcal{E}} t'$ and $t' :_{\mathcal{E}} s$ is deduced with rule replacement. The case where several rules membership are applied before applying rule replacement is easily reduced to this one using rule transitivity: if $t\sigma =_{\mathcal{E}} t_1 =_{\mathcal{E}} \ldots =_{\mathcal{E}} t'$, then $t\sigma =_{\mathcal{E}} t'$, so $(t\sigma){\downarrow} =_A t'{\downarrow}$. There is a sentence $l : s$ *if* $\bigwedge_{i=1}^{n} A_i$ in E, and a substitution ρ such that $t' \equiv l\rho$, and ρ is an $\mathcal{E}$-solution for A_i, $1 \leq i \leq n$. In G we have the subgoal $t{:}s \to tt$, and in R_E we have the rule $l{:}s \to tt$ *if* $\bigwedge_{i=1}^{n} A'_i$ where, by induction hypothesis, ρ is a solution for A'_i, $1 \leq i \leq n$, in $\mathcal{R}_E$, so $l\rho{:}s \to^{1}_{E,A} tt$, that is, $t'{:}s \to^{1}_{E,A} tt$. As

$t' \to^{*}_{E,A} t'{\downarrow}$, we also can apply the same rules to t' in the context $t'{:}s$, so $t'{:}s \to^{*}_{E,A} t'{\downarrow}{:}s$. As tt is a canonical form and $t'{:}s \to^{1}_{E,A} tt$ then, by confluence, $t'{\downarrow}{:}s \to^{*}_{E,A} tt$. But, as $(t\sigma){\downarrow}{:}s =_A t'{\downarrow}{:}s$ (because $(t\sigma){\downarrow} =_A t'{\downarrow}$) then, by strict coherence of $\to^{1}_{E,A}$, $t'{\downarrow}{:}s \to^{*}_{E,A} tt$ implies $(t\sigma){\downarrow}{:}s \to^{*}_{E,A} tt$. As $t\sigma{:}s \to^{*}_{E,A} (t\sigma){\downarrow}{:}s$, we conclude that $t\sigma{:}s \to^{*}_{E,A} tt =_A tt$ so, by definition, $t\sigma{:}s \to_{E,A} tt$ so σ is an $\mathcal{E}$-solution for $t{:}s \to_{E,A} tt$.

Now we prove that if σ is an $\mathcal{E}$-solution for G then σ is an $\mathcal{E}$-solution of F. We prove it by induction on the total number of rewrite steps applied. We consider each possible type of subgoal in G.

1. $eq_k(t, t') \to tt$. Then $eq_k(t\sigma, t'\sigma) \to_{E,A} tt$.
   - One rewrite step: then $eq_k(t\sigma, t'\sigma) \to_{E,A} tt$ with rule $eq_k(x_k, x_k) \to tt$ because $t\sigma =_A t'\sigma$, so $t\sigma =_{\mathcal{E}} t'\sigma$ because $A \subseteq \mathcal{E}$. The sentence in F is $t = t'$, and $t\sigma =_{\mathcal{E}} t'\sigma$, so σ is an $\mathcal{E}$-solution for $t = t'$.
   - $n > 1$ rewrite steps: without loss of generality we assume that the rewritten term in the first rewrite step is t. Then $eq_k(t\sigma, t'\sigma) \to^{1}_{E,A} eq_k((t\sigma)[r\rho]_p, t'\sigma) \to_{E,A} tt$ with rule $l \to r \ \mathit{if}\ c'$ in R_E, because $(t\sigma)|_p =_A l\rho$ (so $(t\sigma)|_p =_{\mathcal{E}} l\rho$) and ρ is an $\mathcal{E}$-solution for all the conditions in c'. Then there must be a corresponding equation $l = r \ \mathit{if}\ c$ in E (the only rules that don't have a counterpart are those related to the new sort *Truth*, and no subterm of t and t' can have a sort *Truth* or kind [*Truth*]) where, by induction hypothesis, ρ is an $\mathcal{E}$-solution for all the conditions in c. By replacement rule, we can deduce $l\rho =_{\mathcal{E}} r\rho$. Then, by repeated application of the congruence rule, we can deduce $(t\sigma)[l\rho]_p =_{\mathcal{E}} (t\sigma)[r\rho]_p$. As $(t\sigma)|_p =_{\mathcal{E}} l\rho$ we can also deduce $t\sigma =_{\mathcal{E}} (t\sigma)[l\rho]_p$ by repeated application of the congruence rule. Then, using the transitivity rule, we can deduce $t\sigma =_{\mathcal{E}} (t\sigma)[r\rho]_p$. As σ is idempotent, r has fresh variables, and $\mathit{Ran}(\rho)$ is away from $\mathit{Var}((t\sigma)[r])$, then $(t\sigma)[r\rho]_p\sigma = (t\sigma)[r\rho]_p$, and $eq_k((t\sigma)[r\rho]_p\sigma, t'\sigma) \to_{E,A} tt$ with less than n rewrite steps so, by induction hypothesis, σ is an $\mathcal{E}$-solution for the sentence $(t\sigma)[r\rho]_p = t'$, and then $t\sigma =_{\mathcal{E}} t'\sigma$.
2. $t{:}s \to tt$. Then $t\sigma{:}s \to_{E,A} tt$.
   - One rewrite step: then $t\sigma{:}s \to_{E,A} tt$ with rule $l{:}s \to tt$ in R_E because there is a substitution ρ such that $t\sigma =_A l\rho$. The sentence in F is $t : s$, and there is a sentence $l : s$ in E and $t\sigma =_A l\rho$, so $t\sigma :_{\mathcal{E}} s$ and σ is an $\mathcal{E}$-solution for $t : s$.
   - $n > 1$ rewrite steps: then $t\sigma{:}s \to^{1}_{E,A} (t\sigma)[r\rho]_p{:}s \to_{E,A} tt$ with rule $l \to r \ \mathit{if}\ c'$ in R_E, because $(t\sigma)|_p =_A l\rho$ (so $(t\sigma)|_p =_{\mathcal{E}} l\rho$) and ρ is an $\mathcal{E}$-solution for all the conditions in c'. Then there must be a corresponding equation $l = r \ \mathit{if}\ c$ in E where, by induction hypothesis, ρ is an $\mathcal{E}$-solution for all the conditions in c. By replacement rule, we can deduce $l\rho =_{\mathcal{E}} r\rho$. Then, by repeated application of the congruence rule, we can deduce $(t\sigma)[l\rho]_p =_{\mathcal{E}} (t\sigma)[r\rho]_p$. As $(t\sigma)|_p =_{\mathcal{E}} l\rho$ we can also deduce $t\sigma =_{\mathcal{E}} (t\sigma)[l\rho]_p$ by repeated application of the congruence rule. Then, using the transitivity rule, we can deduce $t\sigma =_{\mathcal{E}} (t\sigma)[r\rho]_p$. As σ is idempotent, r has fresh variables, and $\mathit{Ran}(\rho)$ is away from $\mathit{Var}((t\sigma)[r])$, then $(t\sigma)[r\rho]_p\sigma = (t\sigma)[r\rho]_p$, and $(t\sigma)[r\rho]_p\sigma{:}s \to_{E,A} tt$ with less than n rewrite steps so, by induction hypothesis, σ is an $\mathcal{E}$-solution for the sentence $(t\sigma)[r\rho]_p : s$, and then by rule membership $t\sigma :_{\mathcal{E}} s$.
3. $t' \to t$, with $t \neq tt$. Then $t'\sigma \to_{E,A} t\sigma$.
   - Zero rewrite steps: then $t\sigma =_A t'\sigma$, so $t\sigma =_{\mathcal{E}} t'\sigma$ because $A \subseteq \mathcal{E}$. The sentence in F is $t := t'$, and $t\sigma =_{\mathcal{E}} t'\sigma$, so σ is an $\mathcal{E}$-solution for $t := t'$.
   - $n > 0$ rewrite steps: then $t\sigma \to^{1}_{E,A} (t\sigma)[r\rho]_p \to_{E,A} t'\sigma$ with rule $l \to r \ \mathit{if}\ c'$ in R_E, because $(t\sigma)|_p =_A l\rho$ (so $(t\sigma)|_p =_{\mathcal{E}} l\rho$) and ρ is an $\mathcal{E}$-solution for all the conditions in c'. Then there must be a corresponding equation $l = r \ \mathit{if}\ c$ in E where, by induction hypothesis, ρ is an $\mathcal{E}$-solution for all the conditions in c. By replacement rule, we can deduce $l\rho =_{\mathcal{E}} r\rho$. Then, by repeated application of the congruence rule, we can deduce $(t\sigma)[l\rho]_p =_{\mathcal{E}} (t\sigma)[r\rho]_p$. As

$(t\sigma)|_p =_{\mathcal{E}} l\rho$ we can also deduce $t\sigma =_{\mathcal{E}} (t\sigma)[l\rho]_p$ by repeated application of the congruence rule. Then, using the transitivity rule, we can deduce $t\sigma =_{\mathcal{E}} (t\sigma)[r\rho]_p$. As σ is idempotent, r has fresh variables, and $Ran(\rho)$ is away from $Var((t\sigma)[r])$, then $(t\sigma)[r\rho]_p\sigma = (t\sigma)[r\rho]_p$, and $(t\sigma)[r\rho]_p\sigma \rightarrow_{E,A} t'\sigma$ with less than n rewrite steps so, by induction hypothesis, σ is an $\mathcal{E}$-solution for the sentence $(t\sigma)[r\rho]_p = t'$, and then $t\sigma =_{\mathcal{E}} t'\sigma$, i.e., σ is an $\mathcal{E}$-solution for $t := t'$.

□

As a conclusion, we can verify that σ is an $\mathcal{E}$-solution for F by checking $G\sigma$ using the relation $\rightarrow_{E,A}$. Conversely, if we find an $\mathcal{E}$-solution σ for G, σ is an $\mathcal{E}$-solution for F.

3.5 Narrowing

Definition 17 ($R \cup E, A$-narrowing). *Given a rewrite theory $\mathcal{R} = (\Sigma, \mathcal{E}, R)$, its associated rewrite theory $\mathcal{R}_E$, a term t in $T_\Sigma(\mathcal{X})$, and a rule $c \equiv l \rightarrow r$ if C in $R \cup R_E$, properly renamed so $Var(c) \cap Var(t) = \emptyset$, if there exists a non-variable position p in $Pos_\Sigma(t)$, and a substitution σ such that $t|_p\sigma =_A l\sigma$, and $C\sigma$ holds, then we write $t \rightsquigarrow^1_{p,\sigma,R\cup E,A} t[r]_p\sigma$ and say that there is a* $R \cup E, A$-narrowing step *from t to $t[r]_p\sigma$.*

E, A-narrowing in $\mathcal{R}_E$ is defined similarly. In conditional narrowing we usually start with a unifier $\sigma' \in CSU_A(t|_p = l)$ and recursively solve the new goal $C\sigma'$ using narrowing, obtaining some σ'' as solution. Then $\sigma = \sigma'\sigma''$ is the desired substitution such that $t \rightsquigarrow^1_{p,\sigma,R\cup E,A} t[r]_p\sigma$.

Example 10. Consider $\mathcal{R} = (\Sigma, \mathcal{E}, R)$, where $S = \{s\}$, $\Omega = \{\{a,b,c\}_s, \{f,g\}_{ss,s}\}$, a rewrite theory with $E = A = \emptyset$, and $R = \{g(b,c) \rightarrow c, f(a,z_s) \rightarrow b \textit{ if } g(b,z_s) \rightarrow c\}$.

Now, if we try to narrow the term $f(x_s,y_s)$ with rule $f(a,z_s) \rightarrow b$ *if* $g(b,z_s) \rightarrow c$ and unifier $\sigma' = \{x_s \mapsto a, y_s \mapsto w_s, z_s \mapsto w_s\}$ we have to prove the condition $g(b,w_s) \rightarrow c$, which can be narrowed with rule $g(b,c) \rightarrow c$ and substitution $\sigma'' = \{w_s \mapsto c\}$, so $g(b,z_s) \rightsquigarrow_{\sigma'',R\cup E,A} c$. Then, by composition of the substitutions σ' and σ'', we get $\sigma = \{x_s \mapsto a, y_s \mapsto c, z_s \mapsto c\}$ and we have $f(x_s,y_s) \rightsquigarrow_{\sigma,R\cup E,A} b$. As a consequence, that will be later proved, $f(x_s,y_s)\sigma \rightarrow_{R\cup E,A} b$.

4 Sentence-normalized rewriting

Let $(\Sigma, \mathcal{E})$ be an FPP executable MEL theory, and $\mathcal{R}_E = (\Sigma', A, R_E)$ its associated rewrite theory. Executability allows us to incrementally construct the substitutions used on $\mathcal{R}_E$ in such a way that we will only generate E, A-normalized substitutions for matching variables. Let $t \in T_\Sigma$ be a term, and $c' \equiv l \rightarrow r$ *if* $\bigwedge_{i=1}^n A'_i$ a conditional rule in R_E (we don't have to prove anything for unconditional rules). If l matches t using σ_0 ($t =_A l\sigma_0$) then for all i, $1 \leq i \leq n$, if A'_i has no matching variables we define $\sigma_i = id$; else if $A'_i \equiv t'_i \rightarrow t_i$ has matching variables (because the corresponding sentence c in E has the condition $A_i \equiv t_i := t'_i$), then $(\Sigma, \mathcal{E})$ being FPP implies that each substitution σ_j, $1 \leq j < i$ instantiates different variables, so $\bigcup_{j=0}^{i-1} \sigma_j$ is properly defined, and $Dom(\bigcup_{j=0}^{i-1} \sigma_j) \cap Var(t_i) = \emptyset$. Then $(t'_i \bigcup_{j=0}^{i-1} \sigma_j \rightarrow t_i \bigcup_{j=0}^{i-1} \sigma_j) \equiv (t'_i \bigcup_{j=0}^{i-1} \sigma_j \rightarrow t_i)$. We define σ_i to be a matching of t_i with $(t'_i \bigcup_{j=0}^{i-1} \sigma_j)\!\downarrow$, that is $t_i\sigma_i =_A (t'_i \bigcup_{j=0}^{i-1} \sigma_j)\!\downarrow$. As we are matching against an E, A-irreducible term, σ_i must be E, A-normalized. The only exception is the first substitution σ_0, which may be not E, A-normalized but doesn't instantiate matching variables. The *extended* substitution σ that we need to apply the rule is $\sigma = \bigcup_{i=0}^n \sigma_i$, where the instantiation of all matching variables, $\bigcup_{i=1}^n \sigma_i$, is E, A-normalized.

Based on this fact, we develop in this section the concepts of *sentence-normalized substitution* and *sentence-normalized rewriting* and show their connection to E,A-rewriting and $R \cup E,A$-rewriting for our unification and reachability goals. As we have already shown the link between $=_{\mathcal{E}}$, $:_{\mathcal{E}}$, and E,A-rewriting, and also the link between $R/\mathcal{E}$-rewriting and $R \cup E,A$-rewriting, all the properties for the narrowing calculi to be presented in the next sections will only have to be related to sentence-normalized rewriting.

Definition 18 (Sentence-normalized Substitution). *Given a narrowable rewrite theory $\mathcal{R} = (\Sigma, \mathcal{E}, R)$, its FPP executable* Mel *theory $(\Sigma, E \cup A)$, and the associated rewrite theory $\mathcal{R}_E = (\Sigma', A, R_E)$, for any conditional rule, $c \equiv l \to r$ if C in R_E or R and substitution σ, the* sentence-normalized substitution *σ_c is defined as $\sigma_c = \sigma|_{Var(l)} \cup \sigma{\downarrow}|_{Extra(c)}$, where $Extra(c) = Var(c) \setminus Var(l)$ is the set of new extra variables in c. In the case of rules in R_E, $Extra(c)$ will only contain matching variables.*

Proposition 1. *Given an FPP executable* Mel *theory $(\Sigma, \mathcal{E})$, its associated rewrite theory $\mathcal{R}_E = (\Sigma', A, R_E)$, and a term $t \in T_{\Sigma'}(\mathcal{X})$, if $t \to^1_{E,A} t[r\sigma]_p$ using a rule $c \equiv l \to r$ if $\bigwedge_{i=1}^n A'_i$ in R_E and an idempotent substitution σ, then $t \to^1_{E,A} t[r\sigma_c]_p$ using the same rule c and substitution σ_c.*

Proof. For unconditional rules there is nothing to prove, because $\sigma_c \equiv \sigma$, so we focus on conditional rules. By definition of σ_c we have that as $t =_A l\sigma$ then $t =_A l\sigma_c$. Now we prove that for $i = 1, \ldots, n$ if the condition $A'_i\sigma$ is verified then the condition $A'_i\sigma_c$ is also verified. We prove it for the three types of conditions:

1. case $eq(t_i\sigma, t'_i\sigma) \to_{E,A} tt$. Then as $eq(t_i\sigma, t'_i\sigma) \to^*_{E,A} eq(t_i\sigma_c, t'_i\sigma_c)$, by confluence of $\to^1_{E,A}$, as tt is a unique normal form, we get $eq(t_i\sigma_c, t'_i\sigma_c) \to_{E,A} tt$.
2. case $t_i\sigma{:}s \to_{E,A} tt$. We use the equivalence between $\to_{E,A}$ and $:_{\mathcal{E}}$. Then $t_i\sigma :_{\mathcal{E}} s$. As $t_i\sigma \to^*_{E,A} t_i\sigma_c$ and $\to^1_{E,A}$ is sort decreasing then $t_i\sigma_c$ has sort s, so $t_i\sigma_c :_{\mathcal{E}} s$ must be derivable. By the equivalence between $\to_{E,A}$ and $:_{\mathcal{E}}$, $t_i\sigma_c{:}s \to_{E,A} tt$
3. case $t'_i\sigma \to_{E,A} t_i\sigma$ (from $t_i := t'_i$). Then $t'_i\sigma \to^*_{E,A} u =_A t_i\sigma$. As for all rules $c \equiv l \to r$ *if* C if $l\rho =_A u$ we also have that $l\rho =_A t_i\sigma$, then the rewrite steps in $\to^1_{E,A}$ are the same for u and $t_i\sigma$. As $t_i\sigma \to^*_{E,A} t_i\sigma_c$, we have $t'_i\sigma \to^*_{E,A} t_i\sigma_c$. We also have $t'_i\sigma \to^*_{E,A} t'_i\sigma_c$. But $t_i\sigma_c$ is a normal form because t_i is a Σ-pattern, all the variables in t_i are matching variables in c and σ_c is E,A-normalized with respect to all matching variables in c. Then, by confluence, it must be the case that $t'_i\sigma_c \to^*_{E,A} t' =_A t_i\sigma_c$, so $t'_i\sigma_c \to_{E,A} t_i\sigma_c$.

