A Comprehensive Combination Framework

SILVIO GHILARDI and ENRICA NICOLINI
Università degli Studi di Milano
and
DANIELE ZUCCHELLI
Università degli Studi di Milano and LORIA & INRIA-Lorraine

We define a general notion of a fragment within higher-order type theory; a procedure for constraint satisfiability in combined fragments is outlined, following Nelson-Oppen schema. The procedure is in general only sound, but it becomes terminating and complete when the shared fragment enjoys suitable noetherianity conditions and admits an abstract version of a 'Keisler-Shelah like' isomorphism theorem. We show that this general decidability transfer result covers recent work on combination in first-order theories as well as in various intensional logics such as description, modal, and temporal logics.

Categories and Subject Descriptors: F.4.1 [**Mathematical Logic and Formal Languages**]: Mathematical Logic—*Mechanical theorem proving*; I.2.3 [**Artificial Intelligence**]: Deduction and Theorem Proving—*Inference engines*; I.2.4 [**Artificial Intelligence**]: Knowledge Representation Formalisms and Methods—*Modal logic*; *Predicate logic*

General Terms: Logic, Algorithms, Decidability

Additional Key Words and Phrases: Combination, Decision Procedures, Higher-Order Logic, Modal and Description Logics, Satisfiability Modulo Theory

1. INTRODUCTION AND GLOBAL OVERVIEW

Decision procedures for fragments of various logics and theories play a central role in many applications of logic in computer science, for instance in formal methods and in knowledge representation. Within these application domains, relevant data appear to be heterogeneously structured, so that modularity in combining and reusing both algorithms and concrete implementations becomes crucial. This is why the development of meta-level frameworks, accepting as input specialized devices, turns out to be of strategic importance for future advances in building powerful, fully or partially automated systems. In this paper, we shall consider one of the most

Research supported in part by the MIUR project "Constructive Methods in Topology, Algebra, and Foundations of Computer Science".

Authors' addresses: Silvio Ghilardi and Daniele Zucchelli, Dipartimento di Scienze dell'Informazione, Università degli Studi di Milano. Via Comelico 39/41, 20135 Milan, Italy. E-mail: {ghilardi,zucchelli}@dsi.unimi.it. Enrica Nicolini, Dipartimento di Matematica, Università degli Studi di Milano. Via Saldini 50, 20133 Milan, Italy. E-mail: nicolini@mat.unimi.it.

A preliminary version of this paper appears in the Proceedings of the 5th International Workshop on Frontiers of Combining Systems (FroCoS 2005).

popular and simple schemata (due to Nelson-Oppen) for designing a cooperation protocol among separate reasoners, we plug it into a higher-order framework, and show how it can be used to cope with various classes of combination problems, often quite far from the originally intended application domain. In this introduction, we incrementally explain the intuitions and the ideas underlying the whole plan of the paper.

1.1 Nelson-Oppen Method: the Universal First-Order Case

The *Nelson-Oppen method* [Nelson and Oppen 1979; Oppen 1980; Tinelli and Harandi 1996] was originally designed in order to combine decision procedures for the *universal fragment* of *first-order theories*; this is the kind of problems arising in software verification, but on the other hand it should be noted that most standard topics in computational algebra [Chenadec 1986] concern equality in finitely presented algebras, and hence they can be equivalently reformulated as decision problems for universal fragments of first-order equational theories.

Recall that deciding the universal fragment of a first-order language for validity is equivalent to deciding *constraints* (i.e. finite conjunction of literals) for satisfiability. The basic feature of Nelson-Oppen method is quite simple: constraints involving mixed signatures are purified into two equisatisfiable pure constraints and then the specialized reasoners try to share all the information they can acquire concerning constraints in the common subsignature, till an inconsistency is detected or a saturation state is reached. We illustrate the procedure by an example:

