


Combined Hierarchical Matching: the Regular Case

Serdar Erbatur 

University of Texas at Dallas, TX, USA

Andrew M. Marshall 

University of Mary Washington, Fredericksburg, VA, USA

Christophe Ringeissen 

Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France

Abstract

Matching algorithms are often central sub-routines in many areas of automated reasoning. They are used in areas such as functional programming, rule-based programming, automated theorem proving, and the symbolic analysis of security protocols. Matching is related to unification but provides a somewhat simplified problem. Thus, in some cases, we can obtain a matching algorithm even if the unification problem is undecidable. In this paper we consider a hierarchical approach to constructing matching algorithms. The hierarchical method has been successful for developing unification algorithms for theories defined over a constructor sub-theory. We show how the approach can be extended to matching problems which allows for the development, in a modular way, of hierarchical matching algorithms. Here we focus on regular theories, where both sides of each equational axiom have the same set of variables. We show that the combination of two hierarchical matching algorithms leads to a hierarchical matching algorithm for the union of regular theories sharing only a common constructor sub-theory.

2012 ACM Subject Classification Theory of computation → Equational logic and rewriting; Theory of computation → Automated reasoning

Keywords and phrases Matching, combination problem, equational theories

Digital Object Identifier 10.4230/LIPIcs.FSCD.2022.6

Acknowledgements We would like to thank the reviewers for their comments that were very helpful to improve the readability of the paper.

1 Introduction

Matching procedures play a central role in automated reasoning and in various declarative programming paradigms such as functional programming or (constraint) logic programming. For example, in rule-based programming [9, 11], matching is needed to apply a rule and thus to perform computations. In automated theorem proving [1, 7], matching is useful to simplify existing facts via contraction inferences. For the verification of security protocols, dedicated provers [8, 20, 25] handle protocols specified in a symbolic way. In these reasoning tools, the capabilities of an intruder are modeled using equational theories, and the reasoning is supported by decision procedures and solvers modulo equational theories, including matching and unification. An equational matching problem is an equational unification problem with free constants where each equation has a ground side. This particular form of equational unification with free constants remains undecidable in general. However, the successful application of equational rewriting in rule-based programming languages [9, 11, 26] has demonstrated the usefulness of developing matching algorithms for particular equational theories such as Associativity (A), Commutativity (C) or Associativity-Commutativity (AC). In many practical applications, the underlying equational theory is defined as a union of theories, like a union of AC -symbols. In that case, it is quite natural to solve the matching



© Serdar Erbatur, Andrew M. Marshall, and Christophe Ringeissen;
licensed under Creative Commons License CC-BY 4.0

7th International Conference on Formal Structures for Computation and Deduction (FSCD 2022).

Editor: Amy P. Felty; Article No. 6; pp. 6:1–6:22

Leibniz International Proceedings in Informatics



LIPIC Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

problem for the union of theories in a modular way by combining the matching algorithms available for the component theories of the union. For unification and matching, there are terminating and complete combination procedures for the union of signature-disjoint theories [34, 3]. These combination procedures can be extended to some non-disjoint unions of theories sharing only constructor symbols, but it is quite difficult to find particular cases where these procedures terminate [12], although several terminating cases have been identified [30, 5, 13, 18, 19]. Here, matching being a restricted form of unification can be helpful since matching can be considered as a simpler problem. For example, A -matching is finitary, that is, the set of solutions of an A -matching problem is finite, whereas A -unification is infinitary. Thus for matching, we may be able to show termination even if we cannot in unification.

In this paper we consider the matching problem in theories $F \cup E$ where E is a constructor sub-theory for $F \cup E$, F being called an E -constructed theory. We show how to apply the relatively recently developed hierarchical combination approach [14, 13, 18, 19] to build $F \cup E$ -matching algorithms. We focus on regular theories for $F \cup E$, where both sides of each equational axiom have the same set of variables. This is a natural assumption since any matching problem has only ground solutions in regular theories. We adopt a new modular definition of E -constructed theory [18] and consider the class of regular theories $F \cup E$ where F is E -constructed. This class is closed by any union of theories $F_1 \cup E$ and $F_2 \cup E$ sharing only symbols in E . We show that combining hierarchical matching algorithms known for $F_1 \cup E$ and $F_2 \cup E$ leads to a hierarchical matching algorithm for the union of $F_1 \cup E$ and $F_2 \cup E$. In our hierarchical matching approach, we consider a new type of layer-reduced term mappings that can be constructed in a modular way to reduce the theory layers of any ground term occurring in a matching problem. In addition, we also show how the hierarchical approach can be used for solving the $F \cup E$ -equality of terms in layer-reduced form, required by a hierarchical $F \cup E$ -matching algorithm. The presented hierarchical approach applies to the important case $R \cup E$ where (R, E) is any E -convergent term rewrite system (TRS for short) where all the symbols in E are constructors, called E -constructed TRS. It applies also to theories $F \cup E$ where F is E -constructed and $F \cup E$ is a finite syntactic theory [28, 23]. In that case the underlying hierarchical algorithm can be simply expressed using some additional mutation rules generalizing the very classical decomposition rule used in syntactic unification. A form of syntacticity can also be applied to E -constructed TRSs which are *innermost-resolvent*, exemplified by distributive theories and exponentiation theories.

Motivating Example from Security Protocols. Modular exponentiation is a common operation found in many theories modeling security protocols [24]. For example, exponentiation with a multiplication operator can be modeled with the following axioms $\{e(e(x, y), z) = e(x, y * z), e(x * y, z) = e(x, z) * e(y, z)\}$ and the AC theory for $*$. Obtaining unification algorithms for this exponentiation theory (and related theories) has proven difficult. In fact, it is undecidable in the case where $*$ is AC [27]. Because of this difficulty, the theory is often changed to include a new operator \otimes , and a modification of the first axiom to $e(e(x, y), z) = e(x, y \otimes z)$. Thus, creating two multiplication operators rather than one. Even in this case, obtaining a unification algorithm is not always possible with several undecidability results having been shown depending on the properties of $*$ and \otimes [22]. However, by using the modular combination result developed in this paper, we can obtain a hierarchical matching algorithm for the exponentiation theories and more. The modular aspect to the combination algorithm is also attractive since we can reuse a matching algorithm for the base theory, AC in this example.

Outline. After this introduction and the next section on preliminaries, the paper is organized as follows. Sections 3 and 4 present the different classes of theories $F \cup E$ considered in the paper, and some modularity results we can obtain for the problems of $F \cup E$ -equality and of $F \cup E$ -matching. The class of E -constructed theories is introduced in Section 3, while Section 4 focuses on E -constructed theories admitting mutation-based matching algorithms. In Section 5, we present our notion of hierarchical $F \cup E$ -matching algorithm. Our results on combining hierarchical $F \cup E$ -matching algorithms are shown in Section 6. In addition, Section 7 shows that our methodology can be applied to get hierarchical decision procedures for the $F \cup E$ -equality. Related work and concluding remarks are discussed in Section 8. Appendix A includes omitted proofs.

2 Preliminaries

We use the standard notation of equational unification [4] and term rewriting systems [2]. Given a first-order signature Σ and a (countable) set of variables V , the set of Σ -terms over variables V is denoted by $T(\Sigma, V)$. Given a (countable) set of constants C disjoint from V and Σ , the set of Σ -terms over $V \cup C$ is denoted in the same way by $T(\Sigma, V \cup C)$. In the following, a Σ -term is assumed to be a term in $T(\Sigma, V \cup C)$. The set of variables (resp., constants) from V (resp., C) occurring in a term $t \in T(\Sigma, V \cup C)$ is denoted by $Var(t)$ (resp., $Cst(t)$). A term t is *ground* if $Var(t) = \emptyset$. A $\Sigma \cup C$ -rooted term is a term whose root symbol is in $\Sigma \cup C$. For any position p in a term t (including the root position ϵ), $t(p)$ is the symbol at position p , $t|_p$ is the subterm of t at position p , and $t[u]_p$ is the term t in which $t|_p$ is replaced by u . A substitution is an endomorphism of $T(\Sigma, V \cup C)$ with only finitely many variables not mapped to themselves. A substitution is denoted by $\sigma = \{x_1 \mapsto t_1, \dots, x_m \mapsto t_m\}$, where the domain of σ is $Dom(\sigma) = \{x_1, \dots, x_m\}$ and the range of σ is $Ran(\sigma) = \{t_1, \dots, t_m\}$. Application of a substitution σ to t is written $t\sigma$. Given a subsignature Σ' of Σ , a Σ' -alien subterm of $t \in T(\Sigma, V \cup C)$ is a $\Sigma \setminus \Sigma'$ -rooted subterm of t such that its superterms are Σ' -rooted. When Σ' is clear from the context, a Σ' -alien subterm is called an alien subterm.

Equational Theories. Given a set E of Σ -axioms (i.e., pairs of terms in $T(\Sigma, V)$, denoted by $l = r$), the *equational theory* $=_E$ is the congruence closure of E under the law of substitutivity (by a slight abuse of terminology, E is often called an equational theory). Equivalently, $=_E$ can be defined as the reflexive transitive closure \leftrightarrow_E^* of an equational step \leftrightarrow_E defined as follows: $s \leftrightarrow_E t$ if there exist a position p of s , $l = r$ (or $r = l$) in E , and substitution σ such that $s|_p = l\sigma$ and $t = s[r\sigma]_p$. An axiom $l = r$ is *regular* if $Var(l) = Var(r)$. An axiom $l = r$ is *collapse-free* if l and r are non-variable terms. An equational theory is *regular* (resp., *collapse-free*) if all its axioms are regular (resp., *collapse-free*). An equational theory E is *finite* if for each term t , there are only finitely many terms s such that $t =_E s$. A theory E is *syntactic* if it has finite *resolvent presentation* S , defined as a finite set of axioms S such that each equality $t =_E u$ has an equational proof $t \leftrightarrow_S^* u$ with at most one equational step \leftrightarrow_S applied at the root position. One can easily check that $C = \{x * y = y * x\}$ (Commutativity) and $AC = \{x * (y * z) = (x * y) * z, x * y = y * x\}$ (Associativity-Commutativity) are regular and collapse-free. Moreover, C and AC are syntactic [23]. A Σ -equation is a pair of Σ -terms denoted by $s =^? t$ or simply $s = t$ when it is clear from the context that we do not refer to an axiom. A *flat* Σ -equation is either an equation between variables or a *non-variable flat* Σ -equation of the form $x_0 = f(x_1, \dots, x_n)$ where x_0, x_1, \dots, x_n are variables and f is a function symbol in Σ . An E -unification problem is a set of Σ -equations, $\Gamma = \{s_1 =^? t_1, \dots, s_n =^? t_n\}$, or equivalently a conjunction of Σ -equations. The set of variables in Γ is denoted by $Var(\Gamma)$.

A solution to Γ , called an *E-unifier*, is a substitution σ such that $s_i\sigma =_E t_i\sigma$ for all $1 \leq i \leq n$. A substitution σ is *more general modulo E* than θ on a set of variables V , denoted as $\sigma \leq_E^V \theta$, if there is a substitution τ such that $x\sigma\tau =_E x\theta$ for all $x \in V$. $\sigma|_V$ denotes the substitution σ restricted to the set of variables V . A *Complete Set of E-Unifiers* of Γ , denoted by $CSU_E(\Gamma)$, is a set of substitutions such that each $\sigma \in CSU_E(\Gamma)$ is an *E-unifier* of Γ , and for each *E-unifier* θ of Γ , there exists $\sigma \in CSU_E(\Gamma)$ such that $\sigma \leq_E^{Var(\Gamma)} \theta$. An *E-unification algorithm* is an algorithm that computes a finite $CSU_E(\Gamma)$ for all *E-unification* problems Γ . An inference rule $\Gamma \vdash \Gamma'$ for *E-unification* is *sound* if each *E-unifier* of Γ' is an *E-unifier* of Γ ; and *complete* if for each *E-unifier* σ of Γ , there exists an *E-unifier* σ' of Γ' such that $\sigma' \leq_E^{Var(\Gamma)} \sigma$. A set of equations $\Gamma = \{x_1 =^? t_1, \dots, x_n =^? t_n\}$ is said to be in *solved form* if each x_i is a variable occurring once in Γ . Given an idempotent substitution $\sigma = \{x_1 \mapsto t_1, \dots, x_n \mapsto t_n\}$ (such that $\sigma\sigma = \sigma$), $\hat{\sigma}$ denotes the corresponding solved form. An inference system for *E-unification* is *sound* if all its inference rules are sound; and *complete* if for each *E-unification* problem Γ on which an inference applies and each *E-unifier* σ of Γ , there exist an *E-unification* problem Γ' inferred from Γ and an *E-unifier* σ' of Γ' such that $\sigma' \leq_E^{Var(\Gamma)} \sigma$. To simplify the notation in our inference rules, we apply them modulo the commutativity of $=^?$ and we often use tuples of terms, such as $\bar{u} = (u_1, \dots, u_n)$, $\bar{v} = (v_1, \dots, v_n)$ to represent the set of equations $\bar{u} =^? \bar{v}$ corresponding to $\{u_1 =^? v_1, \dots, u_n =^? v_n\}$.