□

Proposition 2. *Given a narrowable rewrite theory $\mathcal{R} = (\Sigma, \mathcal{E}, R)$, its FPP executable* Mel *theory $(\Sigma, E \cup A)$, the associated rewrite theory $\mathcal{R}_E = (\Sigma', A, R_E)$, and two terms t, t' in $T_{\Sigma'}(\mathcal{X})$, if $t \to_{R \cup E,A} t'$ then $t{\downarrow} \to_{R \cup E,A} t'$.*

Proof. By induction on the number of $\to^1_{R \cup E,A}$ steps. Remember that $=_A \subseteq =_{\mathcal{E}}$.

Base case. Zero $\to^1_{R \cup E,A}$ steps. Then $t =_{\mathcal{E}} t'$, so $[t]_{\mathcal{E}} = [t']_{\mathcal{E}}$. As $[t]_{\mathcal{E}} = [t{\downarrow}]_{\mathcal{E}}$ then $[t{\downarrow}]_{\mathcal{E}} = [t']_{\mathcal{E}}$, so $t{\downarrow} =_{\mathcal{E}} w$.

Induction case. We consider two cases depending on the first rule used.

- $t \to^1_{E,A} u \to_{R \cup E,A} t'$. By induction hypothesis $u{\downarrow} \to^*_{R \cup E,A} w =_{\mathcal{E}} t'$, but $u{\downarrow} =_A t{\downarrow}$ by confluence of $\to^1_{E,A}$. Then, by strict coherence of $\to^1_{E,A}$ and $\to^1_{R,A}$, $t{\downarrow} \to^*_{R \cup E,A} w' =_A w$, so $t{\downarrow} \to_{R \cup E,A} t'$.

– $t \rightarrow^1_{R,A} u \rightarrow_{R\cup E,A} t'$. By induction hypothesis $u{\downarrow} \rightarrow_{R\cup E,A} t'$. As $t \rightarrow^!_{E,A} t{\downarrow}$, then by $\mathcal{E}$-coherence of $\rightarrow^1_{R,A}$ with $\rightarrow^1_{E,A}$, $t{\downarrow} \rightarrow^1_{R,A} u' =_{\mathcal{E}} u$. By confluence of $\rightarrow^1_{E,A}$, $u' \rightarrow^!_{E,A} u'{\downarrow} =_A u{\downarrow}$ so, by strict coherence of $\rightarrow^1_{E,A}$ and $\rightarrow^1_{R,A}$, $u'{\downarrow} \rightarrow_{R\cup E,A} w =_A t'$. Putting all together, and by definition of $\rightarrow_{R\cup E,A}$, $t{\downarrow} \rightarrow^1_{R,A} u' \rightarrow^!_{E,A} u'{\downarrow} \rightarrow^*_{R\cup E,A} w' =_{\mathcal{E}} w =_A t'$, so $t{\downarrow} \rightarrow_{R\cup E,A} t'$.

□

Proposition 3. *Given a narrowable rewrite theory $\mathcal{R} = (\Sigma, \mathcal{E}, R)$, its FPP executable* Mel *theory $(\Sigma, E \cup A)$, the associated rewrite theory $\mathcal{R}_E = (\Sigma', A, R_E)$, and a term $t \in T_{\Sigma'}(\mathcal{X})$, if $t \rightarrow^1_{R,A} t[r\sigma]_p$ using a rule $c \equiv l \rightarrow r$ if $\bigwedge_{i=1}^n A_i$ in R and an idempotent substitution σ, then $t \rightarrow^1_{R,A} t[r\sigma_c]_p$ using the same rule c and substitution σ_c.*

Proof. For unconditional rules again there is nothing to prove, because $\sigma_c \equiv \sigma$, so we focus on conditional rules. By definition of σ_c we have that as $t =_A l\sigma$ then $t =_A l\sigma_c$. Now we prove that for $i = 1, \ldots, n$ if the condition $A_i\sigma$ is verified then the condition $A_i\sigma_c$ is also verified. We have already proved it for the three types of equational conditions in Proposition 1 using the associated condition A'_i, so the only case left to prove is the one where $A_i \equiv t_i \rightarrow t'_i$ and $t_i\sigma \rightarrow_{R\cup E,A} t'_i\sigma$. We prove it by induction in the number of $\rightarrow^1_{R\cup E,A}$ steps, including those due to all rewriting conditions in the rewriting path.

- Base case. Zero $\rightarrow^1_{R\cup E,A}$ steps. Then $t_i\sigma =_{\mathcal{E}} t'_i\sigma$, and as $t_i\sigma \rightarrow_{E,A} t_i\sigma_c$, and $t'_i\sigma \rightarrow_{E,A} t'_i\sigma_c$ then $t_i\sigma_c =_{\mathcal{E}} t'_i\sigma_c$.
- Induction case. We consider two cases depending on the first rule used.
  - $t_i\sigma \rightarrow^1_{E,A} u \rightarrow_{R\cup E,A} t'_i\sigma$. As σ is idempotent then $t_i\sigma \rightarrow^1_{E,A} u\sigma \rightarrow_{R\cup E,A} t'_i\sigma$. As in case 3 of Proposition 1, $t_i\sigma_c \rightarrow_{E,A} t'\sigma_c$, and by induction hypothesis $u\sigma_c \rightarrow_{R\cup E,A} t'_i\sigma_c$, so $t_i\sigma_c \rightarrow_{R\cup E,A} t'_i\sigma_c$.
  - $t_i\sigma \rightarrow^1_{R,A} u \rightarrow_{R\cup E,A} t'_i\sigma$. As σ is idempotent then $t_i\sigma \rightarrow^1_{R,A} u\sigma \rightarrow_{R\cup E,A} t'_i\sigma$. As $t_i\sigma \rightarrow^*_{E,A} t_i\sigma_c$ then, by $\mathcal{E}$-coherence of $\rightarrow^1_{R,A}$ with $\rightarrow^1_{E,A}$, $t_i\sigma_c \rightarrow^*_{E,A}\rightarrow^1_{R,A} w =_{\mathcal{E}} u\sigma$. $u\sigma \rightarrow^*_{E,A} u\sigma_c \rightarrow^!_{E,A} (u\sigma){\downarrow}$ and, by induction hypothesis, $u\sigma_c \rightarrow_{R\cup E,A} t'_i\sigma_c$. Then, by Proposition 2 $(u\sigma){\downarrow} \rightarrow^*_{R\cup E,A} w' =_{\mathcal{E}} w$. As $w =_{\mathcal{E}} u\sigma$ then $w \rightarrow^!_{E,A} w{\downarrow} =_A (u\sigma){\downarrow}$. Then, by strict coherence of $\rightarrow^1_{E,A}$ and $\rightarrow^1_{R,A}$, $w{\downarrow} \rightarrow^*_{R\cup E,A} w'' =_A w'$. Putting all together, $t_i\sigma_c \rightarrow^*_{E,A}\rightarrow^1_{R,A} w \rightarrow^!_{E,A} w{\downarrow} \rightarrow^*_{R\cup E,A} w'' =_A w' =_{\mathcal{E}} t'_i\sigma_c$, so $t_i\sigma_c \rightarrow_{R\cup E,A} t'_i\sigma_c$.

□

We are interested in computing E, A-normalized solutions for unification and reachability goals using only sentence-normalized substitutions, hence reducing the state space.

Definition 19 (Sentence-normalized Rewriting). *We will use the term* sentence-normalized rewriting (SNR) *and write $t \rightarrow^1_N t'$ (or $t \rightarrow_N t'$) instead of $t \rightarrow^1_{E,A} t'$ (resp., $t \rightarrow_{E,A} t'$), and also write $t \Rightarrow^1_N t'$ (or $t \Rightarrow_N t'$) instead of $t \rightarrow^1_{R\cup E,A} t'$ (resp., $t \rightarrow_{R\cup E,A} t'$), to imply that only sentence-normalized substitutions have been applied in all rewrite steps.*

Note that $\Rightarrow_N$ is related to $\rightarrow_{R\cup E,A}$, $\rightarrow^1_N \subseteq \Rightarrow^1_N$, and $\rightarrow_N \subseteq \Rightarrow_N$. Also note that $\rightarrow^1_N \subseteq \rightarrow^1_{E,A}$, so $\rightarrow_N$ is sound with respect to $\rightarrow_{E,A}$, and $\Rightarrow^1_N \subseteq \rightarrow^1_{R\cup E,A}$, so $\Rightarrow_N$ is sound with respect to $\rightarrow_{R\cup E,A}$. Now we prove that $\rightarrow_N$ is complete with respect to $\rightarrow_{E,A}$ when rewriting to canonical form.

Lemma 4 (Completeness of Sentence-normalized Rewriting to Canonical Form). *Given an FPP executable* Mel *theory $(\Sigma, \mathcal{E})$ and its associated rewrite theory $\mathcal{R}_E = (\Sigma', A, R_E)$, if $t \rightarrow_{E,A} t'$, with $t, t' \in T_{\Sigma'}(\mathcal{X})$ and t' is E, A-irreducible, then $t \rightarrow_N t'$.*

Proof. By induction on the total number of $\rightarrow^1_{E,A}$ steps.

- Base case: $t \rightarrow_{E,A} t'$ with zero $\rightarrow^1_{E,A}$ steps. Then $t' =_A t$, so $t' \rightarrow_N t$.
- Induction case: $t \rightarrow^1_{E,A} t[r\sigma]_p \rightarrow_{E,A} t'$ with a rule $c \equiv l \rightarrow r$ *if* $\bigwedge_{i=1}^n u_i \rightarrow v_i$ and a substitution σ. By Proposition 1 $t \rightarrow^1_{E,A} t[r\sigma_c]_p$ with rule c and substitution σ_c, so $u_i\sigma_c \rightarrow_{E,A} v_i\sigma_c$ for $1 \leq i \leq n$. For $1 \leq i \leq n$ the term v_i in rule c must be a Σ-pattern (maybe with form tt), and σ_c is E, A-normalized with respect to $Var(v_i)$, so $v_i\sigma_c$ is E, A-irreducible. Then, by induction hypothesis, $u_i\sigma_c \rightarrow_N v_i\sigma_c$ for $1 \leq i \leq n$, so $t \rightarrow^1_N t[r\sigma_c]_p$ by definition. We choose the derivation $t \rightarrow^1_N t[r\sigma_c]_p \rightarrow_{E,A} t'$ which must exist by confluence of $\rightarrow^1_{E,A}$ because $t[r\sigma]_p \rightarrow^*_{E,A} t[r\sigma_c]_p$, t' is E, A-irreducible, and $\rightarrow^1_N \subseteq \rightarrow^1_{E,A}$. By induction hypothesis $t[r\sigma_c]_p \rightarrow_N t'$, so $t \rightarrow_N t'$. □

As a consequence, it is always the case that $t \rightarrow^!_N t{\downarrow}$ and also, as $\rightarrow^1_N \subseteq \Rightarrow^1_N$, $t \Rightarrow^*_N t{\downarrow}$.

Proposition 4. *Given a narrowable rewrite theory $\mathcal{R} = (\Sigma, \mathcal{E}, R)$, its FPP executable* Mel *theory $(\Sigma, E \cup A)$, the associated rewrite theory $\mathcal{R}_E = (\Sigma', A, R_E)$, and two terms t, t' in $T_{\Sigma'}(\mathcal{X})$, if $t \Rightarrow_N t'$ then $t{\downarrow} \Rightarrow_N t'$.*

Proof. By induction on the number of $\Rightarrow^1_N$ steps. Remember that $=_A \subseteq =_\mathcal{E}$.

Base case. Zero $\Rightarrow^1_N$ steps. Then $t =_\mathcal{E} t'$, so $[t]_\mathcal{E} = [t']_\mathcal{E}$. As $[t]_\mathcal{E} = [t{\downarrow}]_\mathcal{E}$ then $[t{\downarrow}]_\mathcal{E} = [t']_\mathcal{E}$, so $t{\downarrow} =_\mathcal{E} t'$.

Induction case. $t \Rightarrow^1_N u \Rightarrow_N t'$. By Lemma 4 $u \Rightarrow_N u{\downarrow}$. By induction hypothesis $u{\downarrow} \Rightarrow_N t'$. Putting all together: $t \Rightarrow^1_N u \Rightarrow^*_N u{\downarrow} \Rightarrow_N t'$. □

Lemma 5 (Completeness of Sentence-normalized Rewriting for $\rightarrow_{R/\mathcal{E}}$). *Given a narrowable rewrite theory $\mathcal{R} = (\Sigma, \mathcal{E}, R)$, its FPP executable* Mel *theory $(\Sigma, E \cup A)$, the associated rewrite theory $\mathcal{R}_E = (\Sigma', A, R_E)$, and terms t, t' in $T_{\Sigma'}(\mathcal{X})$, if $t \rightarrow_{R/\mathcal{E}} t'$ then $t \Rightarrow_N t'$.*

Proof. By Theorem 1, as $t \rightarrow_{R/\mathcal{E}} t'$ then $t \rightarrow_{R \cup E,A} t'$. We prove the Lemma by induction on the number of $\rightarrow^1_{R \cup E,A}$ steps.

Base case. Zero $\rightarrow^1_{R \cup E,A}$ steps. Trivial because $t =_\mathcal{E} t'$.

Induction case. $t \rightarrow^1_{R \cup E,A} t[r\sigma]_p \rightarrow_{R \cup E,A} t'$, with some rule $c \equiv l \rightarrow r$ *if* C in R_E or R and substitution σ at position p. By Proposition 1 and Proposition 3, and as $\rightarrow^1_N \subseteq \Rightarrow^1_N$, then $t \Rightarrow^1_N t[r\sigma_c]_p$. As $t[r\sigma]_p \rightarrow^*_{E,A} t[r\sigma_c]_p \rightarrow^!_{E,A} t[r\sigma]_p{\downarrow}$ then, by Lemma 4, $t[r\sigma_c]_p \Rightarrow^*_N t[r\sigma]_p{\downarrow}$. By induction hypothesis $t[r\sigma]_p \Rightarrow_N t'$ so, by Proposition 4 $t[r\sigma]_p{\downarrow} \Rightarrow_N t'$. Putting all together: $t \Rightarrow^1_N t[r\sigma_c]_p \Rightarrow^*_N t[r\sigma]_p{\downarrow} \Rightarrow_N t'$. □

As a direct consequence of Lemma 4 we get the following theorem telling us that with respect to associated unification goals the E, A-normalized E, A-solutions are the same using $\rightarrow_{E,A}$, or $\rightarrow_N$ (which we call *N-solutions*). Recall that $\sigma_c \equiv \sigma$ for any E, A-normalized substitution σ and rule c in R_E.

Theorem 3 (Equivalence of SNR for Solutions of Associated Unification Goals). *Given an FPP executable* Mel *theory $(\Sigma, \mathcal{E})$ and its associated rewrite theory $\mathcal{R}_E = (\Sigma', A, R_E)$, an E, A-normalized substitution σ is an $\mathcal{E}$-solution of a system of sentences F and an E, A-solution of its associated unification goal $G \equiv \bigwedge_{i=1}^n (t_i \rightarrow t'_i)$ (so $t_i\sigma \rightarrow_{E,A} t'_i\sigma$, for $1 \leq i \leq n$) iff $t_i\sigma \rightarrow_N t'_i\sigma$, $i = 1, \ldots, n$ (i.e. σ is an N-solution for G).*

Proof. We prove each part of the double implication separately.

- $\Rightarrow$: for $1 \leq i \leq n$ the term t'_i is a Σ-pattern (maybe with form tt) by definition of system of sentences and associated unification goal, so $t'_i\sigma$ is E, A-irreducible. As $t_i\sigma \rightarrow_{E,A} t'_i\sigma$ then, by Lemma 4, $t_i\sigma \rightarrow_N t'_i\sigma$.
- $\Leftarrow$: Immediate since $\rightarrow^1_N \subseteq \rightarrow^1_{E,A}$. □

Now we prove that conditions and canonical conditions have the same E, A-normalized solutions. This result is important because it will allow us to reduce the state space in our narrowing problems by working only with canonical forms.

Proposition 5. *Given an FPP executable* MEL *theory* $(\Sigma, \mathcal{E})$ *and its associated rewrite theory* $\mathcal{R}_E = (\Sigma', A, R_E)$*, for any conditional* MEL *sentence c in E, and corresponding rule* $c' \equiv s'$ *if* $\bigwedge_{i=1}^{n} u_i \to v_i$ *in* $\mathcal{R}_E$*, if there is an E, A-normalized substitution σ such that* $u_i\sigma \to_{E,A} v_i\sigma$*, for* $1 \leq i \leq n$*, then* $u_i{\downarrow}\sigma \to_N v_i\sigma$*, for* $1 \leq i \leq n$.

Proof. Immediate since FPP and executability imply that, for $1 \leq i \leq n$, v_i is a Σ-pattern and σ is E, A-normalized, so $v_i\sigma$ is E, A-irreducible, and $u_i\sigma \to_{E,A} v_i\sigma$. $u_i \to_{E,A} u_i{\downarrow}$ implies $u_i\sigma \to_{E,A} u_i{\downarrow}\sigma$ so, by confluence of $\to^1_{E,A}$, $u_i{\downarrow}\sigma \to_{E,A} v_i\sigma$. Then by Lemma 4, as $v_i\sigma$ is E, A-irreducible, $u_i{\downarrow}\sigma \to_N v_i\sigma$. □

Now we have, as a direct consequence, the desired result.

Lemma 6 (Equivalence of Solutions for Canonical Unification Goals). *Given an FPP executable* MEL *theory* $(\Sigma, \mathcal{E})$ *and its associated rewrite theory* $\mathcal{R}_E = (\Sigma', A, R_E)$*, an E, A-normalized substitution σ is an E, A-solution of the unification goal* $G \equiv \bigwedge_{i=1}^{n}(t_i \to t'_i)$*, associated to a system of sentences F, iff* $t_i{\downarrow}\sigma \to_N t'_i\sigma$*, for* $1 \leq i \leq n$ *(i.e., σ is an N-solution for $G{\downarrow}$).*

Proof. We have written $t'_i\sigma$ instead of $t'_i{\downarrow}\sigma$ because for unification goals associated to a system of sentences it is always the case that $t'_i{\downarrow} \equiv t'_i$ (t'_i must be tt or a Σ-pattern). We prove each part of the double implication separately.