EXAMPLE 1.1. Suppose that theory T_1 is Presburger arithmetic and that theory T_2 is the theory of two uninterpreted function symbols f, g (f is unary and g is binary). We want to check unsatisfiability modulo $T_1 \cup T_2$ of the constraint

$$\Gamma \equiv \{x + f(y) = x,\ g(f(y) + z, z) \neq g(z, z)\}.$$

We use two decision procedures for satisfiability of literals modulo T_1 and T_2, as black boxes. In the *first step*, Nelson-Oppen method repeatedly abstracts out alien subterms with fresh variables, till an equisatisfiable finite set of literals $\Gamma_1 \cup \Gamma_2$ is produced, where Γ_1 contains only literals in the signature of T_1 and Γ_2 contains only literals in the signature of T_2. In practice, subterms t are replaced by fresh variables x and new equations $x = t$ are added to the current constraint, till the the desired purified status is reached: in the present example, we get

$$\Gamma_1 \equiv \{x + w = x,\ u = w + z\}, \qquad \Gamma_2 = \{w = f(y),\ g(u, z) \neq g(z, z)\}.$$

In the *second step*, information exchange concerning the common subsignature is performed. In our case, for instance, the decision procedure for T_1 realizes that $u = z$ is a logical consequence of $T_1 \cup \Gamma_1$; as soon as the decision procedure for T_2 knows this fact, it reports the inconsistency. Notice that this example is very simple: in general, the exchange of entailed atoms from the common subsignature is not sufficient, one needs to exchange entailed *disjunctions* of atoms (such an exchange of disjunctions of atoms may be implemented for instance by case-split and backtracking).

There are two main problems that must be adequately addressed in this Nelson-Oppen approach, namely termination and completeness of the proposed combined procedure. *Termination* can be guaranteed in case the total amount of exchangeable information is finite, i.e. in case there are only finitely many ‘representative’ atoms in the common subsignature (this is certainly the case if signatures are disjoint). Although this is the most frequently used method to enforce termination, weaker requirements might be sufficient: we shall see in the paper that it is sufficient to assume a 'noetherianity' condition, i.e. it is sufficient to assume finiteness of properly ascending chains of positive constraints. *Completeness* is however the most serious problem: since it might happen that constraint satisfiability in $T_1 \cup T_2$ is undecidable even if constraint satisfiability in T_1 and T_2 are decidable (and even if the signatures of T_1 and T_2 are disjoint, as recently shown in [Bonacina et al. 2006]), conditions for completeness are indispensable. In fact, Nelson-Oppen method was guaranteed to be complete only for disjoint signatures and stably infinite theories,[1] till quite recently, when it was realized [Ghilardi 2004] that stable infiniteness is just a special case of a compatibility notion, which is related to model completions of shared sub-theories. The results of this paper will cover (and also strengthen to the noetherian case) such recent completeness results, as shown by Theorem 5.7 below. For the moment, however, we leave apart termination and completeness issues and concentrate on the *design of extensions* of Nelson-Oppen procedure beyond the case of universal fragments of first-order theories.

1.2 Nelson-Oppen Method: from First-Order to Higher-Order Logic

People working on decidability transfer results for fusions of modal/description logics (see, e.g., [Wolter 1998; Baader et al. 2002]) elaborated interesting and powerful results by applying specific methodologies: the unifying approach of this paper will demonstrate that such methodologies can often be reduced to Nelson-Oppen, provided the latter is revisited in a suitable general framework.