Equational Rewrite Relations. Given a signature Σ , an oriented Σ -axiom is called a rewrite rule of the form $l \rightarrow r$ such that $l, r \in T(\Sigma, V)$, l is not a variable and $Var(r) \subseteq Var(l)$. Let R be a set of rewrite rules and E an equational Σ -theory. For any Σ -terms s and t , s *R, E-rewrites* to t , denoted by $s \rightarrow_{R,E} t$, if there exist a position p of s , $l \rightarrow r \in R$, and substitution σ such that $s|_p =_E l\sigma$ and $t = s[r\sigma]_p$. The term s is said to be *R, E-reducible*, $s|_p$ is called a *redex*, and in the particular case where $s|_p = l\sigma$, s *R-rewrites* to t , denoted by $s \rightarrow_R t$. A term is an *innermost redex* if none of its proper subterms is a redex. The symmetric relation $\leftarrow_R \cup \rightarrow_R \cup =_E$ is denoted by $\longleftrightarrow_{R \cup E}$. The rewrite relation $\rightarrow_{R,E}$ is Church-Rosser modulo E if $\longleftrightarrow_{R \cup E}^*$ is included in $\rightarrow_{R,E}^* \circ =_E \circ \leftarrow_{R,E}^*$. The rewrite relation \rightarrow_R is *E-terminating* if $=_E \circ \rightarrow_R \circ =_E$ is terminating. When \rightarrow_R is *E-terminating*, $\rightarrow_{R,E}$ is Church-Rosser modulo E iff $\rightarrow_{R,E}$ is both locally *E-confluent* and locally *E-coherent* [21]. The rewrite relation $\rightarrow_{R,E}$ is *E-convergent* if \rightarrow_R is *E-terminating* and $\rightarrow_{R,E}$ is Church-Rosser modulo E . When $\rightarrow_{R,E}$ is *E-convergent*, we have that for any terms t, t' , $t \longleftrightarrow_{R \cup E}^* t'$ iff $t \downarrow_{R,E} =_E t' \downarrow_{R,E}$, where $t \downarrow_{R,E}$ (resp., $t' \downarrow_{R,E}$) denotes any normal form of t (resp., t') w.r.t $\rightarrow_{R,E}$. A function symbol that does not occur in $\{l(\epsilon) \mid l \rightarrow r \in R\}$ is called a *constructor* for R . Let Σ_0 be the subsignature of Σ that consists of function symbols occurring in the axioms of E . An *E-convergent* rewrite relation $\rightarrow_{R,E}$ is said to be *E-constructed* if all the symbols in Σ_0 are constructors for R . When R is a finite set of rules, the pair (R, E) is called an *equational term rewrite system* (TRS). We say that a property is satisfied by an equational TRS (R, E) if this property is satisfied by $\rightarrow_{R,E}$. Given a TRS (R, E) , $R^=$ denotes the set of equalities $\{l = r \mid l \rightarrow r \in R\}$, and $R^= \cup E$ is the *equational theory* of (R, E) . For sake of brevity, we may use $R \cup E$ instead of $R^= \cup E$. The rewrite relation $\rightarrow_{R,E}$ and all the related notions introduced above for a set R of rules $l \rightarrow r$ such that $l, r \in T(\Sigma, V)$ are extended in a natural way to any set R of ground rules $l \rightarrow r$ such that $l, r \in T(\Sigma, C)$ and for which the condition $Var(r) \subseteq Var(l)$ is trivially satisfied since $Var(l) = Var(r) = \emptyset$.

For any equational Σ -theory F , an *F-canonizer stable by renaming* is an idempotent mapping $w : T(\Sigma, V) \rightarrow T(\Sigma, V)$ such that for any $s, t \in T(\Sigma, V)$, $s =_F t$ iff $w(s) = w(t)$; for any $t \in T(\Sigma, V)$, $Var(w(t)) \subseteq Var(t)$ and for any variable renaming ϕ whose domain is $Var(t)$, $w(t\phi) = w(t)\phi$. For any finite theory E (resp., any *E-convergent* TRS where E is finite), an *E-canonizer* (resp. a *R \cup E-canonizer*) stable by renaming is computable.

3 *E*-Constructed Theories and their Combinations

We introduce a class of *E*-constructed theories including *E*-constructed TRSs. In this paper, an *E*-constructed theory F is an equational theory F such that $F \cup E$ admits a particular normalizing mapping over ground terms to compute a normal form for each equivalence class modulo $=_{F \cup E}$. To get an *E*-constructed theory, the normal forms must satisfy some particular properties. In previous papers [14, 13, 18, 19], these properties were expressed using a reduction ordering on ground terms. Here, we adopt the idea of expressing these properties thanks to a normalizing mapping defined as an idempotent mapping on ground terms generated by a countable infinite set C of free constants totally ordered by a well-founded ordering $>$.

► **Definition 1** ($>$ -compatible renaming). *Assume C is a countable infinite set of constants and $>$ is a well-founded total ordering on C , meaning that there is no infinite decreasing sequence $c_1 > c_2 > \dots$ of elements of C , and for any $c_1, c_2 \in C$, $c_1 > c_2$ or $c_2 > c_1$ or $c_1 = c_2$. A renaming of a finite subset Cst of C is an injective mapping ξ from Cst to C , which is said to be $>$ -compatible if for any $c_1, c_2 \in Cst$, $c_1 > c_2$ iff $c_1\xi > c_2\xi$. Given a signature Σ , a renaming ξ of Cst uniquely extends to an endomorphism of $T(\Sigma, C)$, also denoted by ξ .*

Through the rest of the paper, we assume that C is a countable infinite set of constants, $>$ is a well-founded total ordering on C , Σ_0 and Σ are two signatures such that $\Sigma_0 \subseteq \Sigma$, E is a regular and collapse-free Σ_0 -theory and F is a Σ -theory.

Let G be a subset of $T(\Sigma, C)$ including C . A Σ_0 -term over a set of terms G is a term $u\sigma$ such that $u \in T(\Sigma_0, V)$ and σ is a substitution such that $Dom(\sigma) = Var(u)$ and $Ran(\sigma) \subseteq G$. By a slight abuse of notation, the set of Σ_0 -terms over G is denoted by $T(\Sigma_0, G)$. A *constant abstraction mapping modulo $F \cup E$* for G is a mapping $\pi : G \setminus C \rightarrow D$ such that D is a set of constants disjoint from C and for any $s, t \in G \setminus C$, $s =_{F \cup E} t$ iff $\pi(s) = \pi(t)$. An inverse mapping of π is any morphism $\pi^{-1} : D \rightarrow G \setminus C$ such that for any $t \in G \setminus C$, $\pi^{-1}(\pi(t)) =_{F \cup E} t$. For any $t \in T(\Sigma_0, G)$, t^{π_0} is called the 0-abstraction of t and is inductively defined as follows:

- $(f(t_1, \dots, t_m))^{\pi_0} = f(t_1^{\pi_0}, \dots, t_m^{\pi_0})$ if $f \in \Sigma_0$,
- $t^{\pi_0} = \pi(t)$ if $t \in G \setminus C$,
- $c^{\pi_0} = c$ if $c \in C$.

Following [6], G is called a Σ_0 -base of $F \cup E$ if for any term $t \in T(\Sigma, C)$ there exists a term $s \in T(\Sigma_0, G)$ such that $t =_{F \cup E} s$, and for any $s, s' \in T(\Sigma_0, G)$, $s =_{F \cup E} s'$ iff $s^{\pi_0} =_{F \cup E} s'^{\pi_0}$.

► **Definition 2** (*E*-constructed theory). *Let C be a countable infinite set of constants, $>$ a well-founded total ordering on C , Σ_0 and Σ two signatures such that $\Sigma_0 \subseteq \Sigma$, E a regular and collapse-free Σ_0 -theory and F a Σ -theory. An *E*-constructed normalizing mapping for $F \cup E$ is an idempotent mapping $NF : T(\Sigma, C) \rightarrow T(\Sigma, C)$ with the following properties:*

- for any $s, t \in T(\Sigma, C)$, $s =_{F \cup E} t$ iff $NF(s) =_E NF(t)$,
- for any $t \in T(\Sigma, C)$, $Cst(NF(t)) \subseteq Cst(t) \cup \{c_0\}$, where c_0 is minimal in C w.r.t $>$,
- for any $t \in T(\Sigma, C)$ and any $>$ -compatible renaming ξ of $Cst(t) \cup \{c_0\}$ such that $c_0\xi = c_0$, we have $NF(t\xi) = NF(t)\xi$,
- for any $f \in \Sigma_0$, any $t_1, \dots, t_m \in T(\Sigma, C)$, $NF(f(t_1, \dots, t_m)) =_E f(NF(t_1), \dots, NF(t_m))$,
- for any $c \in C$, $NF(c) = c$.
- Let $G = \{t \mid t \in T(\Sigma, C), t(\epsilon) \in (\Sigma \setminus \Sigma_0) \cup C, \text{ and } NF(t) = t\}$. For any $t \in T(\Sigma, C)$, $NF(t) \in T(\Sigma_0, G)$.

F is said to be E -constructed if there exists an E -constructed normalizing mapping for $F \cup E$. G is called the Σ_0 -base associated to NF . A term $t \in T(\Sigma, C)$ is NF -normalized if $NF(t) = t$. A substitution σ is NF -normalized if for each $x \in \text{Dom}(\sigma)$, $x\sigma$ is NF -normalized.

In Definition 2, the Σ_0 -base associated to NF is actually a Σ_0 -base of $F \cup E$. Therefore, Σ_0 is a set of constructors for $F \cup E$, following the definition of constructor studied in [6]. By Definition 2, we have that $=_{F \cup E}$ and $=_E$ coincide on Σ_0 -terms. Thus, for any E -constructed theory F , Σ_0 is a set of constructors for $F \cup E$ and the Σ_0 -reduct of $F \cup E$ is E . Moreover, note that $F \cup E$ -equality is decidable if NF is computable and E -equality is decidable.

► **Proposition 3.** For any E -constructed TRS (R, E) , R is an E -constructed theory such that an E -constructed normalizing mapping NF for $R \cup E$ is defined as follows: for any $t \in T(\Sigma, C)$, $NF(t) = t \downarrow_{R, E}$.

► **Example 4.** Through the rest of the paper we will include several examples using the following axioms: $EX = \{e(e(x, y), z) = e(x, y * z)\}$ for exponentiation, $H = \{e(x * y, z) = e(x, z) * e(y, z)\}$, for the homomorphism like property of exponentiation, $EXH = EX \cup H$, and AC for the AC theory of $*$. For each $F = EX, H, EXH$, the theory $F \cup AC$ is finite, and so the $F \cup AC$ -matching problem is finitary. However, the unification problem is undecidable for $EXH \cup AC$ and $H \cup AC$ [27]. For each $F = EX, H, EXH$, orienting the equalities from left to right in F leads to an AC -constructed TRS denoted by (F^\rightarrow, AC) . Then, the AC -constructed (F^\rightarrow, AC) provides an AC -constructed normalized mapping NF since normal forms are stable by variable renaming in equational convergent rewrite systems. Thus, for each $F = EX, H, EXH$, there exists an AC -constructed normalizing mapping NF for $F \cup AC = F^\rightarrow \cup AC$, meaning that F is AC -constructed. For all these AC -constructed theories, the Σ_0 -base G associated to NF corresponds to the set of NF -normalized terms rooted by a symbol not equal to $*$. Notice, if $NF(t)$ is rooted by e then $NF(t) \in G$ and so $NF(t) \in T(\Sigma_0, G)$. When t is not in G , $NF(t)$ is not necessarily in G . Consider for instance $F = EX$, $t = e(e(a, b), c) * a$, and $t' = a * e(a, c * b)$. Then $NF(t) = e(a, b * c) * a$ and $NF(t') = t'$ are terms in $T(\Sigma_0, G) \setminus G$. Since $t =_{EX \cup AC} t'$, we have $NF(t) =_{AC} NF(t')$. Assume a constant abstraction mapping π modulo $EX \cup AC$ for G such that $\pi(e(a, b * c)) = \pi(e(a, c * b)) = d$ since $e(a, b * c) =_{AC} e(a, c * b)$. One can check that $NF(t)^{\pi_0} = d * a =_{AC} a * d = NF(t')^{\pi_0}$.

► **Example 5.** Note, Definition 2 does not require that the theory be orientable into an E -constructed TRS. Theories satisfying a commutative property over an AC -symbol, $*$, such as $PC = \{fc(x * y, v * w) = fc(v * w, x * y)\}$ and $PCC = PC \cup \{fc(o(x), o(y)) = o(x) * o(y)\}$, satisfy Definition 2. For $F = PC, PCC$, one can check that F is AC -constructed and $F \cup AC$ is a finite syntactic theory. For the AC -constructed theory EX defined in Example 4, $EX \cup AC$ is also a finite syntactic theory. Actually, the syntacticness of $EX \cup AC$ follows from [23] since $EX \cup AC$ is collapse-free and $EX \cup AC$ -unification is known to be finitary [16]. While $F \cup AC$ is not orientable into an AC -constructed TRS for $F = PC, PCC$, Example 4 introduces an AC -constructed TRS for EX .