- $\Rightarrow$: $t_i\sigma \to_{E,A} t'_i\sigma$. Since a unification goal associated to a system of sentences has the same form and restrictions as the conditions of the rules in R_E, then, by Proposition 5, $t_i{\downarrow}\sigma \to_N t'_i\sigma$.
- $\Leftarrow$: $t_i{\downarrow}\sigma \to_N t'_i\sigma$. As $\to^1_N \subseteq \to^1_{E,A}$ then $t_i{\downarrow}\sigma \to_{E,A} t'_i\sigma$. $t_i \to^*_{E,A} t_i{\downarrow}$ implies $t_i\sigma \to^*_{E,A} t_i{\downarrow}\sigma$. As a consequence of the last two deductions $t_i\sigma \to_{E,A} t'_i\sigma$. □

We can also prove that with respect to reachability goals the E, A-normalized E, A-solutions are the same using $\to_{R \cup E,A}$, or $\Rightarrow_N$ (which we again call *N-solutions*).

Lemma 7 (Equivalence of Solutions for Canonical Reachability Goals). *Given a narrowable rewrite theory* $\mathcal{R} = (\Sigma, \mathcal{E}, R)$*, its FPP executable* MEL *theory* $(\Sigma, E \cup A)$*, the associated rewrite theory* $\mathcal{R}_E = (\Sigma', A, R_E)$*, and a reachability goal* $G \equiv u_1 \Rightarrow v_1 \wedge \ldots \wedge u_n \Rightarrow v_n \wedge G'$*, where G' is a unification goal associated to a system of sentences F, and an idempotent E, A-normalized substitution σ, these three assertions are equivalent:*

1. *σ is a solution for G,*
2. *σ is an N-solution for $G{\downarrow}$,*
3. *σ is a solution for $G{\downarrow}$.*

Proof. We prove $1 \Rightarrow 2$, $2 \Rightarrow 3$, and $3 \Rightarrow 1$.

- $1 \Rightarrow 2$
  By induction on the number of $\Rightarrow^1_{R/\mathcal{E}}$ steps.
  Base case: zero $\Rightarrow^1_{R/\mathcal{E}}$ steps. Then, by Definition 16, σ is a trivial solution of G, so $u_i\sigma =_{\mathcal{E}} v_i\sigma$, for $1 \leq i \leq n$, and σ is an $\mathcal{E}$-solution for G'. Then, by Lemma 6, σ is also a solution for $u_i{\downarrow} =_{\mathcal{E}} v_i{\downarrow}$, for $1 \leq i \leq n$, and also for $G'{\downarrow}$, i.e., σ is a trivial solution for $G{\downarrow}$.
  Induction case: without losing generality we assume that $u_1\sigma \Rightarrow^1_{R/\mathcal{E}} t \Rightarrow_{R/\mathcal{E}} v_1\sigma$. As σ is idempotent, then $u_1\sigma \Rightarrow^1_{R/\mathcal{E}} t\sigma \Rightarrow_{R/\mathcal{E}} v_1\sigma$. By definition of $=_{\mathcal{E}}$ we have that $u_1 =_{\mathcal{E}} u_1{\downarrow}$ and

$t =_{\mathcal{E}} t{\downarrow}$, so $u_1\sigma =_{\mathcal{E}} u_1{\downarrow}\sigma$ and $t\sigma =_{\mathcal{E}} t{\downarrow}\sigma$. Then, by definition of $\Rightarrow^1_{R/\mathcal{E}}$, $u_1{\downarrow}\sigma \Rightarrow^1_{R/\mathcal{E}} t{\downarrow}\sigma$ and, by Lemma 5, $u_1{\downarrow}\sigma \Rightarrow^1_N t{\downarrow}\sigma$. By induction hypothesis (from now on we write *I.H.*) $t{\downarrow}\sigma \Rightarrow_N v_1{\downarrow}\sigma$, so $u_1{\downarrow}\sigma \Rightarrow_N v_1{\downarrow}\sigma$. Also by I.H. $u_i{\downarrow}\sigma \Rightarrow_N v_i{\downarrow}\sigma$, for $2 \leq i \leq n$, and σ is an N-solution for $G'{\downarrow}$, so σ is an N-solution for $G{\downarrow}$.

- $2 \Rightarrow 3$

  Trivial because $\Rightarrow^1_N \subseteq \Rightarrow^1_{R\cup E,A} \subseteq \Rightarrow^1_{R/\mathcal{E}}$ and σ is an N-solution for $G{\downarrow}$.
- $3 \Rightarrow 1$

  Again by induction on the number of $\Rightarrow^1_{R/\mathcal{E}}$ steps.

  Base case: zero $\Rightarrow^1_{R/\mathcal{E}}$ steps. Then, by Definition 16, σ is a trivial solution of $G{\downarrow}$, so $u_i{\downarrow}\sigma =_{\mathcal{E}} v_i{\downarrow}\sigma$, for $1 \leq i \leq n$, and σ is an $\mathcal{E}$-solution for $G'{\downarrow}$. Then, by Lemma 6, σ is also a solution for $u_i =_{\mathcal{E}} v_i$, for $1 \leq i \leq n$, and also for G', i.e., σ is a trivial solution for G.

  Induction case: without losing generality we assume that $u_1{\downarrow}\sigma \Rightarrow^1_{R/\mathcal{E}} t \Rightarrow_{R/\mathcal{E}} v_1{\downarrow}\sigma$. As $t =_{\mathcal{E}} t{\downarrow}$ then $u_1{\downarrow}\sigma \Rightarrow^1_{R/\mathcal{E}} t{\downarrow} \Rightarrow_{R/\mathcal{E}} v_1{\downarrow}\sigma$ by definition of $\Rightarrow^1_{R/\mathcal{E}}$. As σ is idempotent, then $u_1{\downarrow}\sigma \Rightarrow^1_{R/\mathcal{E}} t{\downarrow}\sigma \Rightarrow_{R/\mathcal{E}} v_1{\downarrow}\sigma$. By definition of $=_{\mathcal{E}}$ we have that $u_1 =_{\mathcal{E}} u_1{\downarrow}$ and $t =_{\mathcal{E}} t{\downarrow}$, so $u_1\sigma =_{\mathcal{E}} u_1{\downarrow}\sigma$ and $t\sigma =_{\mathcal{E}} t{\downarrow}\sigma$. Then, by definition of $\Rightarrow^1_{R/\mathcal{E}}$, $u_1\sigma \Rightarrow^1_{R/\mathcal{E}} t\sigma$. By I.H. $t\sigma \Rightarrow_{R/\mathcal{E}} v_1\sigma$, so $u_1\sigma \Rightarrow_{R/\mathcal{E}} v_1\sigma$. Also by I.H. $u_i\sigma \Rightarrow_{R/\mathcal{E}} v_i\sigma$, for $2 \leq i \leq n$, and σ is a solution for G', so σ is a solution for G.

□

5 Conditional Narrowing for $\mathcal{E}$-solutions

Narrowing allows us to compute solutions for reachability goals. We implement narrowing using a calculus with the following properties:

1. The calculus is *weakly complete*, i.e., for any idempotent $R \cup E, A$-normalized solution of a reachability goal G, the calculus can compute a more general answer for G.
2. The calculus is *sound*, i.e., if the calculus computes an answer σ for a reachability goal G, then σ is a solution of G.

We are going to split the calculus into two parts: the one that solves unification goals and the one that solves reachability goals. We assume that we have an A-unification algorithm that for any equation $t = t'$ returns $CSU_A(t = t')$ away from all the variables in G, so all the unifiers are idempotent.

5.1 Transformation for unification with memberships

As the current existing unification algorithms in Maude are only valid for order-sorted theories, we are going to develop a transformation that allows us to apply these algorithms to our Mel theories at the kind level, and later takes into account membership information provided by the variables in the calculus. As this transformation can impose a lot of extra work to our calculus, because it doesn't make use of order-sorted information for computing A-unifiers, it would be desirable to identify which terms or subterms are not suitable for order-sorted unification and apply the transformation only on those terms or subterms. We show an algorithm that identifies the sorts that cannot be involved in an order-sorted unification. We will apply the transformation only to terms of those sorts.

From S, the set of sorts in our rewrite theory, we define the subset $MB(S)$ of non order-sorted unifiable sorts, see [LM09], as the smallest subset of S such that

1. if $t : s(\ \mathit{if}\ \ c)$ in E is not a subsort declaration then $s \in \mathit{MB}(S)$.
2. if $s \in \mathit{MB}(S)$ and $s \leq s'$, with s, s' in S then $s' \in \mathit{MB}(S)$.
3. if $f : s_1 \cdots s_n \to s$ is an operator declaration, with s in S and $s_i \in \mathit{MB}(S)$ for some i, $1 \leq i \leq n$, then $s \in \mathit{MB}(S)$.

Recall that for simplicity we only allow overloading of operators when their images belong to the same kind. We define $\mathit{OS}(S) = S - \mathit{MB}(S)$. $\mathit{OS}(S)$ is the set of sorts whose terms can be unified by order sorted unification, no memberships can be involved directly or indirectly, via operators, when checking whether a term has any of these sorts or not.

Example 11. In the concurrency specification example, as $\mathtt{u1}, \mathtt{u2} : \mathtt{us}$ is in E, then $\mathtt{us} \in \mathit{MB}(S)$ using case 1. Also, as $x_{\mathtt{us}} \mid y_{\mathtt{u}}, y'_{\mathtt{u}} \mid \mathtt{emptyT} \mid \mathtt{emptyT} : s\ \mathit{if}\ y_{\mathtt{u}}, y'_{\mathtt{u}}, x_{\mathtt{us}} : \mathtt{us}$ is in E, then $\mathtt{s} \in \mathit{MB}(S)$ using case 1. Then $\mathit{MB}(S) = \{\mathtt{us}, \mathtt{s}\}$ and $\mathit{OS}(S) = \{\mathtt{u}, \mathtt{t}, \mathtt{tb}, \mathtt{n}, \mathtt{b}\}$.

Now, given t and t', if $\mathit{ls}(t) \in \mathit{OS}(S)$ and $\mathit{ls}(t') \in \mathit{OS}(S)$, we unify them directly. Else we compute a non well formed substitution $\rho = \{x^i_{s_i} \mapsto y^i_{[s_i]} \mid x^i_{s_i} \in \mathit{Var}(t) \cup \mathit{Var}(t') \wedge s_i \in S\}$, with each $y^i_{[s_i]}$ a fresh unsorted variable, and unify $s\rho$ and $t\rho$, terms that only have unsorted variables. From $\rho = \{x^i_{s_i} \mapsto y^i_{[s_i]}\}^n_{i=1}$ we generate the system of sentences $C = \bigwedge^n_{i=1} y^i_{[s_i]} : s_i$. C and ρ are computed by $\mathit{kinded}(t, t')$ defined below, V is the auxiliary set of already processed variables, function k processes lists of terms, function $k1$ processes individual terms and function $k0$ discards V and returns the pair (C, ρ):

$\mathit{kinded}(t, t') = (\emptyset, \mathit{id})$ if $\mathit{ls}(t) \in \mathit{OS}(S)$ and $\mathit{ls}(t') \in \mathit{OS}(S)$
$\mathit{kinded}(t, t') = k0(k((t, t'), (\emptyset, \mathit{id}, \emptyset)))$ otherwise

$k((t_1, \ldots, t_n), (C, \rho, V)) = k((t_2, \ldots, t_n), k1(t_1, (C, \rho, V)))$
$k((t), (C, \rho, V)) = k1(t, (C, \rho, V))$

$k1(f(t_1, \ldots, t_n), (C, \rho, V)) = k((t_1, \ldots, t_n), (C, \rho, V))$
$k1(c, (C, \rho, V)) = (C, \rho, V)$ where c constant.
$k1(x^i_{\kappa_i}, (C, \rho, V)) = (C, \rho, V)$ if $x^i_{\kappa_i} \in V$ or κ is a kind.
$k1(x^i_{\kappa_i}, (C, \rho, V)) = (C \wedge y^i_{[s_i]} : s_i, \rho \cup \{x^i_{\kappa_i} \mapsto y^i_{[s_i]}\}, V \cup \{x^i_{\kappa_i}\})$ otherwise, with $y^i_{[s_i]}$ a fresh variable.

$k0(C, \rho, V)) = (C, \rho)$

The computed substitution ρ replaces the variables that belong to sorts that cannot be unified with order-sorted algorithms with variables of the corresponding kind. The computed condition C ensures that the new kinded variables are instantiated to terms with the same sort as the original variables.

Lemma 8. *Given an FPP executable* Mel *theory* $(\Sigma, E \cup A)$ *and its associated rewrite theory* $\mathcal{R}_E = (\Sigma', A, R_E)$*, a substitution* σ*, with* $\mathit{Dom}(\sigma) = \mathit{Var}(t) \cup \mathit{Var}(t')$ *(i.e., all variables are at least renamed), is an idempotent A-unifier of two terms* t, t' *in* $T_\Sigma(\mathcal{X})$ *if and only if* $\sigma =_A \rho\gamma\gamma'$*, with* $\mathit{kinded}(t, t') = (C, \rho)$*,* γ *an idempotent A-unifier of* $t\rho$ *and* $t'\rho$*, and* γ' *an* $\mathcal{E}$*-solution for the system of sentences* $C\gamma$*, where all substitutions are always away from all the variables that have previously appeared.*

Proof. We prove each implication separately.

$\Rightarrow)$

The proof for the case where $\mathit{ls}(t) \in \mathit{OS}(S)$ and $\mathit{ls}(t') \in \mathit{OS}(S)$ is trivial because no memberships are involved in the unification of t and t', C is empty, $\rho = \mathit{id}$, $\gamma = \sigma$ and $\gamma' = \mathit{id}$.

Otherwise, if σ is an idempotent A-unifier of t and t', with $\mathit{Dom}(\sigma) = \{x^1_{s_1}, \ldots, x^n_{s_n}, z^1_{k_1}, \ldots, z^m_{k_m}\}$ such that s_i in S $(1 \leq i \leq n)$ and k_j in K $(1 \leq j \leq m)$, then, by construction, for each $x^i_{s_i}$, with $1 \leq i \leq n$, there is a fresh variable $y^i_{[s_i]}$ such that $\rho = \{x^i_{s_i} \mapsto y^i_{[s_i]}\}^n_{i=1}$ and $C = \bigwedge^n_{i=1} y^i_{[s_i]} : s_i$.

We define $\sigma' = \{y^i_{[s_i]} \mapsto x^i_{s_i}\sigma\}^n_{i=1} \cup \{z^j_{k_j} \mapsto z^j_{k_j}\sigma\}^m_{j=1}$. By construction $\sigma = \rho\sigma'$. σ' is idempotent because each $y^i_{[s_i]}$, $1 \leq i \leq n$, is a fresh variable that doesn't appear anywhere else and σ is idempotent. As $t\sigma =_A t'\sigma$ then $(t\rho)\sigma' =_A (t'\rho)\sigma'$, so σ' is an A-unifier for $(t\rho)$ and $(t'\rho)$. Then, there is a substitution $\gamma \in \mathit{CSU}_A(t\rho = t'\rho)$, so $t\rho\gamma =_A t'\rho\gamma$, such that $\sigma' \ll_A \gamma$, and there exists a substitution γ' such that $\sigma' =_A \gamma\gamma'$, so $\sigma =_A \rho\gamma\gamma'$. For each condition $y^i_{[s_i]} : s_i$ in C, $1 \leq i \leq n$, we have that $x^i_{s_i}\rho = y^i_{[s_i]}$, and $x^i_{s_i}\rho\gamma\gamma' =_A x^i_{s_i}\sigma$, so $y^i_{[s_i]}\gamma\gamma' =_A x^i_{s_i}\sigma$. As $x^i_{s_i}\sigma$ has sort s_i because σ is well-formed, then γ' is a solution for $y^i_{[s_i]}\gamma : s_i$, so γ' is an $\mathcal{E}$-solution for the system of sentences $C\gamma$.

$\Leftarrow$)

If $\rho = \{x^i_{s_i} \mapsto y^i_{[s_i]}\}^n_{i=1}$, $C = \bigwedge^n_{i=1} y^i_{[s_i]} : s_i$, γ is an idempotent A-unifier of $t\rho$ and $t'\rho$, and γ' is an $\mathcal{E}$-solution for $C\gamma$, we call $\sigma' = \gamma\gamma'$ and $\sigma = \rho\sigma'$, so $t\sigma =_A t'\sigma$. Now we prove that σ is well-formed. The sorted variables in $\mathit{Var}(t) \cup \mathit{Var}(t')$ are $\{x^i_{s_i}\}^n_{i=1}$. We have that $x^i_{s_i}\sigma = y^i_{[s_i]}\sigma'$,$1 \leq i \leq n$, and γ' is an $\mathcal{E}$-solution for $y^i_{[s_i]}\gamma : s_i$, so $y^i_{[s_i]}\gamma\gamma' : s_i$, i.e, $x^i_{s_i}\sigma : s_i$. σ is idempotent because σ' is away from $\mathit{Vars}(t) \cup \mathit{Vars}(t') \cup \mathit{Vars}(C)$. □

Example 12. In the concurrency specification example, let $t \equiv \mathtt{u2}, x_{\mathtt{u}}$ and $t' \equiv y_{\mathtt{us}}$. Then $\sigma = \{x_{\mathtt{u}} \mapsto \mathtt{u1}, y_{\mathtt{us}} \mapsto \mathtt{u1}, \mathtt{u2}\}$ is an A-unifier of t and t' (because $\mathtt{u1}, \mathtt{u2} =_A \mathtt{u2}, \mathtt{u1}$). As $\mathit{ls}(t) = [\mathtt{us}]$ and $\mathit{ls}(t') = \mathtt{us}$, neither of them belonging to $\mathit{OS}(S)$, we don't have direct A-unification algorithms for t and t', so we compute $\mathit{kinded}(t, t')$, with answer $\rho = \{x_{\mathtt{u}} \mapsto z_{[\mathtt{us}]}, y_{\mathtt{us}} \mapsto v_{[\mathtt{us}]}\}$, and $C = z_{[\mathtt{us}]} : \mathtt{us} \wedge v_{[\mathtt{us}]} : \mathtt{us}$. Now $t\rho \equiv \mathtt{u2}, z_{[\mathtt{us}]}$ and $t'\rho \equiv v_{[\mathtt{us}]}$. There is a many-sorted A-unification algorithm at the kind level for $t\rho$ and $t'\rho$ that returns the answer $\gamma = \{z_{[\mathtt{us}]} \mapsto w_{[\mathtt{us}]}, v_{[\mathtt{us}]} \mapsto \mathtt{u2}, w_{[\mathtt{us}]}\}$. As $C\gamma = w_{[\mathtt{us}]} : \mathtt{us} \wedge \mathtt{u2}, w_{[\mathtt{us}]} : \mathtt{us}$, then $\gamma' = \{w_{[\mathtt{us}]} \mapsto \mathtt{u1}\}$ is an $\mathcal{E}$-solution for the system of sentences $C\gamma$, and $\{x_{\mathtt{u}} \mapsto \mathtt{u1}, y_{\mathtt{us}} \mapsto \mathtt{u1}, \mathtt{u2}\} = \sigma =_A \rho\gamma\gamma' = \{x_{\mathtt{u}} \mapsto \mathtt{u1}, y_{\mathtt{us}} \mapsto \mathtt{u2}, \mathtt{u1}\}$.