In order to illustrate the reduction, we examine the case of description logics. To introduce a description language (see [Baader et al. 2003] for more information), we need a set of atomic concepts $\mathrm{x}, \mathrm{y}, \dots$, a set of role names $R, S, \dots$ and a set of individual names $a, b, \dots$; *concepts* are built up from atomic concepts, Boolean operators $\bot, \top, \sqcap, \sqcup, \neg$, and relativized existential quantification $\exists_R$ (here R is a role name). Concepts only notationally differ from propositional multimodal formulae (in modal logic the notation for $\exists_R$ is the 'possibility' operator $\Diamond_R$), however description logics are richer because they allow to write also assertions. We have three kinds of assertions, namely *concepts assertions* $C(a)$ (here a is an individual name and C is a concept), *role assertions* $R(a, b)$ (here a, b are individual names and R is a role name) and *concept equalities* $C = D$ (here C, D are concepts). A finite set of concept assertions or of role assertions is called an A-Box, whereas a finite set of concept equalities is called a T-Box;[2] a pair given by a T-Box and an A-Box is said to be a *knowledge basis*. The semantics for a description language is rather intuitive: an *interpretation* is a pair $\mathcal{I} = (W^{\mathcal{I}}, \cdot^{\mathcal{I}})$, where $W^{\mathcal{I}}$ is a non-empty set (the *domain*) and $\cdot^{\mathcal{I}}$ is the *interpretation function*, assigning to each atomic concept x a subset $\mathrm{x}^{\mathcal{I}} \subseteq W^{\mathcal{I}}$, to each role name R a binary relation $R^{\mathcal{I}} \subseteq W^{\mathcal{I}} \times W^{\mathcal{I}}$, and to every individual name a an element $a^{\mathcal{I}} \in W^{\mathcal{I}}$. The interpretation function is inductively extended to concepts by interpreting the Boolean operators as intersection, union and complement and by interpreting relativized existential quantification as

$$(\exists R.C)^{\mathcal{I}} \ := \ \{w \in W^{\mathcal{I}} \mid \exists v \ (w, v) \in R^{\mathcal{I}} \wedge v \in C^{\mathcal{I}}\}.$$

[1] A theory T is stably infinite iff every constraint in the signature of T which is satisfiable in a model of T is satisfiable in an infinite model of T.

[2] Sometimes, in the literature, concept inclusions are used instead of concept equalities; the difference is immaterial, as far as concepts are closed under intersection.

An interpretation $\mathcal{I}$ satisfies a concept assertion $C(a)$ iff $a^{\mathcal{I}} \in C^{\mathcal{I}}$, it satisfies a role assertion $R(a,b)$ iff $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$ and it satisfies a concept equality $C = D$ iff $C^{\mathcal{I}} = D^{\mathcal{I}}$; $\mathcal{I}$ is a *model* of a knowledge basis iff it satisfies all assertions from it.

There are different and interesting reasoning tasks in description logic; as soon as concepts are closed under all Boolean operators, such reasoning tasks all reduce to satisfiability problems; satisfiability problems can however be formulated at different levels and it is important to keep such distinctions for instance for complexity reasons. For the purposes of this paper, it is useful to distinguish three different satisfiability problems: (i) the *global* satisfiability problem is the problem of deciding whether there exists a model $\mathcal{I}$ of a given T-Box in which we have $C^{\mathcal{I}} \neq \emptyset$ for a preassigned concept C; (ii) the *local* satisfiability problem is the problem of deciding whether there exists a model of a given A-Box; (iii) the *full* satisfiability problem is the problem of deciding whether there exists a model of a knowledge basis. Problem (i) is a notational variant of the relativized satisfiability problem in modal logic, whereas problem (ii) is the same as the local satisfiability problem in modal logic, if we restrict to A-Boxes consisting on a single concept assertion.

It should be noticed that the above definitions just cover the basic description logic $\mathcal{ALC}$. A lot of expressive interesting extensions of $\mathcal{ALC}$ have been considered in the literature [Baader et al. 2003]: one may add some restricted form of counting, constructors (e.g., Boolean constructors) operating on roles, etc. Such extensions sometimes go beyond the limits of a first-order formalism, for various reasons we are going to analyze.