Unsurprisingly, any E -constructed theory corresponds to an E -convergent rewrite relation on ground terms. In that case, the corresponding set of rules is infinite and so this rewrite relation cannot be used in practice to compute the normal forms. In an E -constructed TRS, the rules are built over terms with variables and they are stable by instantiation. In an E -constructed theory, the corresponding rules are ground and a particular notion of stability is considered to allow a renaming of constants, provided that the renaming is $>$ -compatible. By Definition 2, a normal form of a term does not depend on the names used to denote the constants, but it depends on the ordering of the constants in the term. Since the equational

theory $F \cup E$ is not necessarily regular in Definition 2, a normal form of a term t may have some additional constants not occurring in t . However, a single additional constant suffices, and by Definition 2, it will be the minimal one w.r.t $>$.

► **Lemma 6.** *Let F be an E -constructed theory, and NF an E -constructed normalizing mapping for $F \cup E$. Let R_{NF} be the set of ground rules $t \rightarrow NF(t)$ such that $t \in T(\Sigma, C)$, $NF(t) \neq t$, $t(\epsilon) \in \Sigma \setminus \Sigma_0$ and any strict subterm of t is NF -normalized. Then, $\rightarrow_{R_{NF}, E}$ is E -constructed and for any $t \in T(\Sigma, C)$, $NF(t) =_E t \downarrow_{R_{NF}, E}$.*

► **Example 7.** Continuing from Example 4, consider any AC -constructed TRS $(F \rightarrow, AC)$ where $F = EX, H, EXH$. Let NF be the E -constructed normalizing mapping such that for any $t \in T(\Sigma, C)$, $NF(t) = t \downarrow_{F \rightarrow, AC}$. By Lemma 6, the normal forms w.r.t $(F \rightarrow, AC)$ coincide with the normal forms w.r.t $\rightarrow_{R_{NF}, AC}$ on $T(\Sigma, C)$.

When NF is an E -constructed normalizing mapping for $F \cup E$, a normal form $t \downarrow_{R_{NF}, E}$ is also simply denoted by $t \downarrow_{NF}$. In contrast to [14, 13, 18, 19], the class of E -constructed theories given by Definition 2 is closed by non-disjoint union sharing only symbols in E . In other words, the class of E -constructed theories is modular:

► **Theorem 8.** *Assume F_1 and F_2 are two E -constructed theories sharing only symbols in E such that for $i = 1, 2$, NF_i is an E -constructed normalizing mapping for $F_i \cup E$. Then, NF_1 and NF_2 can be extended to an E -constructed normalizing mapping $NF_{1,2}$ for $F_1 \cup F_2 \cup E$.*

► **Example 9.** Continuing from Examples 5 and 7, for each $F = EX, H, EXH, PC, PCC$ and for each integer $i \geq 1$, let F_i be the theory obtained from F by replacing any function symbol f in F not equal to $*$ by f_i . For instance, $F_i = EX_i = \{e_i(e_i(x, y), z) = e_i(x, y * z)\}$ if $F = EX$, and $F_i = H_i = \{e_i(x * y, z) = e_i(x, z) * e_i(y, z)\}$ if $F = H$. Theorem 8 allows us to combine any number of theories F_i .

From now on, F_i is assumed to be an E -constructed theory with an E -constructed normalizing mapping NF_i for the Σ_i -theory $F_i \cup E$, where $i = 1, 2$. Then, $F_1 \cup F_2 \cup E$ is an E -constructed theory. The E -constructed normalizing mapping $NF_{1,2}$ derived from NF_1 and NF_2 by Theorem 8 is simply denoted by NF , G is the Σ_0 -base associated to NF corresponding to the set of $((\Sigma_1 \cup \Sigma_2) \setminus \Sigma_0) \cup C$ -rooted NF -normalized terms, and π is a constant abstraction mapping modulo $F_1 \cup F_2 \cup E$ for G . Given any $i = 1, 2$ and the subsignature Σ_i of $\Sigma_1 \cup \Sigma_2$, a *term with true i -aliens* is a term t such that for any Σ_i -alien subterm u of t , $u \downarrow_{NF}$ is $\Sigma_{3-i} \setminus \Sigma_0$ -rooted. Given any term t with true i -aliens, the *i -abstraction of t* is denoted by t^{π_i} and defined as follows:

- for any $f \in \Sigma_i$ and any terms t_1, \dots, t_m , $(f(t_1, \dots, t_m))^{\pi_i} = f(t_1^{\pi_i}, \dots, t_m^{\pi_i})$,
- for any $\Sigma_{3-i} \setminus \Sigma_0$ -rooted term t , $t^{\pi_i} = \pi(t \downarrow_{NF})$,
- for any $c \in C$, $c^{\pi_i} = c$.

Given a substitution σ such that $x\sigma$ is a term with true i -aliens for any $x \in \text{Dom}(\sigma)$, we define $\sigma^{\pi_i} = \{x \mapsto (x\sigma)^{\pi_i} \mid x \in \text{Dom}(\sigma)\}$.

► **Lemma 10.** *For any $i = 1, 2$ and any term t with true i -aliens, $t^{\pi_i} =_{F_i \cup E} (t \downarrow_{NF})^{\pi_i}$.*

In general, an E -constructed normalizing mapping is not computable. However, we show that it is possible to get an approximation, called layer-reduced form, which is useful to decide the equality modulo a union of theories $F_1 \cup F_2 \cup E$ where both F_1 and F_2 are E -constructed.

► **Definition 11 (Layer-reduced form).** *Let Σ_0 and Σ be two signatures such that $\Sigma_0 \subseteq \Sigma$, E a Σ_0 -theory, and F an E -constructed Σ -theory with an E -constructed normalizing mapping NF . A layer-reduced form is a term in $T(\Sigma, C)$ defined inductively as follows:*

6:8 Combined Hierarchical Matching

- $f(t_1, \dots, t_m)$ is in layer-reduced form if $f \in \Sigma_0$ and for each $k \in [1, m]$, t_k is in layer-reduced form,
- t is in layer-reduced form if both t and $t \downarrow_{NF}$ are $\Sigma \setminus \Sigma_0$ -rooted,
- c is in layer-reduced form if $c \in C$.

Given any term $s \in T(\Sigma, C)$, a layer-reduced form of s associated to NF modulo $F \cup E$ is a layer-reduced form t such that $s =_{F \cup E} t$.

A layer-reduced term mapping returns a layer-reduced form of any input term.

► **Definition 12** (Layer-reduced term mapping). *Let Σ_0 and Σ be two signatures such that $\Sigma_0 \subseteq \Sigma$, E a Σ_0 -theory, F an E -constructed Σ -theory with an E -constructed normalizing mapping NF , and c_0 the minimal constant in C w.r.t $>$. A layer-reduced term mapping associated to NF for $F \cup E$ is an idempotent mapping $(_) \Downarrow : T(\Sigma, C) \rightarrow T(\Sigma, C)$ such that:*

- for any $t \in T(\Sigma, C)$, $t \Downarrow$ is a layer-reduced form of t associated to NF modulo $F \cup E$ such that $Cst(t \Downarrow) \subseteq Cst(t) \cup \{c_0\}$,
- for any $t \in T(\Sigma, C)$ and any $>$ -compatible renaming ξ of $Cst(t) \cup \{c_0\}$ such that $c_0 \xi = c_0$, we have $(t \xi) \Downarrow = (t \Downarrow) \xi$,
- for any $f \in \Sigma_0$ and any terms $t_1, \dots, t_m \in T(\Sigma, C)$, $f(t_1, \dots, t_m) \Downarrow = f(t_1 \Downarrow, \dots, t_m \Downarrow)$,
- for any $c \in C$, $c \Downarrow = c$.

A \Downarrow -ordering is an $F \cup E$ -compatible total ordering $>_{\Downarrow}$ on $T_{\Downarrow} = \{t \mid t(\epsilon) \in \Sigma \setminus \Sigma_0, t \Downarrow = t\}$ such that for any $t, t' \in T_{\Downarrow}$ and any $>$ -compatible renaming ξ of $Cst(t) \cup Cst(t') \cup \{c_0\}$ with $c_0 \xi = c_0$, we have $t >_{\Downarrow} t'$ iff $t \xi >_{\Downarrow} t' \xi$.

In order to decide $F_1 \cup F_2 \cup E$ -equality in a modular way, we show that a computable layer-reduced term mapping \Downarrow_i and a computable \Downarrow_i -ordering for $F_i \cup E$, together with a decidable $F_i \cup E$ -equality for $i = 1, 2$ are sufficient.

► **Theorem 13.** *Assume F_1 and F_2 are two E -constructed theories sharing only symbols in E such that $F_i \cup E$ has an E -constructed normalizing mapping NF_i , a computable layer-reduced term mapping \Downarrow_i associated to NF_i , a computable \Downarrow_i -ordering, and $F_i \cup E$ -equality is decidable for any $i = 1, 2$. Then, \Downarrow_1 and \Downarrow_2 (resp. , the \Downarrow_1 -ordering and the \Downarrow_2 -ordering) can be extended to a computable layer-reduced term mapping $\Downarrow_{1,2}$ associated to $NF_{1,2}$ (resp. , a computable $\Downarrow_{1,2}$ -ordering) such that for any $i = 1, 2$ and any term t , $t \Downarrow_{1,2}$ is a term with true i -aliens, and $F_1 \cup F_2 \cup E$ -equality is decidable.*

At first glance, the computability of \Downarrow and of its related \Downarrow -ordering seems difficult to obtain. Fortunately, there is a large class of theories for which we get for free the computability of these mappings and related orderings. The following lemmas are very useful to apply our combination results, e.g., Theorem 13:

► **Lemma 14.** *For any E -constructed theory F with a computable $F \cup E$ -canonizer stable by renaming, any computable layer-reduced term mapping \Downarrow has a computable \Downarrow -ordering.*

► **Lemma 15.** *For any E -constructed theory F such that $F \cup E$ is a regular theory with an $F \cup E$ -matching algorithm, a layer-reduced term mapping \Downarrow is computable.*

Proof. Consider the procedure defined as the repeated application of the following inference with a don't care non-determinism:

Expand $(u, t) \vdash (u\sigma, t)$
 where $x \in Var(u)$, $f \in \Sigma_0$, \bar{v} are fresh variables, $\sigma = \{x \mapsto f(\bar{v})\}$, $CSU_{F \cup E}(\{u\sigma = t\}) \neq \emptyset$.

Given any variable x , any term $t \in T(\Sigma, C)$ and the input pair (x, t) , the above procedure is necessarily terminating since F is E -constructed, computing a single pair (u', t) such that $CSU_{F \cup E}(\{u' = t\}) \neq \emptyset$, and for any $\sigma \in CSU_{F \cup E}(\{u' = t\})$, $u'\sigma$ is a layer-reduced form of t modulo $F \cup E$. ◀

► **Remark 16.** The E -constructed theory F_i is said to be E -inner if the normal form by NF_i of any $\Sigma_i \setminus \Sigma_0$ -rooted term in $T(\Sigma_i, C)$ remains $\Sigma_i \setminus \Sigma_0$ -rooted. When F_i is E -inner, the identity mapping provides a layer-reduced term mapping \Downarrow_i for $F_i \cup E$. If both $\Downarrow_1, \Downarrow_2$ are the identity mapping, then $\Downarrow_{1,2}$ remains the identity mapping.

When \Downarrow_i is given by a computable NF_i for $i = 1, 2$, $\Downarrow_{1,2}$ corresponds to the computable $NF_{1,2}$. Let us also mention the disjoint case $(\Sigma_0, E) = (\emptyset, \emptyset)$, where $\Downarrow_{1,2}$ is obtained without using an additional computable \Downarrow_i -ordering for $i = 1, 2$.

► **Example 17.** Continuing from Example 9, we have a computable NF for each theory $F \cup AC$ where $F = EX, H, EXH$. Notice, each of these NF s satisfies Definition 12 and provides a layer-reduced term mapping, \Downarrow . EX and H are regular theories, thus no new constant c_0 is introduced by \Downarrow and $Cst(t\Downarrow) = Cst(t)$. According to Remark 16, EX and PC are AC -inner theories for which a layer-reduced term mapping can be provided by the identity mapping. Contrary to PC , PCC is not AC -inner but $PCC \cup AC$ is finite and we can rely on Lemma 15 to get a computable layer-reduced term mapping. For H and EXH , the corresponding computable NF can be used as a layer-reduced term mapping. Thus, we have a computable \Downarrow for each $F \cup AC$ where $F = EX, H, EXH, PC, PCC$. Applying Theorem 13 we obtain, in a modular way, a computable layer-reduced term mapping for $F_1 \cup \dots \cup F_n \cup AC$, where $F = EX, H, EXH, PC, PCC$. Recall that the construction of the combined layer-reduced term mapping requires computable \Downarrow -orderings. For each $F = EX, H, EXH, PC, PCC$, there exists a computable \Downarrow -ordering since Lemma 14 applies. Finally, note that Theorem 13 does not require that the component theories $F_i \cup AC$ are regular. In the particular case of a non-regular AC -constructed TRS, the layer-reduced term mapping \Downarrow provided by the corresponding computable NF satisfies $Cst(t\Downarrow) \subseteq Cst(t)$ for any term t , and Lemma 14 still applies to get a computable \Downarrow -ordering since AC is a finite theory.