5.2 Calculus for unification strategies and rules

The calculus for unification uses the following strategies:

- Inference rules are applied with leftmost strategy.
- As we are computing E, A-normalized solutions and we have already shown the equivalence of SNR-rewriting with respect to E, A-normalized solutions for unification goals, we have built-in the following strategy in our calculus: we only apply a calculus rule if the composition of all computed substitutions remains idempotent E, A-normalized with respect to all extra variables and all the variables in the initial unification problem. This means that we must keep track of all extra variables that have not been instantiated to ground terms in order to be able to discard any narrowing step that violates this principle.
- As we have also proved the equivalence of E, A-normalized solutions with respect to canonical unification goals, we follow a second strategy in our calculus consisting in canonizing the unification problem after each use of a rule in the calculus, except for rules transitivity and congruence which are the only rules that don't apply substitutions to the unification problem.

We shall later prove that we don't miss any answer with these reductions of the state space. Our calculus for $\mathcal{E}$-solutions is defined by the inference rules in Figure 5. These rules transform *unification problems* of the form $t \to t'$, or $t|_p \to^1 x_k, t[x_k]_p \to t'$ (x_k fresh variable, with $k = [ls(t|_p)]$), both having the same meaning: find a substitution σ such that $t\sigma \to_{E,A} t'\sigma$. Note that in the second type of subproblem, t can be easily reconstructed as $t \equiv t[x_k]_p\rho$, with $\rho = \{x_k \mapsto t|_p\}$. The goal G' represents the rest of unification subproblems that have not been processed yet, if they exist. We show G' in an inference rule only when it can be affected by instantiation and further canonization.

– $[t]$ transitivity

$$\frac{t \to t'}{t \to^1 x_k, x_k \to t'}$$

where $k = [ls(t)]$

– $[c]$ congruence

$$\frac{t|_p \to^1 x_k, t[x_k]_p \to t'}{t|_{p.i} \to^1 y_{k'}, t[y_{k'}]_{p.i} \to t'}$$

where $t|_p \equiv f(t_1, \ldots, t_n)$, $1 \leq i \leq n$, $k' = [ls(t|_{p.i})]$, and $y_{k'}$ is a fresh variable

– $[r]$ reduction

$$\frac{t|_p \to^1 x_k, t[x_k]_p \to t' \ (\wedge G')}{(((C) \wedge t[r]_p \to t'(\wedge G'))\theta)\!\downarrow}$$

where $t|_p \notin \mathcal{X}$, $l \to r$ (*if* C) is a fresh instance of a rule in R_E and $\theta \in CSU_A(t|_p = l)$

– $[e]$ elimination

$$\frac{t \to t' \ (\wedge G')}{(G'\theta)\!\downarrow}$$

where $\theta \in CSU_A(t = t')$

– $[u]$ unification

$$\frac{eq(t, t') \to tt \ (\wedge G')}{(G'\theta)\!\downarrow}$$

where $\theta \in CSU_A(t = t')$

– $[m]$ membership

$$\frac{x_\kappa{:}s \to tt \ (\wedge G')}{(G'\theta)\!\downarrow}$$

where s' is a maximal sort such that $s' \leq s$ and $s' \leq \kappa$, $y_{s'}$ is a fresh variable, and $\theta = \{x_\kappa \mapsto y_{s'}\}$

Fig. 5. Inference rules for $\mathcal{E}$-solution by conditional narrowing.

Note that unification goals are a subset of unification problems. For any subproblem of the form $t \to t'$ (or $eq(t, t') \to tt$), if $t =_A t'$ then we always apply rule elimination (resp., rule unification) with substitution id, which is more general than any other possible computed answer. After applying rule transitivity we get a unification problem $t|_p \to^1 x_k, t[x_k]_p \to t'$, where we perform an actual narrowing step in t using rule reduction, or we perform an actual narrowing step in some proper subterm of t, applying several times rule congruence to reach the desired subterm, followed by an application of rule reduction. Such narrowing steps have a kinded variable x_k as target, because although t may have some sort s when it is sufficiently instantiated with some substitution σ, t will have kind $k = [s]$ for partial instantiations, but it will usually have no sort. Rule membership is needed to lower the type of a variable so it has a desired sort. It is the most general way of instantiating the variable.

Our transformation of the rules in R_E and R generates additional membership subgoals. Many of them are trivial and don't need any further instantiation after, or become trivial after several calculus steps are applied to other subgoals. By canonization, these trivial membership subgoals $t{:}s \to tt$ become $tt \to tt$ and they will be removed from the problem with the elimination rule and substitution id. If subgoals were not canonized, there would be a significant overhead in the calculus only in order to prove this trivial subgoals, not to mention all the overhead generated by applying narrowing to all subgoals that may exist in a rewriting path between a subgoal g and its canonized version $g{\downarrow}$.

Proposition 6. *Given an FPP executable* Mel *theory $(\Sigma, E \cup A)$ and its associated rewrite theory $\mathcal{R}_E = (\Sigma', A, R_E)$, after applying the* transitivity *rule followed by zero or more applications of the* congruence *rule to a unification problem of the form $t \to t'$ we get another unification problem of the form $t|_p \to^1 x_k, t[x_k]_p \to t'$ with k some kind in Σ'.*

Proof. Immediate, by induction on the number of congruence rules applied.

- Base case: zero congruence rules. Then $t \to t' \rightsquigarrow_{[t]} t \to^1 x_k, x_k \to t'$, with $k = [ls(t)]$. In this case $p = \epsilon$ and $t[x_k]_\epsilon \equiv x_k$.
- Induction case. We assume that after applying the congruence rule zero or more times we have: $t \to t' \rightsquigarrow_{[t]} \rightsquigarrow^*_{[c]} (t|_p \to^1 x_k, t[x_k]_p \to t')$. Then, if we apply the congruence rule again, by definition of the rule we get the unification problem $t|_{p.i} \to^1 y_{k'}, t[y_{k'}]_{p.i} \to t'$.

□

This proposition means that after applying the transitivity rule $[t]$ to a unification subgoal and before applying the reduction rule $[r]$, all generated unification subproblems that use the $\to^1$ symbol will have the same shape: $t|_p \to^1 x_k, t[x_k]_p \to t'$, for some $p \in Pos(t)$, with $k = [ls(t|_p)]$, and x_k is a fresh variable. As we always start our inferences from a unification goal, we can assume that a unification subproblem has this shape when there is a $\to^1$ symbol within the subproblem.

When we apply one of the calculus rules for unification to a unification problem G_i with some inference rule $[r]$ and substitution σ_i, yielding another unification problem G_{i+1}, we display it as $G_i \rightsquigarrow_{[r],\sigma_i} G_{i+1}$ and say that there exists a *narrowing step* from G_i to G_{i+1} using the substitution σ_i and the inference rule $[r]$. As a special case, $\sigma_i = id$ when we apply the transitivity or congruence rules. $[r]$ and σ_i may be omitted when their actual values are irrelevant or can be inferred.

Definition 20 (Computed Answer). *Given a unification goal G, if there is a* narrowing path *from G to the* empty problem *□, $G = G_0 \rightsquigarrow_{\sigma_1} G_1 \rightsquigarrow_{\sigma_2} \dots \rightsquigarrow_{\sigma_n}$ □, then we write $G \rightsquigarrow^*_\sigma$ □, with $\sigma = \sigma_1\sigma_2 \dots \sigma_n$, and call $\sigma_{Var(G)}$ a* computed answer *for G.*

The calculus for unification is sound and weakly complete, i.e., complete with respect to idempotent $R{\cup}E, A$-normalized solutions. We will prove completeness of the calculus with respect to canonized goals (by Lemma 3 and E, A-normalized idempotent solutions (more general than $R \cup E, A$-normalized solutions). In this way, we can independently apply this part of the calculus to any FPP executable Mel theory, even if it is the case that the Mel theory is not underlying some rewrite theory. For a condition in R_E, or a unification goal, $G \equiv \bigwedge_{i=1}^n t_i \to t'_i$ we define $G{\downarrow} \equiv \bigwedge_{i=1}^n t_i{\downarrow} \to t'_i{\downarrow}$. Recall that for a unification goal associated to a system of sentences, or a condition in R_E, $G{\downarrow} \equiv \bigwedge_{i=1}^n t_i{\downarrow} \to t'_i$ because t'_i is always tt or a Σ-pattern.

Theorem 4 (Soundness of the Calculus for $\mathcal{E}$-solution). *Given a narrowable rewrite theory $\mathcal{R} = (\Sigma, \mathcal{E}, R)$, its FPP executable* Mel *theory $(\Sigma, E \cup A)$, the associated rewrite theory $\mathcal{R}_E = (\Sigma', A, R_E)$, a system of sentences, and its associated unification goal G, if σ is a computed answer for $G{\downarrow}$ then σ is an idempotent E, A-normalized $\mathcal{E}$-solution for G.*

Proof. By Lemma 6, we only have to prove that σ is an $\mathcal{E}$-solution for $G{\downarrow}$. By construction of σ all computed answers are idempotent E, A-normalized. Now, we prove that σ is an $\mathcal{E}$-solution for each unification subproblem generated by narrowing from an initial unification subproblem $u{\downarrow} \to v$, by induction on the total number of narrowing steps. We prove that if σ is a computed answer for $t \to t'$, or $t|_p \to^1 x^p_{k_p}, t[x^p_{k_p}]_p \to t'$ (where t and all of its subterms are always canonical forms by definition of the calculus, so $t|_p{\downarrow} \equiv t|_p$), then $t\sigma \to_{E,A} t'\sigma$ so σ is an $\mathcal{E}$-solution for $t \to t'$.

Base case, one narrowing step:

- Elimination rule $[e]$. There are two subcases:
  - $tt \to tt$. Trivial with $\sigma = id$.
  - $t \to t'$ and $t\sigma =_A t'\sigma$, so $t \to t' \rightsquigarrow_{[e],\sigma} \square$. Then, by definition of $\to_{E,A}$, $t\sigma \to_{E,A} t'\sigma$.
- Unification rule $[u]$. $eq(t, t') \to tt$ and $t\sigma =_A t'\sigma$, so $eq(t, t') \to tt \rightsquigarrow_{[u],\sigma} \square$. Then, $eq(t\sigma, t'\sigma) \to^1_{E,A} tt$ using rule $eq(x_k, x_k) \to tt$ and substitution $\rho = \{x_k \mapsto t\sigma\}$, so $eq(t\sigma, t'\sigma) \to_{E,A} tt$.
- Membership rule $[m]$. $x_\kappa : s \to tt$, $\sigma = \{x_\kappa \mapsto y_{s'}\}$, s' maximal such that $s' \leq s$ and $s' \leq \kappa$, and $y_{s'}$ a fresh variable, so $x_\kappa : s \to tt \rightsquigarrow_{[m],\sigma} \square$, so $x_\kappa\sigma : s \to tt \equiv y_{s'} : s \to tt$. As $s' \leq s$ there is a membership $z_{s'} : s$ in E, and a rule $z_{s'} : s \to tt$ in R_E, so $y_{s'} : s \to_{E,A} tt$.

Induction case:

- Transitivity rule $[t]$. $t \to t' \rightsquigarrow_{[t]} t \to^1 x_k, x_k \to t' \rightsquigarrow^*_\sigma \square$. By I.H. $t\sigma \to_{E,A} t'\sigma$.
- Reduction rule [r]. The unification subproblem has form $t|_p \to^1 x_k, t[x_k]_p \to t'$, with x_k fresh variable. We apply rule $[r]$ because there is a rule $c \equiv l \to r$ *if* $\bigwedge_{i=1}^n t_i \to t'_i$ in R_E and there is an idempotent substitution θ, with $Dom(\theta) \subseteq Var(t|_p) \cup Var(l)$ and $\theta_{Var(t)}$ E, A-normalized, such that $t\theta =_A l\theta$, and also $Dom(\theta) \cap Var(t'_i) = \emptyset$ because c has fresh variables and t'_i is tt or an FPP Σ-pattern.
  Then, the narrowing derivation is $t|_p \to^1 x_k, t[x_k]_p \to t' \rightsquigarrow_{[r],\rho,\theta} \bigwedge_{i=1}^n (t_i\theta){\downarrow} \to t'_i \wedge (t[r]_p\theta){\downarrow} \to t' \rightsquigarrow^*_{\sigma'} \square$, $(t'\theta){\downarrow} \equiv t'$ because $Dom(\theta) \cap Var(t') = \emptyset$, and t' is tt or an FPP Σ-pattern, with σ' E, A-normalized with respect to all variables in $\bigwedge_{i=1}^n (t_i\theta){\downarrow} \to t'_i \wedge (t[r]_p\theta){\downarrow} \to t'$. Then $\sigma = \theta\sigma'$.
  For $1 \leq i \leq n$, $t_i\theta \to_{E,A} (t_i\theta){\downarrow}$, so $(t_i\theta)\sigma' \to_{E,A} (t_i\theta){\downarrow}\sigma'$. By I.H. $(t_i\theta){\downarrow}\sigma' \to_{E,A} t'_i\sigma'$, so $t_i\sigma \to_{E,A} t'_i\sigma'$, and also $t_i\sigma \to_{E,A} t'_i\sigma$ because $Dom(\theta) \cap Var(t'_i) = \emptyset$, so $t'_i\theta\sigma' \equiv t'_i\sigma'$. As $t|_p\theta =_A l\theta$ then $t|_p\sigma =_A l\sigma$ so $t|_p\sigma \to^1_{E,A} r\sigma$. Then, by definition of $\to^1_{E,A}$, $t[t|_p\sigma]_p \to^1_{E,A} t[r\sigma]_p$, so $t[t|_p\sigma]_p\sigma \to^1_{E,A} t[r\sigma]_p\sigma$ which, as σ is idempotent and $t[t|_p]_p \equiv t$, is equivalent to $t\sigma \to^1_{E,A} t[r]_p\sigma$.
  $t[r]_p\theta \to_{E,A} (t[r]_p\theta){\downarrow}$, so $t[r]_p\theta\sigma' \to_{E,A} (t[r]_p\theta){\downarrow}\sigma'$. As by I.H. $(t[r]_p\theta){\downarrow}\sigma' \to_{E,A} t'\sigma'$, then $t[r]_p\theta\sigma' \to_{E,A} t'\sigma'$, i.e., $t[r]_p\sigma \to_{E,A} t'\sigma$. As $t\sigma \to^1_{E,A} t[r]_p\sigma$, then $t\sigma \to_{E,A} t'\sigma$.
- Congruence rule $[c]$. By I.H. $t\sigma \to_{E,A} t'\sigma$.
  As $t|_{p.i}\sigma \to^1_{E,A} y_{k'}\sigma$, then $t[t|_{p.i}\sigma]_{p.i} \to^1_{E,A} t[y_{k'}\sigma]_{p.i}$ and $t[t|_{p.i}\sigma]_{p.i}\sigma \to^1_{E,A} t[y_{k'}\sigma]_{p.i}\sigma$ which, as σ is idempotent and $t[t|_{p.i}]_{p.i} \equiv t$, is equivalent to $t\sigma \to^1_{E,A} t[y_{k'}]_{p.i}\sigma$. Again, as $t[y_{k'}]_{p.i}\sigma \to_{E,A} t'\sigma$, then $t\sigma \to_{E,A} t'\sigma$.

□

Theorem 5 (Weak Completeness of the Calculus for $\mathcal{E}$-solution). *Given a narrowable rewrite theory $\mathcal{R} = (\Sigma, \mathcal{E}, R)$, its FPP executable* Mel *theory $(\Sigma, E \cup A)$, the associated rewrite* *theory $\mathcal{R}_E = (\Sigma', A, R_E)$, a system of sentences F, and its associated unification goal G, if σ is an idempotent E, A-normalized $\mathcal{E}$-solution for G then there is an idempotent E, A-normalized substitution γ, with $\sigma \ll_A \gamma_{Var(G)}$, such that $G{\downarrow} \rightsquigarrow^*_\gamma \Box$.*

Proof. Every computed answer γ is idempotent E, A-normalized by definition of the calculus. If σ is an $\mathcal{E}$-solution for G then, by Lemma 6, for each unification subgoal $t \rightarrow t'$, $t{\downarrow}\sigma \rightarrow_N t'\sigma$. We prove the theorem using induction on the number of unification subgoals plus the number of $\rightarrow^1_N$ rewrite steps, including the subgoals and rewrite steps due to conditions.

Base case. One subgoal, zero rewrite steps. There are several cases:

- $F \equiv t = t'$, and $eq(t, t'){\downarrow}\sigma \rightarrow_N tt$ because $t{\downarrow}\sigma =_A t'{\downarrow}\sigma$. There are two subcases:
  - $t{\downarrow} =_A t'{\downarrow}$. Then $G{\downarrow} \equiv tt \rightarrow tt \rightsquigarrow_{[e],id} \Box$, and trivially $\sigma \ll_A id$.
  - $t{\downarrow} \neq_A t'{\downarrow}$. Then $G{\downarrow} \equiv eq(t{\downarrow}, t'{\downarrow}) \rightarrow_{tt}$, and there exists $\gamma \in CSU_A(t{\downarrow} = t'{\downarrow})$ such that $\sigma \ll_A \gamma$, so $G{\downarrow} \equiv eq(t{\downarrow}, t'{\downarrow}) \rightarrow_{E,A} t \rightsquigarrow_{[u],\gamma} \Box$.
- $F \equiv t := t'$, and $t'{\downarrow}\sigma \rightarrow_N t\sigma$ because $t{\downarrow}\sigma =_A t{\downarrow}\sigma$ ($t'{\downarrow} \equiv t$ because t is a Σ-pattern). Then there exists $\gamma \in CSU_A(t'{\downarrow} = t)$ such that $\sigma \ll_A \gamma$, so $G{\downarrow} \equiv t'{\downarrow} \rightarrow t \rightsquigarrow_{[e],\gamma} \Box$.
- $F \equiv t : s$, and $t{:}s{\downarrow}\sigma \rightarrow_N tt$ because $t{:}s{\downarrow} \equiv tt$ (i.e., t and $t{\downarrow}$ have sort s). Then, again, $G{\downarrow} \equiv tt \rightarrow tt \rightsquigarrow_{[e],id} \Box$, and trivially $\sigma \ll_A id$.