A *first* (obvious) higher-order source is due to the direct use of higher-order constructors: among such constructors, the most popular ones are fixed points. Notice that usually not all fixed points operators of monotonic concepts are added to the language, on the contrary one preferably makes a selected choice of them: for instance, specific fixed points are used in order to capture relativized existential quantification over transitive closure of roles, or to express path quantifiers (the latter is more typical of temporal logics). In description logics, there is also an implicit use of fixed points: in this implicit use, there are no fixed points in the language at all, but T-Boxes are given an alternative 'definitorial' semantics through certain fixed points, see [Baader et al. 2003] for details. A *second* way of passing the expressive limits of first-order logic is semantically driven and arises when only interpretations in some specific structure (or in a non elementary class of structures) are allowed: this is typical of temporal logic, when satisfiability in specific flows of time (e.g., natural, real numbers) is investigated, but there are similar semantic restrictions in description logics too when concrete domains are taken into consideration [Baader and Hanschke 1991].

There is a *third* hidden higher-order feature that is specific for combination purposes. We illustrate it by an example; the example also clarifies why we think that a simple adaptation of the Nelson-Oppen combination procedure analyzed in Subsection 1.1 works here too.

Example 1.2. Suppose our description language contains roles $R, R^*, S, S^{\smile}$ and that we restrict to interpretations in which R^* is the reflexive-transitive closure of R and $S^{\smile}$ is the converse relation of S; suppose also that we know how to process satisfiability problems involving reflexive-transitive closures and converse relations separately, but we do not know how to process a problem involving both of them (or that we do not want to re-implement from scratch a device for solving such combined problems). Consider the following global satisfiability problem

$$\mathrm{y} = \exists_R \exists_R \mathrm{x}, \quad (\forall_{R^*} \neg \mathrm{x}) \sqcap (\exists_S \forall_{S^\smile} \mathrm{y}) \neq \bot$$

(where $\forall$ is defined as $\neg\exists\neg$). To solve the problem, we first purify it as

$$\begin{aligned} &\mathrm{y} = \exists_R \exists_R \mathrm{x}, \quad (\forall_{R^*} \neg \mathrm{x}) \sqcap \mathrm{z} \neq \bot \\ &\mathrm{z} = \exists_S \forall_{S^\smile} \mathrm{y} \end{aligned}$$

and then we begin constraint propagation. In fact, the 'converse relation' procedure discovers the entailed equality $\mathrm{z} \sqcap \mathrm{y} = \mathrm{z}$ and, using this information, the 'reflexive-transitive closure' procedure can report the unsatisfiability of the problem.

There is a clear parallelism between the procedures employed in Examples 1.1 and 1.2: in both cases, abstraction equations are added to the current problem and propagation of shared information is used to detect inconsistency. However in order to make similarities more explicit, we must be able to translate assertions formulated in the description logic formalism into some more standard logical formalism. A first possibility in this sense is offered by the translation into the formalism of Boolean algebras with operators, but this strategy is unable to cover description logic constructors which does not seem to have a clear algebraic counterpart. Alternatively, we can use translations known from modal logic literature. The simplest of such translations is the so-called *standard translation* [van Benthem 1985] of a concept C to a first-order formula $ST(C, w)$ containing only the free variable w: in more detail, concept names x are translated into atomic formulae $X(w)$, boolean connectives are translated identically and $\exists_R C$ is translated into $\exists v(R(w, v) \wedge ST(C, v))$. However, we must take a little step further: we want to represent description logic satisfiability problems as *equations and disequations between terms*. This is important for the design of our Nelson-Oppen procedure: as it is clear from Example 1.1, the procedure uses abstraction variables, adds to the current constraint equations between an abstraction variable and a term, propagates equations from the shared signature, etc. Our solution is simple: we just associate to a concept C the *second-order* term $\{w \mid ST(C, w)\}$ and consider concept names as second order *variables* in satisfiability problems. In this way, we get back the framework of Example 1.1 (modulo some $\beta\eta$-conversions). Let us for instance turn to Example 1.2:

EXAMPLE 1.2. (Continued) The translated constraint is

$$\begin{aligned} &\{w \mid Y(w)\} = \{w \mid \exists w_1(R(w, w_1) \wedge \exists w_2(R(w_1, w_2) \wedge X(w_2)))\}, \\ &\quad \{w \mid \bot\} \neq \{w \mid \forall v(R^*(w, v) \to \neg X(v)) \wedge \exists w_1(S(w, w_1) \wedge \forall w_2(S^\smile(w_1, w_2) \to Y(w_2)))\}. \end{aligned}$$