Consider now the problem of building an $F_1 \cup F_2 \cup E$ -matching algorithm where both F_1 and F_2 are assumed to be regular (as well as E). In that case, any matching problem has only ground solutions: given any equation $s =_{F_1 \cup F_2 \cup E}^? t$ with $t \in T(\Sigma_1 \cup \Sigma_2, C)$ and any substitution σ such that $s\sigma =_{F_1 \cup F_2 \cup E} t$, $\{s\sigma\} \cup \text{Ran}(\sigma) \subseteq T(\Sigma_1 \cup \Sigma_2, C)$. The following corollary is a direct consequence of Lemma 10 and paves the way for an $F_1 \cup F_2 \cup E$ -matching procedure combining an $F_1 \cup E$ -matching algorithm and an $F_2 \cup E$ -matching algorithm:

► **Corollary 18.** For any $i = 1, 2$, any Σ_i -term s , any term $t \in T(\Sigma_1 \cup \Sigma_2, C)$ such that $t\Downarrow_{1,2} = t$, and any NF -normalized substitution σ , $s\sigma =_{F_1 \cup F_2 \cup E} t$ iff $s(\sigma^{\pi_i}) =_{F_i \cup E} t^{\pi_i}$.

By Corollary 18, the terminating procedure [29, 32, 15, 33] combining the matching algorithms in regular theories remains sound and complete in our extended setting.

► **Theorem 19.** If F_1 and F_2 are two E -constructed theories sharing only symbols in E such that $F_i \cup E$ is a regular theory with a computable layer-reduced term mapping \Downarrow_i , a computable \Downarrow_i -ordering and an $F_i \cup E$ -matching algorithm for $i = 1, 2$, then $F_1 \cup F_2$ is E -constructed and $F_1 \cup F_2 \cup E$ is a regular theory with a computable layer-reduced term mapping $\Downarrow_{1,2}$, a computable $\Downarrow_{1,2}$ -ordering and an $F_1 \cup F_2 \cup E$ -matching algorithm.

Theorem 19 can be applied to finite theories since any finite theory is a particular case of a regular (and collapse-free) theory with a computable layer-reduced term mapping \Downarrow (cf. Lemma 15), a computable \Downarrow -ordering (cf. Lemma 14), and a matching algorithm. Indeed, the matching problem is known to be finitary in any finite theory, thanks to a reduction to syntactic matching via the enumeration of the finitely terms in a given equivalence class modulo the theory. This brute-force method should be avoided whenever it is possible.

4 Finite Syntactic Theories and their Combinations

In this section, we focus on the class of finite syntactic theories. In that class, any theory has a mutation-based matching algorithm. The class of finite syntactic theories is known to be closed by disjoint union [28]. More precisely, if F_1 and F_2 are signature-disjoint finite theories and F_i has a resolvent presentation S_i for $i = 1, 2$, then $F_1 \cup F_2$ is finite and has a resolvent presentation $S_1 \cup S_2$. In the non-disjoint case where F_1 and F_2 are E -constructed theories sharing only symbols in E and $F_i \cup E$ has a resolvent presentation S_i for $i = 1, 2$, it is easy to see that $S_1 \cup S_2$ is not necessarily a resolvent presentation of $F_1 \cup F_2 \cup E$:

► **Example 20.** Consider $(\Sigma_0, E) = (\{c\}, \emptyset)$ and $(\Sigma_i, F_i) = (\{f_i, c\}, \{f_i(x) = c(x)\})$ for $i = 1, 2$. F_i is a resolvent presentation of $F_i \cup E$ and F_i is E -constructed for $i = 1, 2$ but $F_1 \cup F_2$ cannot be a resolvent presentation of $F_1 \cup F_2 \cup E$ since $f_1(x) =_{F_1 \cup F_2 \cup E} f_2(x)$.

To get that the resolvent presentation of $F_1 \cup F_2 \cup E$ is the union of the resolvent presentations of $F_1 \cup E$ and $F_2 \cup E$, a restricted class of E -constructed theories is needed:

► **Definition 21** (*E -capped theory*). *Let E be a regular and collapse-free Σ_0 -theory, F an E -constructed Σ -theory with an E -constructed normalizing mapping NF for $F \cup E$, and G the Σ_0 -base associated to NF . The E -constructed normalizing mapping NF is said to be E -capped if for any $\Sigma \setminus \Sigma_0$ -rooted term $t \in T(\Sigma, C)$, $NF(t)$ is a term $u\sigma \in T(\Sigma_0, G)$ such that $u \in T(\Sigma_0, V)$, $\text{Var}(u) = \text{Dom}(\sigma)$ and $\text{Ran}(\sigma) \subseteq G \setminus C$. An E -constructed theory F with an E -capped normalizing mapping NF is said to be E -capped.*

► **Example 22.** In Definition 21, the term u can be a variable and so any E -inner theory as defined in Remark 16 is E -capped. Consider the theories defined in Examples 4 and 5. For $F = EX, PC$, the theory F is E -capped since F is E -inner. For $F = H, EXH, PCC$, the theory F is E -capped without being E -inner.

When F_1 and F_2 are two E -capped theories sharing only symbols in E , for any $\Sigma_1 \setminus \Sigma_0$ -rooted term t_1 and any $\Sigma_2 \setminus \Sigma_0$ -rooted term t_2 , t_1 cannot be equal to t_2 modulo $F_1 \cup F_2 \cup E$. In [19], the following result has been shown: if F_1 and F_2 are two E -capped theories sharing only symbols in E and $F_i \cup E$ is regular collapse-free with a resolvent presentation S_i for $i = 1, 2$, then $F_1 \cup F_2$ is E -capped and $F_1 \cup F_2 \cup E$ is regular collapse-free with a resolvent presentation $S_1 \cup S_2$.

► **Example 23.** Consider $(\Sigma_0, E) = (\{c\}, \emptyset)$ and $(\Sigma_i, F_i) = (\{f_i, g_i, c\}, \{f_i(x) = c(g_i(x))\})$ for $i = 1, 2$. F_i is a resolvent presentation of $F_i \cup E$ and F_i is E -capped and regular collapse-free for $i = 1, 2$. By the modularity result in [19] mentioned above, $F_1 \cup F_2$ is E -capped and $F_1 \cup F_2$ is a resolvent presentation of $F_1 \cup F_2 \cup E$.

When $F_i \cup E$ is finite, only finitely many distinct non-normalized terms can have the same normal form w.r.t $NF_{1,2}$. Since, for any s, t , $s =_{F_1 \cup F_2 \cup E} t$ iff $NF_{1,2}(s) =_E NF_{1,2}(t)$ where E is necessarily finite, we have that $F_1 \cup F_2 \cup E$ is finite too.

► **Theorem 24.** *If F_1 and F_2 are two E -capped theories sharing only symbols in E such that $F_i \cup E$ is a finite theory with a resolvent presentation S_i for $i = 1, 2$, then $F_1 \cup F_2$ is E -capped and $F_1 \cup F_2 \cup E$ is a finite theory with a resolvent presentation $S_1 \cup S_2$.*

With E -constructed TRSs, another resolvence allows us to get rid of the E -capped assumption.

► **Definition 25** (Innermost-resolvent E -constructed TRS). *An E -constructed TRS (R, E) is said to be innermost-resolvent if any innermost rewrite derivation $s \rightarrow_{R, E}^* t$ includes at most one rewrite step applied at the root position. An innermost-resolvent TRS (R, E) is finite if $R \cup E$ is finite.*

► **Example 26.** Continuing from Example 7, consider any AC -constructed TRS (F^{\rightarrow}, AC) where $F = EX, H, EXH$. Applying the rule corresponding to EX more than once at the root would violate the innermost strategy. The rule corresponding to H moves the constructor symbol $*$ to the root and thus disallows any further root rewriting. Thus, (F^{\rightarrow}, AC) is innermost-resolvent for each $F = EX, H, EXH$.

Following the terminology in [10], $R \cup E$ is 2-syntactic when (R, E) is innermost-resolvent.

► **Theorem 27.** *Let (R_1, E) and (R_2, E) be two finite innermost-resolvent E -constructed TRSs sharing only symbols in E . If $\rightarrow_{R_1 \cup R_2}$ is E -terminating, then $(R_1 \cup R_2, E)$ is a finite innermost-resolvent E -constructed TRS.*

► **Example 28.** Continuing from Examples 20 and 23, consider $(\Sigma_0, E) = (\{c\}, \emptyset)$, $(\Sigma_1, R_1) = (\{f_1, c\}, \{f_1(x) \rightarrow c(x)\})$ and $(\Sigma_2, R_2) = (\{f_2, g_2, c\}, \{f_2(x) \rightarrow c(g_2(x))\})$. (R_1, E) and (R_2, E) are two finite innermost-resolvent E -constructed TRSs sharing only symbols in E and $\rightarrow_{R_1 \cup R_2}$ is E -terminating. By Theorem 27, $(R_1 \cup R_2, E)$ is a finite innermost-resolvent E -constructed TRS.

5 Hierarchical Matching

Norm $\{s = t\} \cup \Gamma \vdash \{s = t\downarrow\} \cup \Gamma$

where s is a non-ground term and t is a ground term such that $t\downarrow \neq t$.

Triv $\{s = t\} \cup \Gamma \vdash \Gamma$

where s, t are ground terms such that $s\downarrow = s$, $t\downarrow = t$, and $s =_{F \cup E} t$.

■ **Figure 1** NT rules.

We investigate in this section the problem of building an $F \cup E$ -matching algorithm in the case F is E -constructed, $F \cup E$ has a computable layer-reduced term mapping \downarrow , a computable \downarrow -ordering, and an E -matching algorithm is known. A hierarchical matching algorithm for $F \cup E$ is defined as an inference system including the inference rules in $NT \cup HM_E$ where NT and HM_E are respectively given in Figure 1 and in Figure 2. The rules in NT are clearly sound and complete in $F \cup E$, by definition of \downarrow . The rules in $HM_E \setminus \{\mathbf{Solve-M}\}$ are sound and complete in any equational theory. To show that **Solve-M** is sound and complete in $F \cup E$, we rely on the 0-abstraction of a term in layer-reduced form. The 0-abstraction of any term $t \in T(\Sigma_0, G)$, denoted by t^{π_0} , has been introduced just before Definition 2 and it can be extended to a larger set of terms. A *term with true 0-aliens* is a term t such that for any Σ_0 -alien subterm u of t , $u\downarrow_{NF}$ is $\Sigma \setminus \Sigma_0$ -rooted. Given any term t with true 0-aliens, the 0-abstraction of t is denoted by t^{π_0} and defined as follows:

6:12 Combined Hierarchical Matching

Rep $\{x = t\} \cup \Gamma \vdash \{x = t\} \cup (\Gamma\{x \mapsto t\})$

where x is a variable occurring in Γ and t is a ground term.

Flatten-M $\{f(\bar{u}) = t\} \cup \Gamma \vdash \{f(\bar{x}) = t, \bar{u} = \bar{x}\} \cup \Gamma$

where $f(\bar{u})$ is a non-ground $\Sigma \setminus \Sigma_0$ -rooted term, t is ground, and \bar{x} are fresh variables.

VA-M $\{s[u] = t\} \cup \Gamma \vdash \{s[x] = t, u = x\} \cup \Gamma$

where s is a non-ground Σ_0 -rooted term, u is a Σ_0 -alien subterm of s , t is a ground, and x is a fresh variable.

Solve-M $\Gamma \cup \Gamma_0 \vdash \Gamma \cup \hat{\sigma}$

where $\Gamma_0 = \{s_k = t_k\}_{k \in K}$, $s_k \in T(\Sigma_0, V \cup C)$ and $t_k \in T(\Sigma, C)$ for each $k \in K$, $\Gamma_0^{\pi_0} = \{s_k = t_k^{\pi_0}\}_{k \in K}$, $CSU_E(\Gamma_0^{\pi_0}) \neq \emptyset$, $\sigma_0 \in CSU_E(\Gamma_0^{\pi_0})$, and $\hat{\sigma}$ is the solved form of $\sigma = \sigma_0 \pi^{-1}$.

■ **Figure 2** HM_E rules.

- for any $f \in \Sigma_0$ and any terms t_1, \dots, t_m , $(f(t_1, \dots, t_m))^{\pi_0} = f(t_1^{\pi_0}, \dots, t_m^{\pi_0})$,
- for any $\Sigma \setminus \Sigma_0$ -rooted term t , $t^{\pi_0} = \pi(t \downarrow_{NF})$,
- for any $c \in C$, $c^{\pi_0} = c$.

Given a substitution σ such that $x\sigma$ is a term with true 0-aliens for any $x \in Dom(\sigma)$, we define $\sigma^{\pi_0} = \{x \mapsto (x\sigma)^{\pi_0} \mid x \in Dom(\sigma)\}$.

► **Lemma 29.** For any term t with true 0-aliens, $t^{\pi_0} =_E (t \downarrow_{NF})^{\pi_0}$.

► **Lemma 30.** For any ground terms s, t in layer-reduced form, we have:

- if s, t are Σ_0 -rooted or $s, t \in C$, then $s =_{F \cup E} t \Leftrightarrow s^{\pi_0} =_E t^{\pi_0}$,
- if s is Σ_0 -rooted and t is $\Sigma \setminus \Sigma_0$ -rooted or $s \in C$ and t is Σ -rooted, then $s \neq_{F \cup E} t$.