Induction case. We consider two subcases:

- Several subgoals in the initial problem (there may be zero $\rightarrow^1_N$ rewrite steps): $G \equiv t \rightarrow t' \wedge G'$. As σ is an $\mathcal{E}$-solution for $t \rightarrow t'$, which has at most the same number of rewrite steps and one less subgoal than G, so I.H. applies and there exists an idempotent E, A-normalized substitution γ such that $\sigma \ll_A \gamma_{Var(t \rightarrow t')}$, so $\sigma = \gamma_{Var(t \rightarrow t')}\rho$ for some idempotent E, A-normalized substitution ρ, such that $t \rightarrow t' \rightsquigarrow^*_\gamma \Box$. Then $t \rightarrow t' \wedge G' \rightsquigarrow^*_\gamma (G'\gamma){\downarrow}$.
  As $\sigma = \gamma_{Var(t \rightarrow t')}\rho$ and $Var(G') \cap Dom(\gamma) \subseteq Var(t \rightarrow t')$ then $G'\gamma \equiv G'\gamma_{Var(t \rightarrow t')}$, so ρ is an $\mathcal{E}$-solution for $G'\gamma$ because $\sigma = \gamma_{Var(t \rightarrow t')}\rho$. Then I.H applies to $(G'\gamma){\downarrow}$, which has at most the same number of rewrite steps and one less subgoal than G, and there exists an idempotent E, A-normalized substitution θ such that $\rho \ll_A \theta_{Var((G'\gamma)){\downarrow}}$ and $(G'\gamma){\downarrow} \rightsquigarrow^*_\theta \Box$.
  $Var((G'\gamma){\downarrow}) \subseteq Var(G\gamma)$ $Dom(\theta) \cap Var(G\gamma) \subseteq Var((G'\gamma){\downarrow})$, so $\theta_{Var((G'\gamma){\downarrow})} = \theta_{Var(G\gamma)}$, and $\rho \ll_A \theta_{Var(G\gamma)}$. Let's call $v = Var(G)$. As $v \cap Dom(\gamma) \subseteq Var(t \rightarrow t')$ then $\gamma_{Var(t \rightarrow t')} = \gamma_v$, and $\sigma = \gamma_v\rho$. Recall that $Dom(\theta_{Var(G\gamma)}) \subseteq Ran(\gamma_v) \cup v$. Then $\sigma = \gamma_v\rho \ll_A \gamma_v\theta_{Var(G\gamma)} = \gamma_v(\theta_{Ran(\gamma_v)} \cup \theta_v) = (\gamma\theta)_v$.
- One subgoal in the initial problem and at least one $\rightarrow^1_N$ rewrite step: $G \equiv t \rightarrow t'$, $t{\downarrow}\sigma \rightarrow^1_N t'' \rightarrow_N t'\sigma$, σ is an N-solution for $G{\downarrow}$, and $t'\sigma$ is a canonical form.
  We check each type of rule that can have been applied in $t{\downarrow}\sigma \rightarrow^1_N t''$:
  1. $eq(x_k, x_k) \rightarrow tt$, so $t \equiv eq(t_1, t_2)$, with $t_1{\downarrow} \neq_A t_2{\downarrow}$ (else $t{\downarrow} \equiv tt$ and there would not be any $\rightarrow^1_N$ step because tt is a canonical form), $t_1{\downarrow}\sigma =_A t_2{\downarrow}\sigma$, $t' \equiv tt$, and $eq(t_1{\downarrow}, t_2{\downarrow}) \rightarrow^1_N tt \rightarrow_N tt$. Then there exists $\gamma \in CSU_A(t_1{\downarrow} = t_2{\downarrow})$ such that $\sigma \ll_A \gamma$, so $eq(t_1{\downarrow}, t_2{\downarrow}) \rightarrow tt \rightsquigarrow_{[t]} eq(t_1{\downarrow}, t_2{\downarrow}) \rightarrow^1 x_k, x_k \rightarrow tt \rightsquigarrow_{[u],\gamma} tt \rightarrow tt \rightsquigarrow_{[e]} \Box$.
  2. $c \equiv l{:}s \rightarrow tt \text{ } if \text{ } C$, so $t \equiv t_1{:}s$, with $ls(t_1{\downarrow}) \not\leq s$ (else $t{\downarrow} \equiv tt$), $t_1{\downarrow}\sigma =_A l\sigma'_c$, σ'_c is an idempotent $\mathcal{E}$-solution for C (E, A-normalized with respect to $Extra(C)$), $t' \equiv tt$, $t'' \equiv tt$, and $t_1{\downarrow}\sigma{:}s \rightarrow^1_N tt \rightarrow_N tt$. $Dom(\sigma) \cap Dom(\sigma'_c) = \emptyset$, so $\sigma \cup \sigma'_c$ is an A-unifier for $t_1{\downarrow} = l$. Let's call $v = Var(t) = Var(G)$, and $w = Var(l)$. Then there exists $\gamma \equiv \gamma_v \cup \gamma_w \in CSU_A(t_1{\downarrow} = l)$ such that $t_1\gamma_v =_A l\gamma_w$, and $\sigma \cup \sigma'_c \ll_A \gamma$, so $\sigma \cup \sigma'_c =_A \gamma\rho$ for some idempotent substitution ρ. σ is E, A-normalized, σ'_c is E, A-normalized except maybe for some subset of $Dom(\gamma)$, so ρ must be E, A-normalized. Then $t_1{:}s \rightarrow tt \rightsquigarrow_{[t]} t_1{:}s \rightarrow^1 x_k, x_k \rightarrow tt \rightsquigarrow_{[r],\gamma} (C\gamma){\downarrow}$. As C has fresh variables then $Dom(\sigma) \cap Var(C) = \emptyset$, so $C\sigma'_c \equiv C(\sigma \cup \sigma'_c) \equiv C(\gamma\rho)$.

σ'_c is an $\mathcal{E}$-solution for C with less than n rewrite steps, so ρ is an idempotent E, A-normalized $\mathcal{E}$-solution for $C\gamma$ with less than n rewrite steps, with $(C\gamma){\downarrow} \equiv (C\gamma_w){\downarrow}$ because $\mathit{Dom}(\gamma_v) \cap \mathit{Var}(C) = \emptyset$. By I.H. there exists θ, with $\rho \ll_A \theta_{\mathit{Var}(C\gamma)}$ such that $(C\gamma){\downarrow} \rightsquigarrow^*_\theta \square$. The composition of the substitutions in the narrowing derivation is $\gamma\theta$. We have to prove that $\sigma \ll_A (\gamma\theta)_v$. As $\mathit{Var}(C\gamma) \cap v = \emptyset$ then $\mathit{Dom}(\theta) \cap v = \emptyset$. As $\sigma \cup \sigma'_c =_A \gamma\rho$, $\mathit{Dom}(\sigma'_c) \cap v = \emptyset$, and $\mathit{Dom}(\sigma) \subseteq v$, then $\sigma = (\gamma_v \rho_{\mathit{Ran}(\gamma_v)}) \cup \rho_v$.
$(C\gamma){\downarrow} \rightsquigarrow^*_\theta \square$ so $\mathit{Dom}(\theta) \subseteq \mathit{Var}(C\gamma) \cup v'$, with v' a set of fresh variables generated by the narrowing calculus, and $\mathit{Ran}(\gamma) = \mathit{Ran}(\gamma_v) = \mathit{Ran}(\gamma_w)$ because $\gamma \in \mathit{CSU}_A(t_1{\downarrow} = l)$ and A is regular. Then, $\mathit{Dom}(\theta) \cap \mathit{Ran}(\gamma) = \mathit{Dom}(\theta) \cap \mathit{Ran}(\gamma_v) \subseteq \mathit{Var}(C\gamma)$, and $\theta_{\mathit{Var}(C\gamma)} \ll_A \theta_{\mathit{Ran}(\gamma)}$. As $\rho \ll_A \theta_{\mathit{Var}(C\gamma)}$, then $\rho \ll_A \theta_{\mathit{Ran}(\gamma_v)}$, and $\rho_{\mathit{Ran}(\gamma_v)} \ll_A \theta_{\mathit{Ran}(\gamma_v)}$.
Now, $\gamma_v \rho_{\mathit{Ran}(\gamma_v)} \ll_A \gamma_v \theta_{\mathit{Ran}(\gamma_v)} = \gamma_v \theta_{\mathit{Ran}(\gamma_v)} \cup \theta_v$ because $\mathit{Dom}(\theta) \cap v = \emptyset$, so $\theta_v = id$. But $\gamma_v \theta_{\mathit{Ran}(\gamma_v)} \cup \theta_v = (\gamma\theta)_v$, so $\gamma_v \rho_{\mathit{Ran}(\gamma_v)} \ll_A (\gamma\theta)_v$.
In conclusion: $\sigma = \gamma_v \rho_{\mathit{Ran}(\gamma_v)} \cup \rho_v \ll_A (\gamma\theta)_v$.

3. $c \equiv l \to r$ *if* C, not in cases 1 or 2. Then $t' \neq tt$, $t'' \neq tt$, and $t{\downarrow}\sigma \to^1_N (t{\downarrow}\sigma)[r\sigma'_c]_p \to_N t'\sigma$ because $(t{\downarrow}\sigma)|_p =_A l\sigma'_c$ and σ'_c is an $\mathcal{E}$-solution for C (E, A-normalized with respect to $\mathit{Extra}(C)$). As σ is E, A normalized and $l \notin \mathcal{X}$, then we cannot rewrite inside a position instantiated by σ or a variable position, so p must be an already existing non variable position in $t{\downarrow}$ (i.e., $p \in \mathit{Pos}_\Sigma(t{\downarrow})$). Also $(t{\downarrow}\sigma)[r\sigma'_c]_p \equiv t{\downarrow}[r\sigma'_c]_p\sigma$ because $\mathit{Dom}(\sigma) \cap \mathit{Var}(r\sigma'_c) = \emptyset$, and $t{\downarrow}[r\sigma'_c]_p\sigma \equiv t{\downarrow}[r]_p(\sigma \cup \sigma'_c)$ because $\mathit{Dom}(\sigma'_c) \cap \mathit{Var}(t{\downarrow}) = \emptyset$ and $\mathit{Dom}(\sigma) \cap \mathit{Dom}(\sigma'_c)) = \emptyset$.
   As in the previous subcase, $t{\downarrow}\sigma \to^1_N t{\downarrow}[r]_p(\sigma \cup \sigma'_c) \to_N t'\sigma$, and there exists $\gamma = \gamma_v \cup \gamma_w \in \mathit{CSU}_A(t{\downarrow}|_p = l)$, with $v = \mathit{Var}(G)$, and $w = \mathit{Var}(c)$, such that $t{\downarrow}|_p\gamma_v =_A l\gamma_w$. Then there exists $\gamma \equiv \gamma_v \cup \gamma_w \in \mathit{CSU}_A(t{\downarrow}|_p{\downarrow} = l)$ such that $t{\downarrow}|_p\gamma_v =_A l\gamma_w$, and $\sigma \cup \sigma'_c \ll_A \gamma$, so $\sigma \cup \sigma'_c =_A \gamma\rho$ for some idempotent substitution ρ. σ is E, A-normalized, σ'_c is E, A-normalized except maybe for some subset of $\mathit{Dom}(\gamma)$, so ρ must be E, A-normalized.
   Although t' is an FPP Σ-pattern, we reason in this part of the proof as if t' could be any term, for compatibility with the equivalent proof for the calculus for reachability, obtaining then a more general result.
   Then $t{\downarrow} \to t'{\downarrow} \rightsquigarrow_{[t]} t{\downarrow} \to^1 x_k, x_k \to t'{\downarrow} \rightsquigarrow^*_{[c]} t{\downarrow}|_p \to^1 y_{k'}, t{\downarrow}[y_{k'}]_p \to t'{\downarrow} \rightsquigarrow_{[r],\gamma} (C\gamma){\downarrow} \wedge (t{\downarrow}[r]_p\gamma){\downarrow} \to (t'{\downarrow}\gamma){\downarrow}$ (recall that $(t'{\downarrow}\gamma){\downarrow} \equiv t'$). Let's call $u = \mathit{Var}(C\gamma \wedge t{\downarrow}[r]_p\gamma)$. ρ is an $\mathcal{E}$-solution for $C\gamma \wedge t{\downarrow}[r]_p\gamma \to (t'{\downarrow}\gamma){\downarrow}$ with one less subgoal and one less rewriting step than the $\mathcal{E}$-solution σ for $t \to t'$, so I.H. applies and there exist an E, A-normalized substitution θ such that $\rho \ll_A \theta_u$ and $(C\gamma){\downarrow} \wedge (t{\downarrow}[r]_p\gamma){\downarrow} \to (t'{\downarrow}\gamma){\downarrow} \rightsquigarrow^*_\theta \square$.
   As $\rho \ll_A \theta_u$ then $\rho \ll_A \theta_{\mathit{Var}(C\gamma)}$, so $\rho_{\mathit{Var}(C\gamma)} \ll_A \theta_{\mathit{Var}(C\gamma)}$. $\sigma \cup \sigma'_c =_A \gamma\rho$, so $\sigma = (\sigma \cup \sigma'_c)_v = (\gamma\rho)_v$. Then we have $\sigma = \gamma_v \rho_{\mathit{Ran}(\gamma_v)} \cup \rho_v$. As $\mathit{Dom}(\gamma_v) \cap \mathit{Dom}(\rho_v) = \emptyset$ then $\gamma_v \rho_{\mathit{Ran}(\gamma_v)} \cup \rho_v = \gamma_v(\rho_{\mathit{Ran}(\gamma_v)} \cup \rho_v)$, and $\sigma = \gamma_v(\rho_{\mathit{Ran}(\gamma_v)} \cup \rho_v) = \gamma_v \rho_{\mathit{Var}(C\gamma)} \ll_A \gamma_v \theta_{\mathit{Var}(C\gamma)} = \gamma_v(\theta_{\mathit{Ran}(\gamma_v)}) \cup \theta_v)$. As $\mathit{Dom}(\gamma_v) \cap \mathit{Dom}(\theta_v) = \emptyset$, then $\gamma_v(\rho_{\mathit{Ran}(\gamma_v)} \cup \rho_v) = \gamma_v \rho_{\mathit{Ran}(\gamma_v)} \cup \rho_v$, so $\sigma \ll_A \gamma_v \rho_{\mathit{Ran}(\gamma_v)} \cup \rho_v = (\gamma\rho)_v$.

□

6 Reachability by conditional narrowing

In this part of the calculus, given an FPP narrowable rewrite theory $\mathcal{R} = (\Sigma, E \cup A, R)$ and a reachability goal G, we will solve the normalized reachability goal $G{\downarrow}$ and prove that it has the same E, A-normalized solutions. We will use a transformed set of rules $\tilde{R}$ where for each rule $l \to r$ (*if* $\bigwedge_{i=1}^n A_i$) in R, there is a rule $l \to r$ (*if* $\bigwedge_{i=1}^n A'_i$) in $\tilde{R}$ such that:

- if A_i has the form $t_i \to t'_i$ then A'_i is $t_i \Rightarrow t'_i$,

- if A_i has the form $t_i : s_i$ then A'_i is $t_i{:}s_i \to tt$,
- if A_i has the form $t_i := t'_i$ then A'_i is $t'_i \to t_i$, and
- if A_i has the form $t_i = t'_i$ then A'_i is $eq(t_i, t'_i) \to tt$.

That is, we apply the same transformation that we used in the rewrite theory associated to a MEL theory, and replace each $\to$ symbol in conditions with a new $\Rightarrow$ symbol, so we can distinguish reachability conditions from equational conditions.

6.1 Calculus for reachability strategies and rules

Reachability by conditional narrowing is achieved using the previous calculus rules in Figure 5, extended with the calculus rules in Figure 6. These new rules transform *reachability problems* that have the form $t \Rightarrow t'$, $t|_p \to^1 x_k, t[x_k]_p \Rightarrow t'$, or $t|_p \Rightarrow^1 x_k, t[x_k]_p \Rightarrow t'$ (x_k being a fresh variable, where $k = [ls(t|_p)]$), all of them having the same meaning: find a substitution σ such that $t\sigma \twoheadrightarrow_{R\cup E,A} t'\sigma$. As for unification goals, reachability goals are a subset of reachability problems. We show the rest of the reachability goal, G', in an inference rule only when it can be affected by instantiation and further canonization.

The calculus for reachability uses the following strategies:

- Inference rules are applied with leftmost strategy.
- As we are computing E, A-normalized solutions and we have already shown the equivalence of SNR-rewriting with respect to E, A-normalized solutions for reachability goals, we have built-in the following strategy in our calculus: we only apply a calculus rule if the composition of all computed substitutions remains idempotent E, A-normalized with respect to all extra variables and all the variables in the initial reachability problem, that is, when we apply a rule $l \to r$ $(if\ C)$ in $\tilde{R}$ the only variables that need not be instantiated with an idempotent E, A-normalized substitution are those in $Var(l)$.
- As we have also proved the equivalence of E, A-normalized solutions with respect to canonical reachability goals, we follow a second strategy in our calculus consisting in canonizing the reachability problem after each use of a conditional rule in the calculus.
- Rule rewrite is applied on $t|_p$ with substitution θ only if **the whole term** $t\theta$ is E,A-normalized.
- A list of reachability problems is kept. Initially the list holds the original problem. Each new reachability problem generated by the calculus is checked against the current list. If the problem is a renaming and/or reordering of any element in the list, it gets discarded.

We explain the meaning of these rules and prove that the calculus is sound and weakly complete. Recall that we have defined $\twoheadrightarrow_{R\cup E,A}$ as $\to^*_{R\cup E,A}; =_A$.