In order to purify it, we take a fresh variable Z and we modify the constraint as follows

$$\begin{aligned} \{w \mid Y(w)\} &= \{w \mid \exists w_1(R(w, w_1) \wedge \exists w_2(R(w_1, w_2) \wedge X(w_2)))\}, \\ \{w \mid \bot\} &\neq \{w \mid \forall v(R^*(w, v) \to \neg X(v)) \wedge Z(w)\} \\ \{w \mid Z(w)\} &= \{w \mid \exists w_1(S(w, w_1) \wedge \forall w_2(S^\smile(w_1, w_2) \to Y(w_2)))\}. \end{aligned}$$

The third equation is our abstraction equation: in fact $\{w \mid Z(w)\}$ is the long $\beta\eta$-normal form of Z and the substitution $Z \mapsto \{w \mid \exists w_1(S(w, w_1) \wedge \forall w_2(S^\smile(w_1, w_2) \to Y(w_2)))\}$, once applied to the remaining equations, gives back the original constraint (modulo β-conversion). The constraint is now purified and propagation can begin and lead to unsatisfiability as before.

We stress that for the whole mechanism to work, there are some formal aspects to be taken care of: abstraction equations should have the same shape as the remaining part of the constraint, which means that *types of the variables and types of the terms that are left and right members of the (dis)equations in the constraint should match.* This is the reason why we need precise definitions of a fragment and of a constraint within higher-order logic: we consider these definitions to be one of the most important contributions of this paper. Once these definitions are settled, one can easily *design* a suitable higher-order generalization of the Nelson-Oppen procedure and then investigate *sufficient conditions* for it to be terminating and/or complete.

1.3 Constraint Satisfiability in Fragments

We choose Church's type theory as our framework for higher-order logic: thus our syntax deals with types and terms, terms being endowed with a (codomain) type. Types can be built up from primitive sorts by using function type constructor, whereas terms can be built up from typed variables and constants, by using λ-abstraction and function evaluation. Types include the truth-value type Ω and constants include symbols for boolean connectives and for equality over each type. Formulae are treated as special terms, namely terms of type Ω, and quantifiers can be introduced through explicit definitions (see Section 2 for more details). If φ has type Ω and x is a variable (of type, say, τ), we write $\{x \in \tau \mid \varphi\}$ (or simply $\{x \mid \varphi\}$) for the term $\lambda x\, \varphi$ of type $\tau \to \Omega$.

The first task to be accomplished in this framework is to find a definition of what we mean by a fragment (only relatively small fragments can indeed have a chance to be decidable in this context). A fragment should be a pair consisting of a signature for type theory and of a recursive set of terms in that signature; however, the discussion of Subsection 1.2 makes clear that the set of terms must enjoy some minimal properties to make the fragment suitable for our combination purposes. These properties are fixed in the notion of an *algebraic fragment* (Definition 3.2 below): basically in an algebraic fragment $\Phi = \langle \mathcal{L}, T \rangle$, the set of terms T in the signature $\mathcal{L}$ must be closed under composition (i.e. under substitution) and must contain all variables whose type is a Φ-type (a Φ-type is the type of a term in T or of a variable occurring free in some $t \in T$). An algebraic fragment $\Phi = \langle \mathcal{L}, T \rangle$ is *interpreted* when a class $\mathcal{S}$ of (ordinary set-theoretic) models for $\mathcal{L}$ is attached to it ($\mathcal{S}$ is assumed to be closed under isomorphisms). All the algebraic fragments we consider are interpreted and if $\mathcal{S}$ is not specified, it is intended to be the class of all $\mathcal{L}$-structures.