► **Corollary 31.** For any ground term t in layer-reduced form, any Σ_0 -term s and any NF -normalized substitution σ , $s\sigma =_{F \cup E} t$ iff $s(\sigma^{\pi_0}) =_E t^{\pi_0}$.

Corollary 31 follows from Lemma 29. It shows that **Solve-M** is sound and complete in $F \cup E$. To solve any $F \cup E$ -matching problem, we need to complete $NT \cup HM_E$ by some inference system, say U , to transform the match-equations that cannot be handled by $NT \cup HM_E$.

► **Definition 32** (Hierarchical matching algorithm). Assume an E -matching algorithm, a computable layer-reduced term mapping \downarrow for $F \cup E$, and an inference system U satisfying the following assumptions:

- (a) no single inference rule in U is sound and complete for an arbitrary equational theory;
- (b) U is sound and complete for $F \cup E$ provided that all the inference rules in U are applied using a don't know non-determinism;
- (c) each equation that can be solved by **Solve-M** must remain unchanged by U .

A hierarchical matching algorithm for $F \cup E$ is an inference system denoted by $HM_E(\downarrow, U)$ and defined by the set of rules in $NT \cup HM_E \cup U$ (cf. Figures 1 and 2) such that the following properties hold for any input set Γ of equations $s = t$ where s or t is ground:

- the repeated application of rules in $HM_E(\downarrow, U)$ terminates with the following order of priority: **Norm**, **Triv**, **Rep**, **Flatten-M**, **VA-M**, U , **Solve-M**;
- any normal form of Γ w.r.t $HM_E(\downarrow, U)$ obtained by the above strategy is $F \cup E$ -unifiable iff it is a matching problem in solved form.

By definition, any hierarchical matching algorithm for $F \cup E$ is a sound and complete $F \cup E$ -matching algorithm. In the following, we give examples of theories with hierarchical matching algorithms.

► **Lemma 33.** *Let DM_R be the inference system given in Figure 3. For any finite innermost-resolvent E -constructed TRSs (R, E) , $R \cup E$ admits a hierarchical matching algorithm of the form $HM_E(\downarrow_{R,E}, DM_R)$.*

In Lemma 33, the soundness and completeness of DM_R follows directly from the assumption that (R, E) is innermost-resolvent and the fact that ground terms are normalized before any rule from DM_R applies. Since $R \cup E$ is finite, $HM_E(\downarrow_{R,E}, DM_R)$ is terminating.

$$\begin{array}{l} \mathbf{Dec} \quad \{f(\bar{v}) = f(\bar{t})\} \cup \Gamma \vdash \{\bar{v} = \bar{t}\} \cup \Gamma \quad \text{where } f \in \Sigma \setminus \Sigma_0 \\ \mathbf{Mut}_R \quad \{f(\bar{v}) = g(\bar{t})\} \cup \Gamma \vdash \{\bar{v} = \bar{l}, \bar{r} = \bar{t}\} \cup \Gamma \quad \text{where } f(\bar{l}) \rightarrow g(\bar{r}) \in R \end{array}$$

■ **Figure 3** DM_R rules.

► **Example 34.** Continuing from Example 26, for each $F = EX, H, EXH$, the AC -constructed TRS (F^\rightarrow, AC) is innermost-resolvent and $F^\rightarrow \cup AC$ has a hierarchical matching algorithm of the form $HM_{AC}(\downarrow_{F^\rightarrow, AC}, DM_{F^\rightarrow})$.

► **Lemma 35.** *Assume F is E -constructed and $F \cup E$ is a finite theory with a resolvent presentation S and a computable layer-reduced term mapping \downarrow . Let DM_S be the inference system obtained from the one in Figure 3 by replacing any rule from R by an equality from S . Then $F \cup E$ has a hierarchical matching algorithm of the form $HM_E(\downarrow, DM_S)$.*

In Lemma 35, the soundness and completeness of DM_S follows directly from the assumption that S is a resolvent presentation of $F \cup E$. In addition, $HM_E(\downarrow, DM_S)$ is terminating since $F \cup E$ is finite.

► **Example 36.** For $F = EX, PC, PCC$, we have that $F \cup AC$ is a finite theory with a resolvent presentation S and a computable layer-reduced term mapping \downarrow . Thus, $HM_{AC}(\downarrow, DM_S)$ is a hierarchical matching algorithm for $F \cup AC$. The resolvent presentation S includes some Σ_0 -equalities for $\Sigma_0 = \{*\}$. These Σ_0 -equalities, corresponding to a resolvent presentation of AC , are not used in the application of DM_S .

6 Hierarchical Matching in Combined E -Constructed Theories

In our hierarchical approach, combining hierarchical matching algorithms parameterized by U_1 and U_2 can be viewed as a hierarchical matching algorithm parameterized by $U_1 \cup U_2$. The following remark details how the inference rules in U_i involving ground Σ_i -terms are extended to handle ground $\Sigma_1 \cup \Sigma_2$ -terms.

► **Remark 37.** Assume an $F_i \cup E$ -matching algorithm of the form $HM_E(\downarrow_i, U_i)$ for $i = 1, 2$. The inference system U_i is defined for matching-equations with ground terms in $T(\Sigma_i, C)$. To handle ground terms in $T(\Sigma_1 \cup \Sigma_2, C)$, U_i must be extended in the expected manner via i -abstraction, leading to a signature extension of U_i defined as follows for any problem P including some function symbol in $\Sigma_{3-i} \setminus \Sigma_0$: $P \vdash_{U_i} Q \pi^{-1}$ if $P^{\pi_i} \vdash_{U_i} Q$, where P^{π_i} denotes the E_i -matching problem obtained from P by replacing each ground side t in P by t^{π_i} . This is sound and complete by Corollary 18, and the fact that ground sides in P are in layer-reduced form since **Norm** is applied eagerly before U_i . In the same way, **Solve-M** has been extended to handle ground sides in $T(\Sigma_1 \cup \Sigma_2, C)$. In that case, the 0-abstraction is used to get an E -matching problem.

► **Theorem 38.** *If F_1 and F_2 are two E -constructed theories sharing only symbols in E such that $F_i \cup E$ is a regular theory with a computable layer-reduced term mapping \Downarrow_i , a computable \Downarrow_i -ordering, and a hierarchical matching algorithm of the form $HM_E(\Downarrow_i, U_i)$ for $i = 1, 2$. Then $F_1 \cup F_2$ is an E -constructed and $F_1 \cup F_2 \cup E$ is a regular theory with a computable layer-reduced term mapping $\Downarrow_{1,2}$, a computable $\Downarrow_{1,2}$ -ordering, and a hierarchical matching algorithm of the form $HM_E(\Downarrow_{1,2}, U_1 \cup U_2)$.*

Proof. The combination algorithm for the matching problem that allows us to obtain Theorem 19 can be expressed as a hierarchical matching algorithm for $F_1 \cup F_2 \cup E$ of the form $HM_E(\Downarrow_{1,2}, \{\mathbf{Solve-M}_1, \mathbf{Solve-M}_2\})$, where $\mathbf{Solve-M}_i$ for $i = 1, 2$ is defined as follows in a way similar to **Solve-M**:

$$\begin{aligned} \mathbf{Solve-M}_i \quad & \Gamma \cup \Gamma_i \vdash \Gamma \cup \hat{\sigma} \\ \text{where } \Gamma_i = & \{s_k = t_k\}_{k \in K}, s_k \in T(\Sigma_i \setminus \Sigma_0, V \cup C), t_k \in T(\Sigma_1 \cup \Sigma_2, C) \text{ for each } k \in K, \\ \Gamma_i^{\pi_i} = & \{s_k = t_k^{\pi_i}\}_{k \in K}, CSU_{F_i \cup E}(\Gamma_i^{\pi_i}) \neq \emptyset, \sigma_i \in CSU_{F_i \cup E}(\Gamma_i^{\pi_i}), \text{ and } \hat{\sigma} \text{ is the solved form of} \\ & \sigma = \sigma_i \pi_i^{-1}. \end{aligned}$$

Assume $F_i \cup E$ has a hierarchical matching algorithm of the form $HM_E(\Downarrow_i, U_i)$ for any $i = 1, 2$. Then, $\mathbf{Solve-M}_i$ can be replaced by $HM_E(\Downarrow_i, U_i)$. Due to the rule application strategy used in any hierarchical matching algorithm, $\mathbf{Solve-M}_i$ applies only on match-equations $s = t$ such that s is a flat non-ground $\Sigma_i \setminus \Sigma_0$ -term, and t is a ground term in layer-reduced form w.r.t $\Downarrow_{1,2}$. Thus, U_i is sufficient to replace $\mathbf{Solve-M}_i$ for $i = 1, 2$, and so the combination matching algorithm is actually of the form $HM_E(\Downarrow_{1,2}, U_1 \cup U_2)$. ◀

► **Example 39.** In Examples 34 and 36, we have shown that $F_i \cup AC$ has a hierarchical matching algorithm for each $F = EX, H, EXH, PC, PCC$. By Theorem 38, $F_1 \cup \dots \cup F_n \cup AC$ has a hierarchical matching algorithm.

Notice, Theorem 38 applies to E -constructed regular theories F_i where $F_i \cup E$ is not necessarily finite. In the particular case of finite theories, we get the following corollaries.

► **Corollary 40.** *Assume (R_1, E) and (R_2, E) are two finite innermost-resolvent E -constructed TRSs sharing only symbols in E . If $\rightarrow_{R_1 \cup R_2}$ is E -terminating, then $(R_1 \cup R_2, E)$ is a finite innermost-resolvent E -constructed TRS and $R_1 \cup R_2 \cup E$ admits a hierarchical matching algorithm of the form $HM_E(\Downarrow_{R_1 \cup R_2, E}, DM_{R_1} \cup DM_{R_2})$.*

Corollary 40 is a continuation of Theorem 27. Interestingly, the hierarchical matching algorithm for $R_1 \cup R_2 \cup E$ can be obtained from Theorem 38 but also as a consequence of Lemma 33 since $HM_E(\Downarrow_{R_1 \cup R_2, E}, DM_{R_1 \cup R_2})$ coincides with $HM_E(\Downarrow_{R_1 \cup R_2, E}, DM_{R_1} \cup DM_{R_2})$.

► **Example 41.** Continuing from Examples 9 and 34, we can combine any number of exponentiation/homomorphic theories $F_i \cup AC$ for $F = EX, H, EXH$ sharing only the AC -symbol $*$ and obtain a hierarchical matching algorithm of the form given by Corollary 40.

► **Corollary 42.** *If F_1 and F_2 are two E -capped theories sharing only symbols in E such that $F_i \cup E$ is a finite theory with a resolvent presentation S_i , a computable layer-reduced term mapping \Downarrow_i , and a hierarchical matching algorithm of the form $HM_E(\Downarrow_i, DM_{S_i})$ for $i = 1, 2$. Then $F_1 \cup F_2$ is E -capped and $F_1 \cup F_2 \cup E$ is a finite theory with a resolvent presentation $S_1 \cup S_2$, a computable layer-reduced term mapping $\Downarrow_{1,2}$, and a hierarchical matching algorithm of the form $HM_E(\Downarrow_{1,2}, DM_{S_1} \cup DM_{S_2})$.*

Corollary 42 is a continuation of Theorem 24. Again, the hierarchical matching algorithm for $F_1 \cup F_2 \cup E$ can be obtained from Theorem 38 but also as a consequence of Lemma 35 since $HM_E(\Downarrow_{1,2}, DM_{S_1 \cup S_2})$ coincides with $HM_E(\Downarrow_{1,2}, DM_{S_1} \cup DM_{S_2})$.

► **Example 43.** Continuing from Examples 9 and 36, we can combine any number of finite syntactic theories $F_i \cup AC$ for $F = EX, PC, PCC$ sharing only the AC -symbol $*$ and obtain a finite syntactic theory with a hierarchical matching algorithm of the form given by Corollary 42.