- The reflexivity rule applies the $=_\mathcal{E}$ part of the definition for $\twoheadrightarrow_{R\cup E,A}$. It is the only rule that having a $\Rightarrow$ symbol as an antecedent doesn't have a $\Rightarrow$ symbol as a consequent, so it has always to be applied in every derivation from a subproblem of the form $t_i \Rightarrow t'_i$ to get rid of the $\Rightarrow$ symbol. If a solution σ generates a derivation with zero rewrite steps in $\twoheadrightarrow_{R\cup E,A}$, this means that $t_i\sigma =_E t'_i\sigma$, so we can find this substitution or a more general one by applying the reflexivity rule. The resulting subproblem $eq(t_i, t'_i)$ will be solved using the calculus rules for unification.
- The transitivity rule has been expanded. Now it can also apply the $\to^1_{R,A}$ part of the definition for $\twoheadrightarrow_{R\cup E,A}$, invoking the use of the congruence and rewrite rules to generate one actual reachability step ($\Rightarrow^1$) for reachability subgoals $t \Rightarrow t'$.
- The congruence rule has been expanded to deal with $\to^1$ followed by $\Rightarrow$, and $\Rightarrow^1$ followed by $\Rightarrow$. It has the same meaning as in the calculus for $\mathcal{E}$-Solution.

– $[x]$ reflexivity

$$\frac{t \Rightarrow t'}{eq(t,t')\downarrow \to tt}$$

– $[t]$ transitivity

$$\frac{t \Rightarrow t'}{t \to^1 x_k, x_k \Rightarrow t'} \qquad \frac{t \Rightarrow t'}{t \Rightarrow^1 x_k, x_k \Rightarrow t'}$$

where $t \notin \mathcal{X}$, and $k = [ls(t)]$

– $[c]$ congruence

$$\frac{t|_p \to^1 x_k, t[x_k]_p \Rightarrow t'}{t|_{p.i} \to^1 y_{k'}, t[y_{k'}]_{p.i} \Rightarrow t'} \qquad \frac{t|_p \Rightarrow^1 x_k, t[x_k]_p \Rightarrow t'}{t|_{p.i} \Rightarrow^1 y_{k'}, t[y_{k'}]_{p.i} \Rightarrow t'}$$

with $t|_p \equiv f(t_1, \ldots, t_n)$, $1 \leq i \leq n$, $t_i \notin \mathcal{X}$, $k' = [ls(t|_{p.i})]$, and $y_{k'}$ fresh variable

– $[r]$ reduction

$$\frac{t|_p \to^1 x_k, t[x_k]_p \Rightarrow t' \ (\wedge G')}{(((C) \wedge t[r]_p \Rightarrow t'(\wedge G'))\theta)\downarrow}$$

where $t|_p \notin \mathcal{X}$, $l \to r$ $(\mathit{if}\ C)$ is a fresh rule in R_E and $\theta \in CSU_A(t|_p = l)$

– $[w]$ rewrite

$$\frac{t|_p \Rightarrow^1 x_k, t[x_k]_p \Rightarrow t' \ (\wedge G')}{(((C) \wedge t[r]_p \Rightarrow t' \ (\wedge G'))\theta)\downarrow}$$

where $t|_p \notin \mathcal{X}$, $l \to r$ $(\mathit{if}\ C)$ is a fresh rule in $\tilde{R}$, $\theta \in CSU_A(t|_p = l)$, and $t\theta$ E,A-normalized

Fig. 6. Inference rules for reachability by conditional narrowing.

- The reduction rule has been expanded to deal with $\rightarrow^1$ followed by $\Rightarrow$. It also has the same meaning as in the calculus for $\mathcal{E}$-Solution.
- The rewrite rule is the only one that may generate instantiations, by using some rule $\tilde{r}$ from $\tilde{R}$. It gets rid of the $\Rightarrow^1$ symbol generated by the transitivity rule, and propagated by the congruence rule, transforming equational and membership conditions in rule r from R into their equivalent unification conditions in $\mathcal{R}_E$. After the congruence rule has selected a subterm $t|_p$, we apply the rewrite rule, using rule $\tilde{r}$ with some A-unifier θ, *only if* the whole instantiated term $t\theta$ is E, A-normalized. This is an improvement over previous reachability calculi for narrowing that only required the instantiated subterm $t|_p\theta$ to be E, A-normalized.

Theorem 6 (Soundness of the Calculus for Reachability). *Given a narrowable rewrite theory $\mathcal{R} = (\Sigma, \mathcal{E}, R)$, its FPP executable* Mel *theory $(\Sigma, E \cup A)$, the associated rewrite theory $\mathcal{R}_E = (\Sigma', A, R_E)$, and a reachability goal G, if σ is a computed answer for $G\!\downarrow$, using the transformed set of rules $\tilde{R}$, then σ is a solution for G.*

Proof. We prove that given a reachability problem $G \equiv g(\wedge G')$, if $G\!\downarrow \rightsquigarrow^*_\sigma \square$ then σ is a solution for G in $\rightarrow_{R/\mathcal{E}}$. In particular, we prove that if $g \equiv t \dashrightarrow^1 x_k, x_k \Rightarrow t' \rightsquigarrow^*_\sigma \square$, where $\dashrightarrow$ can be either $\rightarrow$ or $\Rightarrow$, then σ is a solution for $t \Rightarrow t'$ in $\rightarrow_{R/\mathcal{E}}$. As $\mathcal{R}$ is narrowable, and by Lemma 7, it is enough to prove that σ is an N-solution for $G\!\downarrow$. Soundness of the calculus for reachability is proved by induction on the total number of narrowing steps for each unification subproblem generated by narrowing from $G\!\downarrow$. By our previous proof of soundness in Theorem 4, we know that if we compute a solution σ for $t \rightarrow t'$ then $t\sigma \rightarrow_N t'\sigma$.

Base case: one narrowing step. The calculus rules in Figure 6 cannot compute a solution in one narrowing step, so we are in one of the base cases already proved for Theorem 4, with some unification goal $G\!\downarrow \equiv t \rightarrow t'$, so σ is an $\mathcal{E}$-solution for $G\!\downarrow$, hence a solution for $G\!\downarrow$ in $\rightarrow_{R\cup E,A}$.

Induction case: The cases where the first rule applied to $g\!\downarrow$ is shown in Figure 5 has already been proved for unification goals in Theorem 4. The same proof is valid for reachability goals *mutatis mutandis*, so we only check the cases where the first rule applied to $g\!\downarrow$ is one of the rules in Figure 6.

- Reflexivity rule: any computed answer σ is a solution for $t = t'$ and $G'\!\downarrow$. Then, as seen in the base case, σ is a solution for $g \equiv t \Rightarrow t'$ and, by I.H., σ is also a solution for $G'\!\downarrow$, so σ is a solution for $G\!\downarrow$ in $\rightarrow_{R\cup E,A}$. We skip the part of the proof related to G' in the rest of cases, as it is always the same.
- Transitivity rule: by I.H. σ is a solution for $t \Rightarrow t'$ in $\rightarrow_{R\cup E,A}$.
- Congruence rule: by I.H. σ is a solution for $t \Rightarrow t'$ in $\rightarrow_{R\cup E,A}$.
- Reduction rule: the reachability subproblem has form $t|_p \rightarrow^1 x_k, t[x_k]_p \Rightarrow t'$, with x_k a fresh variable. We apply rule $[r]$ because there is a rule $c \equiv l \rightarrow r$ *if* $\bigwedge_{i=1}^n t_i \rightarrow t'_i$ in R_E and there is an idempotent substitution θ, with $Dom(\theta) \subseteq Var(t|_p) \cup Var(l)$ and $\theta_{Var(t)}$ E, A-normalized, such that $t\theta =_A l\theta$, and also $Dom(\theta) \cap Var(t'_i) = \emptyset$ because c has fresh variables and t'_i is tt or an FPP Σ-pattern.
  Then, the narrowing derivation is $t|_p \rightarrow^1 x_k, t[x_k]_p \Rightarrow t' \rightsquigarrow_{[n],c,\theta} \bigwedge_{i=1}^n (t_i\theta)\!\downarrow \rightarrow t'_i \wedge (t[r]_p\theta)\!\downarrow \Rightarrow (t'\theta)\!\downarrow \rightsquigarrow^*_{\sigma'} \square$, with σ' E, A-normalized with respect to all variables in $\bigwedge_{i=1}^n (t_i\theta)\!\downarrow \rightarrow t'_i \wedge (t[r]_p\theta)\!\downarrow \Rightarrow (t'\theta)\!\downarrow$. Then $\sigma = \theta\sigma'$.
  For $1 \leq i \leq n$, $t_i\theta \rightarrow_{E,A} (t_i\theta)\!\downarrow$, so $(t_i\theta)\sigma' \rightarrow_{E,A} (t_i\theta)\!\downarrow\!\sigma'$. By I.H. $(t_i\theta)\!\downarrow\!\sigma' \rightarrow_{E,A} t'_i\sigma'$, so $t_i\sigma \rightarrow_{E,A} t'_i\sigma'$, and also $t_i\sigma \rightarrow_{E,A} t'_i\sigma$ because $Dom(\theta) \cap Var(t'_i) = \emptyset$, so $t'_i\theta\sigma' \equiv t'_i\sigma'$. As $t|_p\theta =_A l\theta$ then $t|_p\sigma =_A l\sigma$ so $t|_p\sigma \rightarrow^1_{E,A} r\sigma$. Then, by definition of $\rightarrow^1_{E,A}$, $t[t|_p\sigma]_p \rightarrow^1_{E,A} t[r\sigma]_p$, so $t[t|_p\sigma]_p\sigma \rightarrow^1_{E,A} t[r\sigma]_p\sigma$ which, as σ is idempotent and $t[t|_p]_p \equiv t$, is equivalent to $t\sigma \rightarrow^1_{E,A} t[r]_p\sigma$.

$t[r]_p\theta \rightarrow_{E,A} (t[r]_p\theta)\!\downarrow$, so $t[r]_p\theta\sigma' \rightarrow_{E,A} (t[r]_p\theta)\!\downarrow\!\sigma'$. By I.H. $(t[r]_p\theta)\!\downarrow\!\sigma' \rightarrow_{R\cup E,A} (t'\theta)\!\downarrow\!\sigma'$. As has been shown in other cases, then $t[r]_p\theta\sigma' \rightarrow_{R\cup E,A} t'\theta\sigma'$, i.e., $t[r]_p\sigma \rightarrow_{R\cup E,A} t'\sigma$. As $t\sigma \rightarrow^1_{E,A} t[r]_p\sigma$, then $t\sigma \rightarrow_{R\cup E,A} t'\sigma$.

- Rewrite rule: the reachability subproblem has form $t|_p \Rightarrow^1 x_k, t[x_k]_p \Rightarrow t'$, with x_k a fresh variable. We apply rule $[r]$ because there is a rule $c \equiv l \Rightarrow r \ \mathit{if} \ \bigwedge_{i=1}^n t_i \dashrightarrow t'_i$ in $\tilde{R}$ (where $\dashrightarrow$ can be either $\rightarrow$ or $\Rightarrow$) and there is an idempotent substitution θ, with $\mathit{Dom}(\theta) \subseteq \mathit{Var}(t|_p) \cup \mathit{Var}(l)$ and $\theta_{\mathit{Var}(t)}$ E,A-normalized, such that $t\theta =_A l\theta$.

  Then, the narrowing derivation is $t|_p \Rightarrow^1 x_k, t[x_k]_p \Rightarrow t' \leadsto_{[w],c,\theta} \bigwedge_{i=1}^n (t_i\theta)\!\downarrow \dashrightarrow (t'_i\theta)\!\downarrow \wedge (t[r]_p\theta)\!\downarrow \Rightarrow (t'\theta)\!\downarrow \leadsto^*_{\sigma'} \Box$, with σ' E,A-normalized with respect to all variables in $\bigwedge_{i=1}^n (t_i\theta)\!\downarrow \dashrightarrow (t'_i\theta)\!\downarrow \wedge (t[r]_p\theta)\!\downarrow \Rightarrow (t'\theta)\!\downarrow$. Then $\sigma = \theta\sigma'$.

  By I.H. σ' is an solution of $(t_i\theta)\!\downarrow \dashrightarrow (t'_i\theta)\!\downarrow$, for $1 \leq i \leq n$. Then, by Lemma 7, σ' is a solution for $t_i\theta \dashrightarrow t'_i\theta$, so σ is a solution for $t_i \dashrightarrow t'_i$, and $t|_p\sigma \rightarrow^1_{R\cup E,A} r\sigma$.

  Then, by definition of $\rightarrow^1_{R\cup E,A}$, $t[t|_p\sigma]_p \rightarrow^1_{R\cup E,A} t[r\sigma]_p$, so $t[t|_p\sigma]_p\sigma \rightarrow^1_{R\cup E,A} t[r\sigma]_p\sigma$ which, as σ is idempotent and $t[t|_p]_p \equiv t$, is equivalent to $t\sigma \rightarrow^1_{R\cup E,A} t[r]_p\sigma$.

  By I.H. σ' is a solution for $(t[r]_p\theta)\!\downarrow \Rightarrow (t'\theta)\!\downarrow$. Then, by Lemma 7, σ' is a solution for $t[r]_p\theta \Rightarrow t'\theta$, so σ is a solution for $t[r]_p \Rightarrow t'$, and $t[r]_p\sigma \rightarrow_{R\cup E,A} t'\sigma$. As $t\sigma \rightarrow^1_{E,A} t[r]_p\sigma$, then $t\sigma \rightarrow_{R\cup E,A} t'\sigma$.

□

Theorem 7 (Weak Completeness of the Calculus for Reachability). *Given a narrowable rewrite theory $\mathcal{R} = (\Sigma, \mathcal{E}, R)$, its FPP executable* Mel *theory $(\Sigma, E \cup A)$, the associated rewrite theory $\mathcal{R}_E = (\Sigma', A, R_E)$, and a reachability goal G, if σ is an idempotent $R \cup E, A$-normalized solution for G, using the transformed set of rules $\tilde{R}$, then there is an idempotent E, A-normalized substitution γ, with $\sigma \ll_A \gamma_{\mathit{Var}(G)}$, such that $G\!\downarrow \leadsto^*_\gamma \Box$.*

Proof. Every computed answer γ is idempotent E, A-normalized by definition of the calculus. We prove the theorem using induction on the number of reachability subgoals plus the number of $\Rightarrow^1_N$ rewrite steps, including the subgoals and rewrite steps due to conditions. The proof is exactly the same already shown in Theorem 5 (as we have $\Rightarrow^1_N$ rewriting we also have $\rightarrow^1_N$ rewriting), where there are only two cases left to prove.

- The first case is the base case with one subgoal and zero $\Rightarrow^1_N$ rewrite steps. Then $t\sigma =_{\mathcal{E}} t'\sigma$, so σ is a solution for the sentence $t = t'$ and also for the unification goal $eq(t, t') \rightarrow tt$. There are two subcases.
  - If $t\!\downarrow =_{\mathcal{E}} t'\!\downarrow$, then $t\!\downarrow \Rightarrow t'\!\downarrow \leadsto_{[x]} tt \rightarrow tt \leadsto_{[e]} \Box$, and $\gamma = id$, so $\sigma \ll_A \gamma_{\mathit{Var}(G)}$.
  - If $t\!\downarrow \neq_{\mathcal{E}} t'\!\downarrow$, then $t\!\downarrow \Rightarrow t'\!\downarrow \leadsto_{[x]} eq(t\!\downarrow, t'\!\downarrow) \rightarrow tt$, but we have already proved in Theorem 5 that if σ is an $\mathcal{E}$-solution for this unification problem then there exists a substitution γ, with $\sigma \ll_A \gamma_{\mathit{Var}(G)}$, such that $eq(t\!\downarrow, t'\!\downarrow) \rightarrow tt \leadsto^*_\gamma \Box$ and $\sigma \ll_A \gamma_{\mathit{Var}(G)}$.
- The second case is the induction subcase for one subgoal and at least one $\Rightarrow^1_N$ rewrite step, where we apply a rule $c \equiv l \rightarrow r \ \mathit{if} \ C$ in R to $t\sigma$. As σ is a solution for G then, by Lemma 7, σ is an N-solution for $G\!\downarrow$. Then $G\!\downarrow \equiv t\!\downarrow \Rightarrow t'\!\downarrow$, and $t\!\downarrow\!\sigma \Rightarrow^1_N (t\!\downarrow\!\sigma)[r\sigma'_c]_p \Rightarrow_N t'\!\downarrow\!\sigma$ because $(t\!\downarrow\!\sigma)|_p =_A l\sigma'_c$ and σ'_c is a solution for C (E, A-normalized with respect to $\mathit{Extra}(c)$). As σ is $R \cup E, A$ normalized and $l \notin \mathcal{X}$, then we cannot rewrite inside a position instantiated by σ or a variable position, so p must be an already existing non variable position in $t\!\downarrow$ (i.e., $p \in \mathit{Pos}_\Sigma(t\!\downarrow)$). Also $(t\!\downarrow\!\sigma)[r\sigma'_c]_p \equiv t\!\downarrow\![r\sigma'_c]_p\sigma$ because $\mathit{Dom}(\sigma) \cap \mathit{Var}(r\sigma'_c) = \emptyset$, and $t\!\downarrow\![r\sigma'_c]_p\sigma \equiv t\!\downarrow\![r]_p(\sigma \cup \sigma'_c)$ because $\mathit{Dom}(\sigma'_c) \cap \mathit{Var}(t\!\downarrow) = \emptyset$ and $\mathit{Dom}(\sigma) \cap \mathit{Dom}(\sigma'_c)) = \emptyset$.