7 Hierarchical Decision Procedures for the Word-Problem

In a hierarchical matching algorithm for $F \cup E$, it is mandatory to be able to decide $F \cup E$ -equality of terms in layer-reduced form (cf. rules in NT , Figure 1). In a way similar to hierarchical $F \cup E$ -matching, it is possible to follow a simple hierarchical approach for solving the particular $F \cup E$ -unification problem where both sides of each equation are ground terms in layer-reduced form. In that particular case, we assume a decidable E -equality, an E -constructed theory F , and a computable layer-reduced term mapping \Downarrow for $F \cup E$. Consider the following inference rule:

$$\mathbf{Solve-W} \quad \{s = t\} \cup \Gamma \vdash \Gamma \quad \text{where } (s(\epsilon), t(\epsilon)) \in \Sigma_0 \text{ or } s, t \in C \text{ and } s^{\pi_0} =_E t^{\pi_0}$$

together with an inference system U satisfying the same assumptions (a) and (b) as in Definition 32 and for which each equation that can be solved by **Solve-W** must remain unchanged by U . A *hierarchical decision procedure for the $F \cup E$ -equality of terms in layer-reduced form w.r.t \Downarrow* is an inference system denoted by $HW_E(U)$ and defined by the set of rules in $\{\mathbf{Solve-W}\} \cup U$ such that, for any input set Γ of equations $s = t$ where s and t are ground terms in layer-reduced form w.r.t \Downarrow , the repeated application of rules in $HW_E(U)$ terminates with the order of priority $U, \mathbf{Solve-W}$, and the empty set of equations is the unique $F \cup E$ -unifiable normal form w.r.t $HW_E(U)$. By definition, $HW_E(U)$ is a sound, complete, and terminating procedure deciding the $F \cup E$ -equality of terms in layer-reduced form. There are two major classes of E -constructed theories F with a hierarchical decision procedure for the $F \cup E$ -equality of terms in layer-reduced form:

1. If (R, E) is an E -constructed TRS and E is finite, then $HW_E(\{\mathbf{Dec}\})$ is a hierarchical decision procedure for the $R \cup E$ -equality of terms in layer-reduced form w.r.t $\Downarrow_{R,E}$, where **Dec** is given in Figure 3. This holds since the $\Sigma \setminus \Sigma_0$ -symbols do not occur in E .
2. If F is E -constructed and $F \cup E$ is a finite theory with a resolvent presentation S , then $HW_E(\{\mathbf{Dec}, \mathbf{Mut-W}_S\})$ is a hierarchical decision procedure for the $F \cup E$ -equality of terms in layer-reduced form, where **Dec** is given in Figure 3 and **Mut-W_S** is as follows:
 $\mathbf{Mut-W}_S \quad \{f(\bar{v}) = g(\bar{t})\} \cup \Gamma \vdash \Gamma \quad \text{where } f(\bar{l}) = g(\bar{r}) \in S, CSU_{F \cup E}(\{\bar{l} = \bar{v}, \bar{r} = \bar{t}\}) \neq \emptyset.$
This can be shown using the same proof argument as in Lemma 35.

The class of E -constructed theories F with a hierarchical decision procedure for the $F \cup E$ -equality of terms in layer-reduced form satisfies a modular property described below.

► **Theorem 44.** *Under the same assumptions as in Theorem 13, if $HW_E(U_i)$ is a hierarchical decision procedure for the $F_i \cup E$ -equality of terms in layer-reduced form w.r.t \Downarrow_i , for $i = 1, 2$, then, $HW_E(U_1 \cup U_2)$ is a hierarchical decision procedure for the $F_1 \cup F_2 \cup E$ -equality of terms in layer-reduced form w.r.t $\Downarrow_{1,2}$.*

Proof. By Theorem 13, $HW_E(\{\mathbf{Solve-W}_1, \mathbf{Solve-W}_2\})$ is a hierarchical decision procedure for the $F_1 \cup F_2 \cup E$ -equality of terms in layer-reduced form w.r.t $\Downarrow_{1,2}$, where for $i = 1, 2$, **Solve-W_i** is as follows:

$$\mathbf{Solve-W}_i \quad \{s = t\} \cup \Gamma \vdash \Gamma \quad \text{where } s(\epsilon), t(\epsilon) \in \Sigma_i \setminus \Sigma_0 \text{ and } s^{\pi_i} =_{F_i \cup E} t^{\pi_i}.$$

The use of U_i being extended to ground mixed terms via i -abstraction (cf. Remark 37), **Solve- W_i** can be replaced by U_i . ◀

Notice, Theorem 44 applies to E -constructed theories F_i where $F_i \cup E$ can be non-regular.

8 Related Work and Concluding Remarks

The theories $F \cup E$ we are interested in are conservative extensions of E for which the symbols in the signature Σ_0 of E are constructors, meaning that $F \cup E$ admits a Σ_0 -basis [6, 35]. In [6], a modularity result was shown for the computability of normal forms over the Σ_0 -basis. This result requires that normal forms are stable by variable renaming. In contrast, we rely on a stability by constant renaming, provided that the renaming follows an arbitrary total ordering over the constants. Moreover, we give a modular construction for computable layer-reduced term mappings which are sufficient approximations of the normalizing mappings used to define the E -constructed theories. The notion of layer-reduced form is well-known in the context of disjoint combination [31], but this is the first time a modular construction of layer-reduced forms is proposed for theories sharing constructors modulo E . The combination problem for both unification and matching in constructor-sharing theories has been investigated for a while [12, 5, 32, 14, 15, 33] but we now consider the general case of constructors modulo E to go beyond the case of absolutely free constructors. We have shown that our hierarchical approach is a well-suited framework to deal with non-absolutely free constructors. This hierarchical approach has been initiated to study the unification problem in various classes of E -constructed theories [18, 19]. As shown here, the restriction to the matching problem allows us to get hierarchical matching algorithms for larger classes of E -constructed theories, and this completes the terminating cases that have been recently identified for hierarchical unification [18, 19]. The modularity results shown here for the matching problem can be viewed as non-disjoint extensions of the ones known in the disjoint case for the matching problem both in regular theories [29] and in finite syntactic theories [28].

In the future, we are interested in developing new decision procedures for combined theories sharing constructors modulo E . More precisely, we target the knowledge problems considered in protocol analysis, for which some first results have been obtained for combined theories sharing absolutely free constructors [17]. Again, the hierarchical approach seems very useful to move from absolutely free constructors to constructors modulo E . More generally, our project consists in applying the hierarchical approach to constraint solving problems that occur in protocol analysis, including particular forms of disunification problems.

References

- 1 Alessandro Armando, Silvio Ranise, and Michaël Rusinowitch. A rewriting approach to satisfiability procedures. *Inf. Comput.*, 183(2):140–164, 2003.
- 2 Franz Baader and Tobias Nipkow. *Term rewriting and all that*. Cambridge University Press, 1998.
- 3 Franz Baader and Klaus U. Schulz. Unification in the union of disjoint equational theories: Combining decision procedures. *J. Symb. Comput.*, 21(2):211–243, 1996.
- 4 Franz Baader and Wayne Snyder. Unification theory. In John Alan Robinson and Andrei Voronkov, editors, *Handbook of Automated Reasoning (in 2 volumes)*, pages 445–532. Elsevier and MIT Press, 2001.
- 5 Franz Baader and Cesare Tinelli. Combining decision procedures for positive theories sharing constructors. In Sophie Tison, editor, *Rewriting Techniques and Applications, 13th Interna-*

- tional Conference, RTA 2002, Copenhagen, Denmark, July 22-24, 2002, Proceedings*, volume 2378 of *Lecture Notes in Computer Science*, pages 352–366. Springer, 2002.
- 6 Franz Baader and Cesare Tinelli. Deciding the word problem in the union of equational theories. *Inf. Comput.*, 178(2):346–390, 2002.
 - 7 Leo Bachmair, Harald Ganzinger, Christopher Lynch, and Wayne Snyder. Basic paramodulation. *Inf. Comput.*, 121(2):172–192, 1995.
 - 8 Bruno Blanchet. Modeling and verifying security protocols with the Applied Pi calculus and ProVerif. *Foundations and Trends in Privacy and Security*, 1(1-2):1–135, 2016.
 - 9 Peter Borovanský, Claude Kirchner, H el ene Kirchner, and Pierre-Etienne Moreau. ELAN from a rewriting logic point of view. *Theor. Comput. Sci.*, 285(2):155–185, 2002.
 - 10 Alexandre Boudet and Evelyne Contejean. On n -syntactic equational theories. In H el ene Kirchner and Giorgio Levi, editors, *Algebraic and Logic Programming, Third International Conference, Volterra, Italy, September 2-4, 1992, Proceedings*, volume 632 of *Lecture Notes in Computer Science*, pages 446–457. Springer, 1992.
 - 11 Manuel Clavel, Francisco Dur an, Steven Eker, Patrick Lincoln, Narciso Mart ı-Oliet, Jos e Meseguer, and Carolyn L. Talcott, editors. *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.
 - 12 Eric Domenjoud, Francis Klay, and Christophe Ringeissen. Combination techniques for non-disjoint equational theories. In Alan Bundy, editor, *Automated Deduction - CADE-12, 12th International Conference on Automated Deduction, Nancy, France, June 26 - July 1, 1994, Proceedings*, volume 814 of *Lecture Notes in Computer Science*, pages 267–281. Springer, 1994.
 - 13 Ajay Kumar Eeralla, Serdar Erbatur, Andrew M. Marshall, and Christophe Ringeissen. Rule-based unification in combined theories and the finite variant property. In Carlos Mart ın-Vide, Alexander Okhotin, and Dana Shapira, editors, *Language and Automata Theory and Applications - 13th International Conference, LATA 2019, St. Petersburg, Russia, March 26-29, 2019, Proceedings*, volume 11417 of *Lecture Notes in Computer Science*, pages 356–367. Springer, 2019.
 - 14 Serdar Erbatur, Deepak Kapur, Andrew M. Marshall, Paliath Narendran, and Christophe Ringeissen. Hierarchical combination. In Maria Paola Bonacina, editor, *Automated Deduction - CADE-24 - 24th International Conference on Automated Deduction, Lake Placid, NY, USA, June 9-14, 2013. Proceedings*, volume 7898 of *Lecture Notes in Computer Science*, pages 249–266. Springer, 2013.
 - 15 Serdar Erbatur, Deepak Kapur, Andrew M. Marshall, Paliath Narendran, and Christophe Ringeissen. Unification and matching in hierarchical combinations of syntactic theories. In Carsten Lutz and Silvio Ranise, editors, *Frontiers of Combining Systems - 10th International Symposium, FroCoS 2015, Wroclaw, Poland, September 21-24, 2015. Proceedings*, volume 9322 of *Lecture Notes in Computer Science*, pages 291–306. Springer, 2015.
 - 16 Serdar Erbatur, Andrew M. Marshall, Deepak Kapur, and Paliath Narendran. Unification over distributive exponentiation (sub)theories. *J. Autom. Lang. Comb.*, 16(2-4):109–140, 2011.
 - 17 Serdar Erbatur, Andrew M. Marshall, and Christophe Ringeissen. Notions of knowledge in combinations of theories sharing constructors. In Leonardo de Moura, editor, *Automated Deduction - CADE 26 - 26th International Conference on Automated Deduction, Gothenburg, Sweden, August 6-11, 2017, Proceedings*, volume 10395 of *Lecture Notes in Computer Science*, pages 60–76. Springer, 2017.
 - 18 Serdar Erbatur, Andrew M. Marshall, and Christophe Ringeissen. Terminating non-disjoint combined unification. In Maribel Fern andez, editor, *Logic-Based Program Synthesis and Transformation - 30th International Symposium, LOPSTR 2020, Bologna, Italy, September 7-9, 2020, Proceedings*, volume 12561 of *Lecture Notes in Computer Science*, pages 113–130. Springer, 2020.
 - 19 Serdar Erbatur, Andrew M. Marshall, and Christophe Ringeissen. Non-disjoint combined unification and closure by equational paramodulation. In Boris Konev and Giles Reger, editors,

- Frontiers of Combining Systems - 13th International Symposium, FroCoS 2021, Birmingham, UK, September 8-10, 2021, Proceedings*, volume 12941 of *Lecture Notes in Computer Science*, pages 25–42. Springer, 2021.
- 20 Santiago Escobar, Catherine A. 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, Tutorial Lectures*, volume 5705 of *Lecture Notes in Computer Science*, pages 1–50. Springer, 2007.
 - 21 Jean-Pierre Jouannaud and Hélène Kirchner. Completion of a set of rules modulo a set of equations. *SIAM J. Comput.*, 15(4):1155–1194, 1986.
 - 22 Deepak Kapur, Paliath Narendran, and Lida Wang. An E-unification algorithm for analyzing protocols that use modular exponentiation. In Robert Nieuwenhuis, editor, *Rewriting Techniques and Applications, 14th International Conference, RTA 2003, Valencia, Spain, June 9-11, 2003, Proceedings*, volume 2706 of *Lecture Notes in Computer Science*, pages 165–179. Springer, 2003.
 - 23 Claude Kirchner and Francis Klay. Syntactic theories and unification. In *Proceedings of the Fifth Annual Symposium on Logic in Computer Science (LICS '90), Philadelphia, Pennsylvania, USA, June 4-7, 1990*, pages 270–277. IEEE Computer Society, 1990.
 - 24 Catherine Meadows and Paliath Narendran. A unification algorithm for the group Diffie-Hellman protocol. In *Informal Proceedings of the Workshop on Issues in the Theory of Security (WITS)*, 2002.
 - 25 Simon Meier, Benedikt Schmidt, Cas Cremers, and David A. Basin. The TAMARIN prover for the symbolic analysis of security protocols. In Natasha Sharygina and Helmut Veith, editors, *Computer Aided Verification - 25th International Conference, CAV 2013, Saint Petersburg, Russia, July 13-19, 2013. Proceedings*, volume 8044 of *Lecture Notes in Computer Science*, pages 696–701. Springer, 2013.
 - 26 Pierre-Etienne Moreau, Christophe Ringeissen, and Marian Vittek. A pattern matching compiler for multiple target languages. In Görel Hedin, editor, *Compiler Construction, 12th International Conference, CC 2003, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2003, Warsaw, Poland, April 7-11, 2003, Proceedings*, volume 2622 of *Lecture Notes in Computer Science*, pages 61–76. Springer, 2003.
 - 27 Paliath Narendran. Solving linear equations over polynomial semirings. In *Proceedings, 11th Annual IEEE Symposium on Logic in Computer Science, New Brunswick, New Jersey, USA, July 27-30, 1996*, pages 466–472. IEEE Computer Society, 1996.
 - 28 Tobias Nipkow. Proof transformations for equational theories. In *Proceedings of the Fifth Annual Symposium on Logic in Computer Science (LICS '90), Philadelphia, Pennsylvania, USA, June 4-7, 1990*, pages 278–288. IEEE Computer Society, 1990.
 - 29 Tobias Nipkow. Combining matching algorithms: The regular case. *J. Symb. Comput.*, 12(6):633–654, 1991.
 - 30 Christophe Ringeissen. Unification in a combination of equational theories with shared constants and its application to primal algebras. In Andrei Voronkov, editor, *Logic Programming and Automated Reasoning, International Conference LPAR'92, St. Petersburg, Russia, July 15-20, 1992, Proceedings*, volume 624 of *Lecture Notes in Computer Science*, pages 261–272. Springer, 1992.
 - 31 Christophe Ringeissen. Combining decision algorithms for matching in the union of disjoint equational theories. *Inf. Comput.*, 126(2):144–160, 1996.
 - 32 Christophe Ringeissen. Matching in a class of combined non-disjoint theories. In Franz Baader, editor, *Automated Deduction - CADE-19, 19th International Conference on Automated Deduction Miami Beach, FL, USA, July 28 - August 2, 2003, Proceedings*, volume 2741 of *Lecture Notes in Computer Science*, pages 212–227. Springer, 2003.
 - 33 Christophe Ringeissen. Building and combining matching algorithms. In Carsten Lutz, Uli Sattler, Cesare Tinelli, Anni-Yasmin Turhan, and Frank Wolter, editors, *Description Logic, Theory Combination, and All That - Essays Dedicated to Franz Baader on the Occasion of His*

60th Birthday, volume 11560 of *Lecture Notes in Computer Science*, pages 523–541. Springer, 2019.

- 34 Manfred Schmidt-Schauß. Unification in a combination of arbitrary disjoint equational theories. *J. Symb. Comput.*, 8(1/2):51–99, 1989.
- 35 Cesare Tinelli and Christophe Ringeissen. Unions of non-disjoint theories and combinations of satisfiability procedures. *Theor. Comput. Sci.*, 290(1):291–353, 2003.

A Technical Appendix

Let us first introduce an ordering on $T(\Sigma, C)$ that will be useful in our proofs. This ordering reuses the classical LPO reduction ordering [2] which is defined with respect to a precedence. Actually, any total ordering on $T(\Sigma, C)$ would work provided that it is stable by $>$ -compatible renaming. Assume an arbitrary total ordering $>_{\Sigma \cup C}$ on $\Sigma \cup C$ such that the restriction of $>_{\Sigma \cup C}$ to C is $>$, and all the symbols in Σ are greater than all the symbols in C w.r.t $>_{\Sigma \cup C}$. For any $s, t \in T(\Sigma, C)$, we write $s >_{LPO} t$ if s is greater than t w.r.t the LPO ordering whose precedence is given by the restriction of $>_{\Sigma \cup C}$ to $\Sigma \cup Cst(s) \cup Cst(t)$.

In a straightforward way, any F -canonizer stable by renaming corresponds to an idempotent mapping from $T(\Sigma, C)$ to $T(\Sigma, C)$, also denoted by w , such that for any $s, t \in T(\Sigma, C)$, $s =_F t$ iff $w(s) = w(t)$; for any $t \in T(\Sigma, C)$, $Cst(w(t)) \subseteq Cst(t)$ and for any $>$ -compatible renaming ξ of $Cst(t) \cup \{c_0\}$ with $c_0\xi = c_0$, $w(t\xi) = w(t)\xi$.

► **Definition 45.** Let F be an equational Σ -theory, and w an F -canonizer stable by renaming. Given any terms $t, t' \in T(\Sigma, C)$, we define $t >_w t'$ if $w(t) >_{LPO} w(t')$.

► **Lemma 46.** The ordering $>_w$ given in Definition 45 satisfies the following properties:

- $>_w$ is F -compatible.
- For any $t, t' \in T(\Sigma, C)$, we have that either $t =_F t'$ or $t >_w t'$ or $t' >_w t$.
- For any $t, t' \in T(\Sigma, C)$ and any $>$ -compatible renaming ξ of $Cst(t) \cup Cst(t') \cup \{c_0\}$ with $c_0\xi = c_0$, we have $t\xi >_w t'\xi$ iff $t >_w t'$.

Proof. Consider any $u, t, t', u' \in T(\Sigma, C)$.

- $u =_F t >_w t' =_F u'$ implies $w(u) = w(t) >_{LPO} w(t') = w(u')$, and so $u >_w u'$.
- Since $>_{LPO}$ is total, we have $t >_w t'$ or $t' >_w t$ for any t, t' such that $t \neq_F t'$.
- For any $>$ -compatible renaming ξ of $Cst(t) \cup Cst(t') \cup \{c_0\}$ with $c_0\xi = c_0$, we have $t\xi >_w t'\xi$ iff $w(t\xi) >_{LPO} w(t'\xi)$ iff $w(t)\xi >_{LPO} w(t')\xi$. Due to the chosen precedence for the LPO ordering, we have $w(t)\xi >_{LPO} w(t')\xi$ iff $w(t) >_{LPO} w(t')$. Thus, $t\xi >_w t'\xi$ iff $t >_w t'$. ◀

A.1 Theorems

In the next two proofs, we use an additional notion of constant abstraction mapping:

► **Definition 47.** Assume F is an equational Σ -theory, Cst is a finite subset of C , AT is a finite subset of $T(\Sigma, C) \setminus C$ such that $(\bigcup_{u \in AT} Cst(u)) \subseteq Cst$, NC is a finite subset of $C \setminus (Cst \cup \{c_0\})$, and \gg is an F -compatible ordering which is total on AT . A mapping $\Pi : AT \rightarrow NC$ is said to be a $(\gg, =_F)$ -ordered constant abstraction mapping with a range out of Cst if for any $u, v \in AT$, $\Pi(u) > \Pi(v)$ iff $u \gg v$ and $\Pi(u) = \Pi(v)$ iff $u =_F v$. Under these assumptions, Π^{-1} is any arbitrary morphism from NC to AT such that for any $u \in AT$, $(\Pi(u))\Pi^{-1} =_F u$. For any term $t \in T(\Sigma, C) \setminus C$ such that $Cst(t) \subseteq Cst$, t^Π denotes the term obtained from t by replacing any subterm u of t occurring in AT by $\Pi(u)$.

For any $\Sigma_i \setminus \Sigma_0$ -rooted term t , the set of Σ_i -alien subterms of t is denoted by $Alien(t)$.

Proof of Theorem 8. Given two E -constructed normalizing mappings NF_1 and NF_2 for $F_1 \cup E$ and $F_2 \cup E$ respectively, we show how to combine them in order to construct an E -constructed normalizing mappings $NF_{1,2}$ for $F_1 \cup F_2 \cup E$ in a way $NF_{1,2}$ coincides with NF_i on $T(\Sigma_i, C)$ for any $i = 1, 2$.

Consider w is any E -canonizer stable by renaming. Since w does not need to be computable, such a mapping always exists. Then, $>_w$ is the ordering given in Definition 45.

$NF_{1,2}$ is inductively defined as follows:

- For any $c \in C$, $NF_{1,2}(c) = c$.
- Let t be any Σ_0 -rooted term of the form $f(t_1, \dots, t_m)$. Then, we define $NF_{1,2}(t) = f(NF_{1,2}(t_1), \dots, NF_{1,2}(t_m))$.
- Let t be any $\Sigma_i \setminus \Sigma_0$ -rooted term. If $Alien(t) = \emptyset$, then $NF_{1,2}(t) = NF_i(t)$. Otherwise, let t' be the term obtained from t by replacing each $u \in Alien(t)$ by $NF_{1,2}(u)$. If $Alien(t') = \emptyset$, then $NF_{1,2}(t) = NF_i(t')$. Otherwise, let $\Pi : Alien(t') \rightarrow NC$ be a $(>_w, =_E)$ -ordered constant abstraction mapping with a range out of $Cst(t')$. We define $NF_{1,2}(t) = (NF_i(t'^{\Pi}))\Pi^{-1}$.

One can check that $NF_{1,2}$ inherits all the properties stating that NF_1 and NF_2 are E -constructed normalizing mappings, including the property that NF is stable by $>$ -compatible renaming (third item of Definition 2) thanks to Lemma 46. ◀

Proof of Theorem 13. Let NF be the E -constructed normalizing mapping obtained from NF_1 and NF_2 by applying Theorem 8. Just like any layer-reduced term mapping, it is sufficient to define $\Downarrow_{1,2}$ on $(\Sigma_1 \cup \Sigma_2) \setminus \Sigma_0$ -rooted terms. Then, $\Downarrow_{1,2}$ uniquely extends to $\Sigma_0 \cup C$ -rooted terms. The definition of $\Downarrow_{1,2}$ bears similarities with the construction of NF detailed in the proof of Theorem 8.

For any $\Sigma_i \setminus \Sigma_0$ -rooted term t , $t\Downarrow_{1,2}$ is inductively defined as follows:

If $Alien(t) = \emptyset$, then $t\Downarrow_{1,2} = t\Downarrow_i$. Otherwise, let t' the term obtained from t by replacing each $u \in Alien(t)$ by $u\Downarrow_{1,2}$. If $Alien(t') = \emptyset$, then $t\Downarrow_{1,2} = t'\Downarrow_i$. Otherwise, let $\Pi : Alien(t') \rightarrow NC$ be a $(>_{\Downarrow_{1,2}}, =_{F_1 \cup F_2 \cup E})$ -ordered constant abstraction mapping with a range out of $Cst(t')$. We define $t\Downarrow_{1,2} = ((t'^{\Pi})\Downarrow_i)\Pi^{-1}$ if $(t'^{\Pi})\Downarrow_i \neq t'^{\Pi}$, otherwise $t\Downarrow_{1,2} = t'$.

The $>_{\Downarrow_{1,2}}$ ordering used above is inductively defined as follows:

- Let s, t be any $\Sigma_i \setminus \Sigma_0$ -rooted terms such that $s\Downarrow_{1,2} = s$ and $t\Downarrow_{1,2} = t$. If $Alien(s) = Alien(t) = \emptyset$, then $s >_{\Downarrow_{1,2}} t$ iff $s >_{\Downarrow_i} t$. Otherwise, let $\Pi : Alien(s) \cup Alien(t) \rightarrow NC$ be a $(>_{\Downarrow_{1,2}}, =_{F_1 \cup F_2 \cup E})$ -ordered constant abstraction mapping with a range out of $Cst(s) \cup Cst(t)$. We define $s >_{\Downarrow_{1,2}} t$ iff $s^{\Pi} >_{\Downarrow_i} t^{\Pi}$.
- Let s be any $\Sigma_2 \setminus \Sigma_0$ -rooted term such that $s\Downarrow_{1,2} = s$ and t any $\Sigma_2 \setminus \Sigma_0$ -rooted term such that $t\Downarrow_{1,2} = t$, we define $s >_{\Downarrow_{1,2}} t$ (this choice is arbitrary).

According to our assumptions on the stability by renaming of both \Downarrow_i and $>_{\Downarrow_i}$, it is important to note that $\Downarrow_{1,2}$ and $>_{\Downarrow_{1,2}}$ are well-defined since we get the same results independently from the chosen Π . Then, we can prove the following statements:

- For any $\Sigma_i \setminus \Sigma_0$ -rooted term $t \in T(\Sigma_1 \cup \Sigma_2, C)$ such that $t\Downarrow_{1,2} = t$, t is a term with true i -aliens, $t\Downarrow_{NF}$ is $\Sigma_i \setminus \Sigma_0$ -rooted, and a renaming of t^{π_i} can be effectively built.
- For any $t \in T(\Sigma_1 \cup \Sigma_2, C)$, $t\Downarrow_{1,2}$ is a computable layer-reduced form of t associated to NF modulo $F_1 \cup F_2 \cup E$.

These statements are proved by induction using the height of layers of a term $t \in T(\Sigma_1 \cup \Sigma_2, C)$, denoted by $hl(t)$ and defined as follows:

- If t is a Σ_0 -rooted term $f(t_1, \dots, t_m)$, then $hl(t) = \max_{k=1, \dots, m} hl(t_k)$.
- If t is $\Sigma_i \setminus \Sigma_0$ -rooted, then (if $Alien(t) \neq \emptyset$, then $hl(t) = 1 + \max_{u \in Alien(t)} hl(u)$, else $hl(t) = 0$).
- If $t \in C$, then $hl(t) = 0$.

Eventually, the decidability of $F_1 \cup F_2 \cup E$ -equality is a direct consequence of Lemma 48 (cf. Section A.2). \blacktriangleleft

Proof of Theorem 27. First of all, note that an E -convergent TRS (R, E) over the signature Σ is innermost-resolvent iff for any $s \in T(\Sigma, V)$, any innermost derivation $s \rightarrow_{R,E}^* s \downarrow_{R,E}$ includes at most one rewrite step applied at the root position. This holds because any innermost derivation $s \rightarrow_{R,E}^* t$ can be extended to an innermost derivation $s \rightarrow_{R,E}^* t \rightarrow_{R,E}^* s \downarrow_{R,E}$.