  Then, $t\!\downarrow\!\sigma \Rightarrow^1_N t\!\downarrow\![r]_p(\sigma \cup \sigma'_c) \Rightarrow_N t'\!\downarrow\!\sigma$, and there exists $\gamma = \gamma_v \cup \gamma_w \in \mathit{CSU}_A(t\!\downarrow\!|_p = l)$, with $v = \mathit{Var}(G)$, and $w = \mathit{Var}(c)$, such that $t\!\downarrow\!|_p\gamma_v =_A l\gamma_w$. Then there exists $\gamma \equiv \gamma_v \cup \gamma_w \in \mathit{CSU}_A(t\!\downarrow\!|_p\!\downarrow = l)$ such that $t\!\downarrow\!|_p\gamma_v =_A l\gamma_w$, and $\sigma \cup \sigma'_c \ll_A \gamma$, so $\sigma \cup \sigma'_c =_A \gamma\rho$ for some

idempotent substitution ρ. σ is E, A-normalized, σ'_c is E, A-normalized except maybe for some subset of $Dom(\gamma)$, so ρ must be E, A-normalized.
Then $t{\downarrow} \Rightarrow t'{\downarrow} \rightsquigarrow_{[t]} t{\downarrow} \Rightarrow^1 x_k, x_k \Rightarrow t'{\downarrow} \rightsquigarrow^*_{[c]} t{\downarrow}|_p \Rightarrow^1 y_{k'}, t{\downarrow}[y_{k'}]_p \Rightarrow t'{\downarrow} \rightsquigarrow_{[r],\gamma} (C\gamma){\downarrow} \wedge (t{\downarrow}[r]_p\gamma){\downarrow} \Rightarrow t'{\downarrow}$. Let's call $u = \mathit{Var}(C\gamma \wedge t{\downarrow}[r]_p\gamma)$. ρ is a solution for $C\gamma \wedge t{\downarrow}[r]_p\gamma \to (t'{\downarrow}\gamma){\downarrow}$ (recall that $(t'{\downarrow}\gamma){\downarrow} \equiv (t'\gamma){\downarrow}$) with one less subgoal and one less rewriting step than the solution σ for $t \Rightarrow t'{\downarrow}$, so I.H. applies and there exist an E, A-normalized substitution θ such that $\rho \ll_A \theta_u$ and $(C\gamma){\downarrow} \wedge (t{\downarrow}[r]_p\gamma){\downarrow} \Rightarrow (t'\gamma){\downarrow} \rightsquigarrow^*_\theta \Box$.
As $\rho \ll_A \theta_u$ then $\rho \ll_A \theta_{\mathit{Var}(C\gamma)}$, so $\rho_{\mathit{Var}(C\gamma)} \ll_A \theta_{\mathit{Var}(C\gamma)}$. $\sigma \cup \sigma'_c =_A \gamma\rho$, so $\sigma = (\sigma \cup \sigma'_c)_v = (\gamma\rho)_v$. Then we have $\sigma = \gamma_v\rho_{\mathit{Ran}(\gamma_v)} \cup \rho_v$. As $Dom(\gamma_v) \cap Dom(\rho_v) = \emptyset$ then $\gamma_v\rho_{\mathit{Ran}(\gamma_v)} \cup \rho_v = \gamma_v(\rho_{\mathit{Ran}(\gamma_v)} \cup \rho_v)$, and $\sigma = \gamma_v(\rho_{\mathit{Ran}(\gamma_v)} \cup \rho_v) = \gamma_v\rho_{\mathit{Var}(C\gamma)} \ll_A \gamma_v\theta_{\mathit{Var}(C\gamma)} = \gamma_v(\theta_{\mathit{Ran}(\gamma_v)}) \cup \theta_v)$. As $Dom(\gamma_v) \cap Dom(\theta_v) = \emptyset$, then $\gamma_v(\rho_{\mathit{Ran}(\gamma_v)} \cup \rho_v) = \gamma_v\rho_{\mathit{Ran}(\gamma_v)} \cup \rho_v$, so $\sigma \ll_A \gamma_v\rho_{\mathit{Ran}(\gamma_v)} \cup \rho_v = (\gamma\rho)_v$.

□

7 Example

We show an application of our calculus using the concurrent specification example. This is an excerpt of the Maude specification for the example:

```
mod CONCUR is
sorts User UserSet Tool ToolBox Nat Boolean State .
subsorts User < UserSet . subsorts Tool < ToolBox .

ops u1 u2 u3 : -> User . op emptyU : -> UserSet .
ops t1 t2 t3 : -> Tool . op emptyT : -> ToolBox .
op 0 : -> Nat . op s : Nat -> Nat .
op ok : -> Boolean . op init : -> State .
op _,_ : [UserSet] [UserSet] -> [UserSet] [comm assoc id: emptyU] .
op _;_ : ToolBox ToolBox -> ToolBox [comm assoc id: emptyT] .
op _|_|_|_ : UserSet UserSet ToolBox ToolBox -> [State] .
op count : ToolBox -> Nat .
op _<_ : Nat Nat -> Boolean .

vars M N : Nat .
vars U U’ : User . vars US US’ : UserSet .
vars T T’ : Tool . vars TB TB’ : ToolBox .

mb u1, u2 : UserSet .
...
mb u1, u2, u3 : UserSet .
cmb US | emptyU | TB | TB’ : State
    if count(TB ; TB’) < s(s(s(s(s(0))))) = ok [label M1] .
cmb US | U | TB | TB’ : State
    if U, US : UserSet /\ count(TB ; TB’) < s(s(s(0))) = ok .
cmb US | U, U’ | emptyT | emptyT : State if U, U’, US : UserSet .

eq count(emptyT) = 0 [label E1] .
eq count(T ; TB) = s(count(TB)) [label E2] .
eq 0 < s(N) = ok . eq s(M) < s(N) = M < N .

crl init => US | emptyU | TB | emptyT
    if US | emptyU | TB | emptyT : State [label R1 nonexec] .
crl US | US' | T ; T' ; TB | emptyT => US | US' | TB | T ; T'
    if US | US' | TB | T ; T' : State [label R2 nonexec] .
...
endm
```

We will abbreviate $\mathtt{emptyT}$ to ϵ_t and $\mathtt{emptyU}$ to ϵ_u, and consider the reachability goal $G \equiv \mathtt{init} \Rightarrow x_{\mathtt{u}} \mid y_{\mathtt{us}} \mid z^1_{\mathtt{t}}; z^2_{\mathtt{t}} \mid z^1_{\mathtt{t}}; z^2_{\mathtt{t}}$, where from the initial $\mathtt{State}$ $\mathtt{init}$, we want to reach a $\mathtt{[State]}$ with one waiting $\mathtt{User}$, two $\mathtt{Tools}$ in the $\mathtt{ToolBox}$, and the same two $\mathtt{Tools}$ in the workbench. The reachability goal is already normalized. We also abbreviate $x_{\mathtt{u}} \mid y_{\mathtt{us}} \mid z^1_{\mathtt{t}}; z^2_{\mathtt{t}} \mid z^1_{\mathtt{t}}; z^2_{\mathtt{t}}$ to F. Recall that there is a sort *Truth* (T) with constant tt in the associated rewrite theory. The label of the used sentences in each reduction or rewrite calculus step can be found between square brackets at the end of each sentence.

1. $\mathtt{init} \Rightarrow F \rightsquigarrow_{[t]}$

   Rule transitivity is always needed before an application of rule rewrite.
2. $\mathtt{init} \Rightarrow^1 x^1_{[\mathsf{s}]}, x^1_{[\mathsf{s}]} \Rightarrow F \rightsquigarrow_{[w],\mathtt{R1}}$

   Rule rewrite is applied with the transformed rule for $\mathtt{R1}$, where the membership condition has now form $x^2_{\mathtt{us}} \mid \epsilon_u \mid x^3_{\mathtt{tb}} \mid \epsilon_t : s \to tt$. We apply the transformation for unification with memberships and compute $\rho_0 = id$ and $C_0 = \emptyset$ because $\mathtt{init}$ has no variables.
3. $x^2_{\mathtt{us}} \mid \epsilon_u \mid x^3_{\mathtt{tb}} \mid \epsilon_t : s \to tt \wedge x^2_{\mathtt{us}} \mid \epsilon_u \mid x^3_{\mathtt{tb}} \mid \epsilon_t \Rightarrow F \rightsquigarrow_{[t]} \rightsquigarrow_{[r],\mathtt{M1},\sigma_1}$

   We apply rule transitivity again, followed by rule reduction with the fresh rule $x^4_{\mathtt{us}} \mid \epsilon_u \mid x^5_{\mathtt{tb}} \mid x^6_{\mathtt{tb}} : s \to tt$ *if* $eq(\mathtt{count}(x^5_{\mathtt{tb}}; x^6_{\mathtt{tb}}) < \mathtt{s(s(s(s(s(0))))),ok}) \to tt$ associated to the conditional membership $\mathtt{M1}$. We omit the result of the transitivity step. We compute $\rho_1 = \{x^2_{\mathtt{us}} \mapsto x^2_{[\mathsf{us}]}, x^3_{\mathtt{tb}} \mapsto x^3_{[\mathsf{tb}]}, x^4_{\mathtt{us}} \mapsto x^4_{[\mathsf{us}]}, x^5_{\mathtt{tb}} \mapsto x^5_{[\mathsf{tb}]}, x^6_{\mathtt{tb}} \mapsto x^6_{[\mathsf{tb}]}\}$ and $C_1 = \{x^2_{[\mathsf{us}]} : \mathtt{us} \wedge x^3_{[\mathsf{tb}]} : \mathtt{tb} \wedge x^4_{[\mathsf{us}]} : \mathtt{us} \wedge x^5_{[\mathsf{tb}]} : \mathtt{tb} \wedge x^6_{[\mathsf{tb}]} : \mathtt{tb}\}$. Instead of solving the unification problem and use the obtained unifier, we apply ρ_1 to the whole reachability problem and add the condition associated to C_1 in $\mathcal{R}_E$ in front of the reachability problem, which is an equivalent approach for leftmost narrowing, because in this way we must solve the unification problem before we can continue with the reachability problem. The obtained unifier is $\sigma_1 = \{x^2_{[\mathsf{us}]} \mapsto x^7_{[\mathsf{us}]}, x^3_{[\mathsf{tb}]} \mapsto x^8_{[\mathsf{tb}]}, x^4_{[\mathsf{us}]} \mapsto x^7_{[\mathsf{us}]}, x^5_{[\mathsf{tb}]} \mapsto x^8_{[\mathsf{tb}]}, x^6_{[\mathsf{tb}]} \mapsto \epsilon_t\}$. The condition $x^6_{[\mathsf{tb}]} : \mathtt{tb} \to tt$ becomes $\epsilon_t : \mathtt{tb} \to tt$ which after canonization is $tt \to tt$. Also $\mathtt{count}(x^5_{\mathtt{tb}}; x^6_{\mathtt{tb}})$ becomes $\mathtt{count}(x^8_{[\mathsf{tb}]}; \epsilon_t)$, and then it becomes $\mathtt{count}(x^8_{[\mathsf{tb}]})$ after canonization.
4. $x^7_{[\mathsf{us}]} : \mathtt{us} \to tt \wedge x^8_{[\mathsf{tb}]} : \mathtt{tb} \to tt \wedge x^7_{[\mathsf{us}]} : \mathtt{us} \to tt \wedge x^8_{[\mathsf{tb}]} : \mathtt{tb} \to tt \wedge tt \to tt \wedge$
   $eq(\mathtt{count}(x^8_{[\mathsf{tb}]}) < \mathtt{s(s(s(s(s(0))))),ok}) \to tt \wedge x^7_{[\mathsf{us}]} \mid \epsilon_u \mid x^8_{[\mathsf{tb}]} \mid \epsilon_t \Rightarrow F \rightsquigarrow_{[m],\{x^7_{[\mathsf{us}]} \mapsto x^7_{\mathtt{us}}\}}$
5. $x^8_{[\mathsf{tb}]} : \mathtt{tb} \to tt \wedge tt \to tt \wedge x^8_{[\mathsf{tb}]} : \mathtt{tb} \to tt \wedge tt \to tt \wedge$
   $eq(\mathtt{count}(x^8_{[\mathsf{tb}]}) < \mathtt{s(s(s(s(s(0))))),ok}) \to tt \wedge x^7_{\mathtt{us}} \mid \epsilon_u \mid x^8_{[\mathsf{tb}]} \mid \epsilon_t \Rightarrow F \rightsquigarrow_{[m],\{x^8_{[\mathsf{tb}]} \mapsto x^8_{\mathtt{tb}}\}}$
6. $tt \to tt \wedge tt \to tt \wedge tt \to tt \wedge eq(\mathtt{count}(x^8_{\mathtt{tb}}) < \mathtt{s(s(s(s(s(0))))),ok}) \to tt \wedge$
   $x^7_{\mathtt{us}} \mid \epsilon_u \mid x^8_{\mathtt{tb}} \mid \epsilon_t \Rightarrow F \rightsquigarrow_{[e]} \rightsquigarrow_{[e]} \rightsquigarrow_{[e]}$

   Rule elimination removes the trivial subgoals. We have finished solving the unification problem, with unifier $\{x^2_{\mathtt{us}} \mapsto x^7_{\mathtt{us}}, x^3_{\mathtt{tb}} \mapsto x^8_{\mathtt{tb}}, x^4_{\mathtt{us}} \mapsto x^7_{\mathtt{us}}, x^5_{\mathtt{tb}} \mapsto x^8_{\mathtt{tb}}, x^6_{\mathtt{tb}} \mapsto \epsilon_t\}$, so we continue with the reachability problem. We apply rules transitivity, congruence, and reduction several times using the rules associated to equations E1 and E2. As the involved sorts are in $OS(S)$ we can use the order-sorted unification algorithms without any transformation.

7. $\mathit{eq}(\mathtt{count}(x^8_{\mathtt{tb}}) < \mathtt{s(s(s(s(s(0)))))}, \mathtt{ok}) \to \mathit{tt} \wedge x^7_{\mathtt{us}} \mid \epsilon_u \mid x^8_{\mathtt{tb}} \mid \epsilon_t \Rightarrow F \rightsquigarrow_{[t]}$
8. $\mathit{eq}(\mathtt{count}(x^8_{\mathtt{tb}}) < \mathtt{s(s(s(s(s(0)))))}, \mathtt{ok}) \to^1 x^9_{[T]}, x^9_{[T]} \to \mathit{tt} \wedge x^7_{\mathtt{us}} \mid \epsilon_u \mid x^8_{\mathtt{tb}} \mid \epsilon_t \Rightarrow F \rightsquigarrow_{[c]}$
9. $\mathtt{count}(x^8_{\mathtt{tb}}) < \mathtt{s(s(s(s(s(0)))))} \to^1 x^{10}_{[b]}, \mathit{eq}(x^{10}_{[b]}, \mathtt{ok}) \to \mathit{tt} \wedge x^7_{\mathtt{us}} \mid \epsilon_u \mid x^8_{\mathtt{tb}} \mid \epsilon_t \Rightarrow F \rightsquigarrow_{[c]}$
10. $\mathtt{count}(x^8_{\mathtt{tb}}) \to^1 x^{11}_{[\mathtt{n}]}, \mathit{eq}(x^{11}_{[\mathtt{n}]} < \mathtt{s(s(s(s(s(0)))))}, \mathtt{ok}) \to \mathit{tt} \wedge x^7_{\mathtt{us}} \mid \epsilon_u \mid x^8_{\mathtt{tb}} \mid \epsilon_t \Rightarrow F \rightsquigarrow_{[r],\mathtt{E2},\sigma_2}$
    Rule reduction is applied with equation E2 and unifier $\sigma_2 = \{x^8_{\mathtt{tb}} \mapsto x^{12}_{\mathtt{t}}; x^{13}_{\mathtt{tb}}\}$.
11. $\mathit{eq}(\mathtt{s}(\mathtt{count}(x^{13}_{\mathtt{tb}})) < \mathtt{s(s(s(s(s(0)))))}, \mathtt{ok}) \to \mathit{tt} \wedge x^7_{\mathtt{us}} \mid \epsilon_u \mid x^{12}_{\mathtt{t}}; x^{13}_{\mathtt{tb}} \mid \epsilon_t \Rightarrow F \rightsquigarrow_{[t]}$
    We omit some steps here ...
12. $\mathit{eq}(\mathtt{s}(\mathtt{s}(\mathtt{s}(\mathtt{s}((\mathtt{count}(x^{17}_{\mathtt{tb}})))))) < \mathtt{s(s(s(s(s(0)))))}, \mathtt{ok}) \to \mathit{tt} \wedge$
    $x^7_{\mathtt{us}} \mid \epsilon_u \mid x^{12}_{\mathtt{t}}; x^{14}_{\mathtt{t}}; x^{15}_{\mathtt{t}}; x^{16}_{\mathtt{t}}; x^{17}_{\mathtt{tb}} \mid \epsilon_t \Rightarrow F \rightsquigarrow_{[t]}$
    We also omit some steps here ...
13. $\mathtt{count}(x^{18}_{\mathtt{tb}}) \to^1 x^{19}_{[\mathtt{n}]}, \mathit{eq}_{[\mathtt{b}]}(\mathtt{s}(\mathtt{s}(\mathtt{s}(\mathtt{s}((x^{19}_{[\mathtt{n}]}))))) < \mathtt{s(s(s(s(s(0)))))}, \mathtt{ok}) \to \mathit{tt} \wedge$
    $x^7_{\mathtt{us}} \mid \epsilon_u \mid x^{12}_{\mathtt{t}}; x^{14}_{\mathtt{t}}; x^{15}_{\mathtt{t}}; x^{16}_{\mathtt{t}}; x^{18}_{\mathtt{tb}} \mid \epsilon_t \Rightarrow F \rightsquigarrow_{[r],\mathtt{E1},\sigma_3}$
    Now $\sigma_3 = \{x^{18}_{\mathtt{tb}} \mapsto \epsilon_t\}$, so $x^{19}_{[\mathtt{n}]} \mapsto 0$. Observe the great simplification of the reachability problem after canonization.
14. $\mathit{tt} \to \mathit{tt} \wedge x^7_{\mathtt{us}} \mid \epsilon_u \mid x^{12}_{\mathtt{t}}; x^{14}_{\mathtt{t}}; x^{15}_{\mathtt{t}}; x^{16}_{\mathtt{t}} \mid \epsilon_t \Rightarrow F \rightsquigarrow_{[e]}$
15. $x^7_{\mathtt{us}} \mid \epsilon_u \mid x^{12}_{\mathtt{t}}; x^{14}_{\mathtt{t}}; x^{15}_{\mathtt{t}}; x^{16}_{\mathtt{t}} \mid \epsilon_t \Rightarrow F \rightsquigarrow_{[t]}$
16. $x^7_{\mathtt{us}} \mid \epsilon_u \mid x^{12}_{\mathtt{t}}; x^{14}_{\mathtt{t}}; x^{15}_{\mathtt{t}}; x^{16}_{\mathtt{t}} \mid \epsilon_t \Rightarrow^1 x^{20}_{[\mathtt{s}]}, x^{20}_{[\mathtt{s}]} \Rightarrow F \rightsquigarrow_{[w],\mathtt{R2}}$
    In order to apply rule rewrite with rule R2 we compute ρ_2 and C_2, as in step 3, and use them to obtain an A-unifier. We skip these steps, and show the resulting reachability problem, where $x^{12}_{\mathtt{t}}$ and $x^{14}_{\mathtt{t}}$ have been moved to the workbench. Observe that the condition in R2 is trivial after substitution and canonization.
17. $\mathit{tt} \to \mathit{tt} \wedge x^7_{\mathtt{us}} \mid \epsilon_u \mid x^{15}_{\mathtt{t}}; x^{16}_{\mathtt{t}} \mid x^{12}_{\mathtt{t}}; x^{14}_{\mathtt{t}} \Rightarrow F \rightsquigarrow_{[e]}$
18. $x^7_{\mathtt{us}} \mid \epsilon_u \mid x^{15}_{\mathtt{t}}; x^{16}_{\mathtt{t}} \mid x^{12}_{\mathtt{t}}; x^{14}_{\mathtt{t}} \Rightarrow x_{\mathtt{u}} \mid y_{\mathtt{us}} \mid z^1_{\mathtt{t}}; z^2_{\mathtt{t}} \mid z^1_{\mathtt{t}}; z^2_{\mathtt{t}} \rightsquigarrow_{[x]}$
    Now we remove the ⇒ symbol, unifying both terms.
19. $\mathit{eq}(x^7_{\mathtt{us}} \mid \epsilon_u \mid x^{15}_{\mathtt{t}}; x^{16}_{\mathtt{t}} \mid x^{12}_{\mathtt{t}}; x^{14}_{\mathtt{t}}, x_{\mathtt{u}} \mid y_{\mathtt{us}} \mid z^1_{\mathtt{t}}; z^2_{\mathtt{t}} \mid z^1_{\mathtt{t}}; z^2_{\mathtt{t}}) \to \mathit{tt} \rightsquigarrow_{[u],\sigma_4} \Box$
    Again, the A unifier σ_4 is obtained by previously computing ρ_3, where all sorted variables are replaced with kinded variables and C_3, which forces each kinded variable to have the specific sort that it had before applying ρ_3. As a final result, we get the computed answer $\sigma = \sigma_4|_{Var(G)} = \{x_{\mathtt{u}} \mapsto x^{21}_{\mathtt{u}}, y_{\mathtt{us}} \mapsto \epsilon_u, z^1_{\mathtt{t}} \mapsto x^{22}_{\mathtt{t}}, z^2_{\mathtt{t}} \mapsto x^{23}_{\mathtt{t}}\}$

So we have found a very general computed answer for our reachability problem $G \equiv \mathtt{init} \Rightarrow x_{\mathtt{u}} \mid y_{\mathtt{us}} \mid z^1_{\mathtt{t}}; z^2_{\mathtt{t}} \mid z^1_{\mathtt{t}}; z^2_{\mathtt{t}}$, where $x_{\mathtt{u}}$ can be any `user`, $y_{\mathtt{us}}$ must be `emptyU`, and $z^1_{\mathtt{t}}$ and $z^2_{\mathtt{t}}$ can be any `tool`.