Let us now check that $(R_1 \cup R_2, E)$ is E -convergent. First, $\rightarrow_{R_1 \cup R_2}$ is assumed to be E -terminating. Second, $(R_1 \cup R_2, E)$ is Church-Rosser modulo E since both (R_1, E) and (R_2, E) are E -constructed. Consequently, $(R_1 \cup R_2, E)$ is E -convergent.

Consider an innermost derivation $s \rightarrow_{R_1 \cup R_2, E}^* s \downarrow_{R_1 \cup R_2, E}$, where s is assumed to be $\Sigma_i \setminus \Sigma_0$ -rooted for any $i = 1, 2$. This innermost derivation can be divided in two parts. First, we normalize all the alien subterms of s , leading to a term t whose aliens are now normalized. Second, we normalize t until $s \downarrow_{R_1 \cup R_2, E}$ is reached. Thus, we have an innermost derivation $s \rightarrow_{R_1 \cup R_2, E}^* t \rightarrow_{R_1 \cup R_2, E}^* s \downarrow_{R_1 \cup R_2, E}$. All the rules in the innermost derivation $t \rightarrow_{R_1 \cup R_2, E}^* s \downarrow_{R_1 \cup R_2, E}$ are necessarily rules from R_i because t is a Σ_i -rooted term whose alien subterms are normalized, and so the alien subterms remain in the substitution part of any rule application. Consequently, $t \rightarrow_{R_i, E}^* s \downarrow_{R_1 \cup R_2, E}$. Since (R_i, E) is innermost-resolvent, $t \rightarrow_{R_i, E}^* s \downarrow_{R_1 \cup R_2, E}$ includes at most one rewrite step applied at the root position. Moreover, all the rules in $s \rightarrow_{R_1 \cup R_2, E}^* t$ are applied below the root position. Consequently, $s \rightarrow_{R_1 \cup R_2, E}^* t \rightarrow_{R_i, E}^* s \downarrow_{R_1 \cup R_2, E}$ includes at most one rewrite step applied at the root position, and so $(R_1 \cup R_2, E)$ is innermost-resolvent. \blacktriangleleft

A.2 Lemmas

Proof of Lemma 6. Let $\rightarrow_{R_{NF}/E}$ be $=_E \circ \rightarrow_{R_{NF}} \circ =_E$. By definition of R_{NF} , $\rightarrow_{R_{NF}/E}$ is an optimally reducing rewrite relation where the length of any derivation starting from any $t \in T(\Sigma, C)$ is bounded by the size of t , and so $\rightarrow_{R_{NF}/E}$ is terminating. For any $t \in T(\Sigma, C)$, $t \rightarrow_{R_{NF}/E}^* NF(t)$. Let us check that $=_{F \cup E}$ coincides with $\leftarrow_{R_{NF} \cup E}^*$. For any $s, t \in T(\Sigma, C)$, $s =_{F \cup E} t$ implies $NF(s) =_E NF(t)$ where $s \rightarrow_{R_{NF}/E}^* NF(s)$ and $t \rightarrow_{R_{NF}/E}^* NF(t)$. Thus, $=_{F \cup E} \subseteq \leftarrow_{R_{NF} \cup E}^*$. Conversely, $\leftarrow_{R_{NF} \cup E}^* \subseteq =_{F \cup E}$ since for any $l \rightarrow r \in R_{NF}$, $l =_{F \cup E} r$. For any peak $s \leftarrow_{R_{NF}, E} t \rightarrow_{R_{NF}, E} s'$, we have $s \rightarrow_{R_{NF}/E}^* NF(s) =_E NF(s') \leftarrow_{R_{NF}/E}^* s'$. Since Σ_0 is a set of constructors for R_{NF} , $\rightarrow_{R_{NF}, E}$ is E -coherent, and the proof $s \rightarrow_{R_{NF}/E}^* \circ =_E \circ \leftarrow_{R_{NF}/E}^* s'$ can be turned into $s \rightarrow_{R_{NF}, E}^* \circ =_E \circ \leftarrow_{R_{NF}, E}^* s'$. Thus, $\rightarrow_{R_{NF}, E}$ is locally E -confluent. Thanks to [21], $\rightarrow_{R_{NF}, E}$ is E -convergent, and more precisely E -constructed. \blacktriangleleft

Proof of Lemma 10. If t is $\Sigma_{3-i} \setminus \Sigma_0$ -rooted, then $t^{\pi_i} = \pi(t \downarrow_{NF}) = \pi((t \downarrow_{NF}) \downarrow_{NF}) = (t \downarrow_{NF})^{\pi_i}$.

For any Σ_i -rooted t , let t' be the term obtained from t by replacing each Σ_i -alien subterm u of t by $u \downarrow_{NF}$. To prove that $(t')^{\pi_i} =_{F_i \cup E} (t \downarrow_{NF})^{\pi_i}$ follows from the construction of $NF_{1,2}$, consider $R = R_{NF_{1,2}}$ and $R_i = \{l \rightarrow r \mid l(\epsilon) \in \Sigma_i \setminus \Sigma_0, l \rightarrow r \in R\}$ for $i = 1, 2$. We have the following property: if s is a Σ_i -rooted term such that its Σ_i -alien subterms are $NF_{1,2}$ -normalized, then $s \rightarrow_{R, E} s'$ implies $s \rightarrow_{R_i, E} s'$, s' is either $NF_{1,2}$ -normalized or a Σ_i -rooted term such that its Σ_i -alien subterms are $NF_{1,2}$ -normalized, and in both cases $s^{\pi_i} =_{F_i \cup E} (s')^{\pi_i}$, by definition of R_i . Then, by induction on the length of the derivation $t' \rightarrow_{R, E}^* t \downarrow_{NF}$ we prove that $(t')^{\pi_i} =_{F_i \cup E} (t \downarrow_{NF})^{\pi_i}$. Finally, we get $t^{\pi_i} =_{F_i \cup E} (t \downarrow_{NF})^{\pi_i}$ since $t^{\pi_i} = (t')^{\pi_i}$. \blacktriangleleft

6:22 Combined Hierarchical Matching

Proof of Lemma 14. Let w be a computable $F \cup E$ -canonizer stable by renaming, and $>_w$ the corresponding ordering introduced in Definition 45. Given any terms $t, t' \in T_{\Downarrow}$, we define $t >_{\Downarrow} t'$ if $t >_w t'$. Thus, $>_{\Downarrow}$ is computable. By Lemma 46, $>_{\Downarrow}$ fulfills all the properties of a \Downarrow -ordering as given in Definition 12. \blacktriangleleft

Proof of Lemma 29. If t is $\Sigma \setminus \Sigma_0$ -rooted, then $t^{\pi_0} = \pi(t \downarrow_{NF}) = \pi((t \downarrow_{NF}) \downarrow_{NF}) = (t \downarrow_{NF})^{\pi_0}$. Assume now t is Σ_0 -rooted. Let t' be the term obtained from t by replacing each Σ_0 -alien subterm u of t by $u \downarrow_{NF}$. According to Definition 2, we have $(t')^{\pi_0} =_E (t \downarrow_{NF})^{\pi_0}$. Since $t^{\pi_0} = (t')^{\pi_0}$, we get $t^{\pi_0} =_E (t \downarrow_{NF})^{\pi_0}$. \blacktriangleleft

Proof of Lemma 30.

- Consider s and t are Σ_0 -rooted. Then, $s \downarrow_{NF}$ and $t \downarrow_{NF}$ are Σ_0 -rooted, and we have the following equivalences. First, $s =_{F \cup E} t$ iff $s \downarrow_{NF} =_E t \downarrow_{NF}$. Second, $s \downarrow_{NF} =_E t \downarrow_{NF}$ iff $(s \downarrow_{NF})^{\pi_0} =_E (t \downarrow_{NF})^{\pi_0}$ by Definition 2. Then, $(s \downarrow_{NF})^{\pi_0} =_E (t \downarrow_{NF})^{\pi_0}$ implies that $s^{\pi_0} =_{F \cup E} t^{\pi_0}$ by Lemma 29. Thus, $s^{\pi_0} =_{F \cup E} t^{\pi_0}$ if $s =_{F \cup E} t$. Conversely, by definition of the 0-abstraction, $s =_{F \cup E} t$ if $s^{\pi_0} =_{F \cup E} t^{\pi_0}$.
- Consider $s, t \in C$. Then, $s =_{F \cup E} t$ iff $s^{\pi_0} = s =_E t = t^{\pi_0}$.
- Consider s is Σ_0 -rooted and t is $\Sigma \setminus \Sigma_0$ -rooted. Assume $s =_{F \cup E} t$. Then, $s \downarrow_{NF} =_E t \downarrow_{NF}$ where $s \downarrow_{NF}$ is Σ_0 -rooted and $t \downarrow_{NF}$ is $\Sigma \setminus \Sigma_0$ -rooted. This is impossible since E is regular collapse-free and the symbols in $\Sigma \setminus \Sigma_0$ do not occur in E .
- Consider $s \in C$ and t is Σ -rooted. Assume $s =_{F \cup E} t$. Then, $s \downarrow_{NF} =_E t \downarrow_{NF}$ where $s \downarrow_{NF} \in C$ and $t \downarrow_{NF}$ is Σ -rooted. This is impossible since E is regular collapse-free and any constant in C can only be E -equal to itself. \blacktriangleleft

The following lemma is similar to Lemma 30. It is used in the proof of Theorem 13.

► **Lemma 48.** *The layer-reduced term mapping $\Downarrow_{1,2}$ associated to $NF_{1,2}$ satisfies the following properties for any $((\Sigma_1 \cup \Sigma_2) \setminus \Sigma_0) \cup C$ -rooted terms s and t such that $s \Downarrow_{1,2} = s$ and $t \Downarrow_{1,2} = t$:*

- if s, t are $\Sigma_i \setminus \Sigma_0$ -rooted for some $i = 1, 2$, then $s =_{F_1 \cup F_2 \cup E} t \Leftrightarrow s^{\pi_i} =_{F_i \cup E} t^{\pi_i}$,
- if $s, t \in C$, then $s =_{F_1 \cup F_2 \cup E} t \Leftrightarrow s = t$,
- otherwise, $s \neq_{F_1 \cup F_2 \cup E} t$.

Proof.

- Consider s and t are $\Sigma_i \setminus \Sigma_0$ -rooted for some $i = 1, 2$. Then, $s \downarrow_{NF}$ and $t \downarrow_{NF}$ are $\Sigma_i \setminus \Sigma_0$ -rooted, and we have the following equivalences. First, $s =_{F_1 \cup F_2 \cup E} t$ iff $s \downarrow_{NF} =_E t \downarrow_{NF}$. Second, $s \downarrow_{NF} =_E t \downarrow_{NF}$ iff $(s \downarrow_{NF})^{\pi_i} =_E (t \downarrow_{NF})^{\pi_i}$ since $s \downarrow_{NF}$ and $t \downarrow_{NF}$ are $\Sigma_i \setminus \Sigma_0$ -rooted, and the symbols in $\Sigma_i \setminus \Sigma_0$ do not occur in E . Then, $(s \downarrow_{NF})^{\pi_i} =_E (t \downarrow_{NF})^{\pi_i}$ implies that $s^{\pi_i} =_{F_i \cup E} t^{\pi_i}$ by Lemma 10. Thus, $s^{\pi_i} =_{F_i \cup E} t^{\pi_i}$ if $s =_{F_1 \cup F_2 \cup E} t$. Conversely, by definition of the i -abstraction, $s =_{F_1 \cup F_2 \cup E} t$ if $s^{\pi_i} =_{F_i \cup E} t^{\pi_i}$.
- Consider $s, t \in C$. Then, $s =_{F_1 \cup F_2 \cup E} t$ iff $s =_E t$ iff $s = t$.
- Consider s is $\Sigma_1 \setminus \Sigma_0$ -rooted and t is $\Sigma_2 \setminus \Sigma_0$ -rooted. Assume $s =_{F_1 \cup F_2 \cup E} t$. Then, $s \downarrow_{NF} =_E t \downarrow_{NF}$ where $s \downarrow_{NF}$ is $\Sigma_1 \setminus \Sigma_0$ -rooted and $t \downarrow_{NF}$ is $\Sigma_2 \setminus \Sigma_0$ -rooted. This is impossible since E is regular collapse-free and the symbols in $(\Sigma_1 \cup \Sigma_2) \setminus \Sigma_0$ do not occur in E .
- Consider $s \in C$ and t is $\Sigma_1 \cup \Sigma_2$ -rooted. Assume $s =_{F_1 \cup F_2 \cup E} t$. Then, $s \downarrow_{NF} =_E t \downarrow_{NF}$ where $s \downarrow_{NF} \in C$ and $t \downarrow_{NF}$ is $\Sigma_1 \cup \Sigma_2$ -rooted. This is impossible since E is regular collapse-free and any constant in C can only be E -equal to itself. \blacktriangleleft