8 Related work, conclusions and future work

A classic reference in equational conditional narrowing modulo is the work of Bockmayr [Boc93]. The topic is addressed here for Church-Rosser equational conditional term rewriting systems with empty axioms, but non terminating axioms (like ACU) are not allowed. The intimate relationship between rewriting and reachability problems was shown by Hullot [Hul80], where he proved that any normalized solution to a reachability problem could be lifted to a narrowing derivation that computed a more general solution. The idea of a reduction phase between narrowing steps was already shown by Fribourg in the language SLOG [Fri85]. An inductive proof method for properties of reduction relations has been presented by Gnaedig and H. Kirchner [GK07], where proof trees are generated using narrowing and an abstraction mechanism, and abstraction constraints are used to control narrowing. Non conditional narrowing modulo order-sorted equational logics is covered by Meseguer and Thati [MT07] and it is being used for cryptographic protocol analysis. Feuillade and Genet [FG03] have also studied reachability in term rewriting systems for cryptographic protocol verification. The idea of constraint solving by narrowing in combined algebraic domains was presented by H. Kirchner and Ringeissen [KR94], where the supported theories had unconstrained equalities and the rewrite rules had constraints from an algebraic built-in structure. Equivalence of $R/\mathcal{E}$ and $R \cup E, A$ rewriting was proved by Viry [Vir94] for unsorted rewrite theories. Membership equational logic was defined by Meseguer [Mes97]. Comon studied the completion of rewrite systems with membership constraints [Com98a, Com98b]. A rewrite system for Mel theories that allows unification by rewriting is presented by Durán et al. [DLM$^+$08a]. Strategies, which also play a main role in narrowing, have been studied by Antoy, Echahed and Hanus [AEH94]. Their needed narrowing strategy, for inductively sequential rewrite systems, generates only narrowing steps leading to a computed answer. Recently Escobar, Sasse, and Meseguer [ESM12] have developed the concepts of variant and folding variant, a narrowing strategy for order-sorted unconditional rewrite theories that terminates on those theories having the *finite variant property.* C. Kirchner, H. Kirchner, and F. Nahon [KKN13] have developed a narrowing-based proof search method for inductive theorems in a deduction modulo framework. Foundations for order-sorted conditional rewriting have been published by Meseguer [Mes14]. Cholewa, Escobar, and Meseguer [CEM14] have defined a new hierarchical method, called layered constraint narrowing, to solve narrowing problems in order-sorted conditional equational theories and given new theoretical results on that matter, including the definition of constrained variants for order-sorted conditional rewrite theories. Order-sorted conditional narrowing with constraint solvers has been addressed by Rocha et al. [RMM14].

In this work we have presented a new definition of $\to^1_{R,A}$ and $R \cup E, A$-rewriting, a definition of a new concept of narrowable rewrite theory, and developed two narrowing calculi for unification in membership equational logic and reachability in narrowable rewrite theories, with the following characteristics, to the best of our knowledge:

- a larger class of rewrite theories is accepted by the calculus with respect to previous work, admitting extra variables with no restrictions in equational, membership or rewrite conditions.
- also a larger class of reachability goals is admitted for solving, compared to previous work,
- both calculi use a leftmost strategy,
- both calculi follow a strategy, consisting in applying a calculus rule only if the composition of all computed substitutions remains normalized with respect to all extra variables and all the variables in the initial problem,
- both calculi follow a strategy consisting in normalizing all terms before each narrowing step,
- the calculus for reachability follows a strategy consisting in applying narrowing to a subterm with some substitution only if the whole term remains normalized when instantiated with the same substitution,
- the calculus for reachability follows a strategy consisting in keeping a list of reachability problems. Initially the list holds the original problem. Each new reachability problem gener-

ated by the calculus is checked against the current list. If the problem is a renaming and/or reordering of any element in the list, it gets discarded,

- both calculi are sound and weakly complete, i.e., complete with respect to idempotent normalized answers.

Previous work for executable rewrite theories, which used non-normalized terms and substitutions, and where a strategy for applying the calculi was shown, was implemented using Maude. The implementation, together with some examples and instructions for its use, is available at http://maude.sip.ucm.es/cnarrowing/. This new version of the calculi has not been implemented yet, but we plan to make use of it in our current line of investigation, that concerns the extension of the calculi to handle constraints and their connection with external constraint solvers for domains such as finite domains, integers, Boolean values, etc., that could greatly improve the performance of any implementation.

Our future work will focus on the use of constraint solvers on the parts of a condition that have a suitable domain, which will exclude the use of any other type of narrowing or unification algorithm on these parts of the condition. As we are performing symbolic analysis of the state space, we only need to ensure feasibility of the condition, instead of solving it using narrowing, to go on with our narrowing derivation. Finally, if we find a *narrowing path* from $t(\bar{x})$ to $t'(\bar{x})$, displayed as $t(\bar{x}) \leadsto^{*}_{R,\mathcal{E}} t'(\bar{x})$, where we instantiate part or all the variables in $\bar{x}$, the accumulated condition must be feasible for this instantiation of variables. The answer will consist then of a substitution and a set of constraints that can be, if needed, instantiated to actual solutions (for instance, to serve as a counterexample). The use of constraint solvers on conditions can greatly reduce the inherent risk of state explosion, always preexisting in conditional rewriting and narrowing, even turning an infinite state space for a narrowing problem without constraint solvers into a finite one. We plan to use the rewriting language Maude [CDE$^+$07] which it is currently being extended to allow for the use of constraint solvers, so it will support all the features needed in our work, as a framework where we shall develop our prototypes.

References

[AEH94] Sergio Antoy, Rachid Echahed, and Michael Hanus. A needed narrowing strategy. In Hans-Juergen Boehm, Bernard Lang, and Daniel M. Yellin, editors, *Conference Record of POPL'94: 21st ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, Portland, Oregon, USA, January 17-21, 1994*, pages 268–279. ACM Press, 1994.

[AMPP14] Luis Aguirre, Narciso Martí-Oliet, Miguel Palomino, and Isabel Pita. Conditional narrowing modulo in rewriting logic and Maude. In Escobar [Esc14], pages 80–96.

[BJ87] George Boolos and Richard C. Jeffrey. *Computability and logic (2. ed.)*. Cambridge University Press, 1987.

[BM06] Roberto Bruni and José Meseguer. Semantic foundations for generalized rewrite theories. *Theoretical Computer Science*, 360(1-3):386–414, 2006.

[BM12] Kyungmin Bae and José Meseguer. Model checking LTLR formulas under localized fairness. In Francisco Durán, editor, *Rewriting Logic and Its Applications - 9th International Workshop, WRLA 2012, Held as a Satellite Event of ETAPS, Tallinn, Estonia, March 24-25, 2012, Revised Selected Papers*, volume 7571 of *Lecture Notes in Computer Science*, pages 99–117. Springer, 2012.

[BM14] Kyungmin Bae and José Meseguer. Infinite-state model checking of LTLR formulas using narrowing. In Escobar [Esc14], pages 113–129.

[Boc93] Alexander Bockmayr. Conditional narrowing modulo a set of equations. *Applicable Algebra in Engineering, Communication and Computing*, 4:147–168, 1993.

[CDE$^+$02] Manuel Clavel, Francisco Durán, Steven Eker, Patrick Lincoln, Narciso Martí-Oliet, José Meseguer, and José F. Quesada. Maude: Specification and programming in rewriting logic. *Theoretical Computer Science*, 285(2):187–243, 2002.

[CDE$^+$07] Manuel Clavel, Francisco Durán, Steven Eker, Patrick Lincoln, Narciso Martí-Oliet, José Meseguer, and Carolyn Talcott. *All About Maude - A High-Performance Logical Framework: How to Specify, Program, and Verify Systems in Rewriting Logic*, volume 4350 of *Lecture Notes in Computer Science*. Springer, 2007.

[CDH$^+$07] Manuel Clavel, Francisco Durán, Joe Hendrix, Salvador Lucas, José Meseguer, and Peter Csaba Ölveczky. The maude formal tool environment. In Till Mossakowski, Ugo Montanari, and Magne Haveraaen, editors, *Algebra and Coalgebra in Computer Science, Second International Conference, CALCO 2007, Bergen, Norway, August 20-24, 2007, Proceedings*, volume 4624 of *Lecture Notes in Computer Science*, pages 173–178. Springer, 2007.

[CEM14] Andrew Cholewa, Santiago Escobar, and José Meseguer. Constrained Narrowing for Conditional Equational Theories Modulo Axioms. Technical report, http://hdl.handle.net/2142/50289, C.S. Department, University of Illinois at Urbana-Champaign, August 2014.

[Com98a] Hubert Comon. Completion of rewrite systems with membership constraints. part I: deduction rules. *Journal of Symbolic Computation*, 25(4):397–419, 1998.

[Com98b] Hubert Comon. Completion of rewrite systems with membership constraints. part II: constraint solving. *Journal of Symbolic Computation*, 25(4):421–453, 1998.

[DLM$^+$08a] Francisco Durán, Salvador Lucas, Claude Marché, José Meseguer, and Xavier Urbain. Proving operational termination of membership equational programs. *Higher-Order and Symbolic Computation*, 21(1-2):59–88, 2008.

[DLM08b] Francisco Durán, Salvador Lucas, and José Meseguer. MTT: The Maude termination tool (system description). In Alessandro Armando, Peter Baumgartner, and Gilles Dowek, editors, *Automated Reasoning, 4th International Joint Conference, IJCAR 2008, Sydney, Australia, August 12-15, 2008, Proceedings*, volume 5195 of *Lecture Notes in Computer Science*, pages 313–319. Springer, 2008.

[DM12a] Francisco Durán and José Meseguer. On the Church-Rosser and coherence properties of conditional order-sorted rewrite theories. *The Journal of Logic and Algebraic Programming*, 81(7-8):816–850, 2012.

[DM12b] Francisco Durán and José Meseguer. On the church-rosser and coherence properties of conditional order-sorted rewrite theories. *Journal of Logic and Algebraic Programming*, 81(7-8):816–850, 2012.

[EMM09] Santiago Escobar, Catherine Meadows, and José Meseguer. Maude-NPA: Cryptographic protocol analysis modulo equational properties. In Alessandro Aldini, Gilles Barthe, and Roberto Gorrieri, editors, *Foundations of Security Analysis and Design V*, volume 5705 of *Lecture Notes in Computer Science*, pages 1–50. Springer, 2009.

[Esc14] Santiago Escobar, editor. *Rewriting Logic and Its Applications - 10th International Workshop, WRLA 2014, Held as a Satellite Event of ETAPS, Grenoble, France, April 5-6, 2014, Revised Selected Papers*, volume 8663 of *Lecture Notes in Computer Science*. Springer, 2014.

[ESM12] Santiago Escobar, Ralf Sasse, and José Meseguer. Folding variant narrowing and optimal variant termination. *The Journal of Logic and Algebraic Programming*, 81(7-8):898–928, 2012.

[Fay78] M.J. Fay. *First-order Unification in an Equational Theory*. University of California, 1978.

[FG03] Guillaume Feuillade and Thomas Genet. Reachability in conditional term rewriting systems. *Electronic Notes in Theoretical Computer Science*, 86(1):133–146, 2003.

[Fri85] Laurent Fribourg. SLOG: A logic programming language interpreter based on clausal superposition and rewriting. In *Proceedings of the 1985 Symposium on Logic Programming, Boston, Massachusetts, USA, July 15-18, 1985*, pages 172–184. IEEE-CS, 1985.

[GK07] Isabelle Gnaedig and Hélène Kirchner. Narrowing, abstraction and constraints for proving properties of reduction relations. In Hubert Comon-Lundh, Claude Kirchner, and Hélène Kirchner, editors, *Rewriting, Computation and Proof, Essays Dedicated to Jean-Pierre Jouannaud on the Occasion of His 60th Birthday*, volume 4600 of *Lecture Notes in Computer Science*, pages 44–67. Springer, 2007.

[GM86] Elio Giovannetti and Corrado Moiso. A completeness result for e-unification algorithms based on conditional narrowing. In Mauro Boscarol, Luigia Carlucci Aiello, and Giorgio

Levi, editors, *Foundations of Logic and Functional Programming, Workshop, Trento, Italy, December 15-19, 1986, Proceedings*, volume 306 of *Lecture Notes in Computer Science*, pages 157–167. Springer, 1986.

[Ham00] Mohamed Hamada. Strong completeness of a narrowing calculus for conditional rewrite systems with extra variables. *Electronic Notes in Theoretical Computer Science*, 31:89–103, 2000.

[Hul80] Jean-Marie Hullot. Canonical forms and unification. In Wolfgang Bibel and Robert A. Kowalski, editors, *5th Conference on Automated Deduction, Les Arcs, France, July 8-11, 1980, Proceedings*, volume 87 of *Lecture Notes in Computer Science*, pages 318–334. Springer, 1980.

[KKN13] Claude Kirchner, Hélène Kirchner, and Fabrice Nahon. Narrowing based inductive proof search. In Andrei Voronkov and Christoph Weidenbach, editors, *Programming Logics - Essays in Memory of Harald Ganzinger*, volume 7797 of *Lecture Notes in Computer Science*, pages 216–238. Springer, 2013.

[KR94] Hélène Kirchner and Christophe Ringeissen. Constraint solving by narrowing in combined algebraic domains. In Pascal Van Hentenryck, editor, *Logic Programming, Proceedings of the Eleventh International Conference on Logic Programming, Santa Marherita Ligure, Italy, June 13-18, 1994*, pages 617–631. MIT Press, 1994.

[LM09] Salvador Lucas and José Meseguer. Operational termination of membership equational programs: the order-sorted way. *Electronic Notes in Theoretical Computer Science*, 238(3):207–225, 2009.

[LMM05] Salvador Lucas, Claude Marché, and José Meseguer. Operational termination of conditional term rewriting systems. *Information Processing Letters*, 95(4):446–453, 2005.

[Mes90] José Meseguer. Rewriting as a unified model of concurrency. In J.C.M. Baeten and J.W. Klop, editors, *CONCUR '90 Theories of Concurrency: Unification and Extension*, volume 458 of *Lecture Notes in Computer Science*, pages 384–400. Springer, 1990.

[Mes97] José Meseguer. Membership algebra as a logical framework for equational specification. In Francesco Parisi-Presicce, editor, *Recent Trends in Algebraic Development Techniques, 12th International Workshop, WADT'97, Tarquinia, Italy, June 1997, Selected Papers*, volume 1376 of *Lecture Notes in Computer Science*, pages 18–61. Springer, 1997.

[Mes12] José Meseguer. Twenty years of rewriting logic. *Journal of Logic and Algebraic Programming*, 81(7-8):721–781, 2012.

[Mes14] José Meseguer. Strict Coherence of Conditional Rewriting Modulo Axioms. Technical report, http://hdl.handle.net/2142/50288, C.S. Department, University of Illinois at Urbana-Champaign, August 2014.

[MH94] Aart Middeldorp and Erik Hamoen. Completeness results for basic narrowing. *Applicable Algebra in Engineering, Communication and Computing*, 5:213–253, 1994.

[MM02] Narciso Martí-Oliet and José Meseguer. Rewriting logic: roadmap and bibliography. *Theoretical Computer Science*, 285(2):121–154, 2002.

[MT07] José Meseguer and Prasanna Thati. Symbolic reachability analysis using narrowing and its application to verification of cryptographic protocols. *Higher-Order and Symbolic Computation*, 20(1-2):123–160, 2007.

[Plo72] Gordon Plotkin. Building in equational theories. *Machine Intelligence 7*, 1972.

[RMM14] Camilo Rocha, José Meseguer, and César A. Muñoz. Rewriting modulo SMT and open system analysis. In Escobar [Esc14], pages 247–262.

[Vir94] Patrick Viry. Rewriting: An effective model of concurrency. In Constantine Halatsis, Dimitris G. Maritsas, George Philokyprou, and Sergios Theodoridis, editors, *PARLE '94: Parallel Architectures and Languages Europe, 6th International PARLE Conference, Athens, Greece, July 4-8, 1994, Proceedings*, volume 817 of *Lecture Notes in Computer Science*, pages 648–660. Springer, 1994.