Given an algebraic fragment $\Phi = \langle \mathcal{L}, T \rangle$, a Φ-atom is an equation like $t_1 = t_2$ for terms $t_1, t_2 \in T$ having the same type, a Φ-literal is a Φ-atom or the negation of a Φ-atom, a Φ-clause is a disjunction of Φ-literals and a Φ-constraint is a conjunction of Φ-literals. Now the *word problem* for an interpreted algebraic fragment $\Phi = \langle \mathcal{L}, T, \mathcal{S} \rangle$ is the problem of deciding unsatisfiability of a negative Φ-literal in all $\mathcal{A} \in \mathcal{S}$, whereas the *constraint satisfiability problem* for Φ is the problem of deciding the satisfiability of a Φ-constraint in some $\mathcal{A} \in \mathcal{S}$.

Notice that there are truly higher-order interpreted algebraic fragments whose word problem is decidable (see for instance Friedman theorem for simply typed λ-calculus [Friedman 1975]) and also whose constraint satisfiability problem is decidable (see Rabin results on monadic second order logic over trees [Rabin 1969]). Literature on modal/temporal/description logics provides many examples of interpreted algebraic fragments having decidable word or constraint satisfiability problems (the former usually corresponds to local satisfiability, whereas the latter to global satisfiability, but the precise relationship depends on the specific version of the fragment that is adopted, see the remarks below). The whole literature on computational algebra deals with examples of interpreted algebraic fragments consisting of first-order individual terms with decidable word or conditional word problems (the latter correspond to our constraint satisfiability problems, given the convexity of equational theories).

It should be pointed out that a 'honest' fragment can be formally turned into an algebraic fragment in many ways and this flexibility is crucial when *designing appropriate satisfiability problems* and also when considering *combined algebraic fragments* (these are defined to be the minimum algebraic fragments extending two given ones, see the definition at the beginning of Section 4).

Let us examine, for instance, the case of the fragments corresponding to the description languages illustrated in Subsection 1.2. If we take the fragment consisting of terms that are $\beta\eta$-equivalent to terms of the kind $\{w \mid ST(C, w)\}$, then the only type in this fragment is $W \to \Omega$, and constraint satisfiability in this fragment is equivalent to simultaneous satisfiability of finitely many concepts w.r.t. a given T-Box;[3] if we combine this fragment with another fragment of the same kind (but based on a disjoint description language), we get the fragment corresponding to the fusion (i.e. to the union) of the underlying description languages (this is precisely what happened in Example 1.2).

By contrast, if we take the fragment consisting (besides variables) of terms of the kind $ST(C, w)$ and $R(w, v)$, then the combination with another fragment of the same kind does not produce anything really interesting (just variables can be used to try to form combined terms by replacement); the types in the fragment are now Ω and W, which means that the fragment is ready for non trivial combinations with fragments containing non-variable first-order individual terms, like the fragments considered in Example 1.1 above. From the point of view of satisfiability problems, constraint satisfiability in this fragment is essentially local satisfiability (i.e. satisfiability of an A-Box), see Example 3.11 below for a more detailed analysis.

Finally, for satisfiability of a full knowledge basis (i.e. of an A-Box w.r.t. a T-Box), the fragment comprising (besides variables) all terms $\{w \mid ST(C, w)\}$, $ST(C, w)$, and $R(w, v)$ should be taken into consideration. Such a more complex fragment is ready for a nontrivial combination with another similar fragment as well as with a fragment consisting of first-order individual terms. The moral of this discussion is that an algebraic fragment *must be appropriately designed*, taking into consideration both actual expressive power and potential combination opportunities.

[3] This is the problem of finding a model $\mathcal{I}$ of a given T-Box in which we simultaneously have $C_1^{\mathcal{I}} \neq \emptyset, \ldots, C_n^{\mathcal{I}} \neq \emptyset$. If the fragment is interpreted in a class of structures $\mathcal{S}$ closed under disjoint unions (as it is usually the case), then this simultaneous satisfiability reduces to standard satisfiability (in the global sense). This reduction can be recovered as an instance of a general convexity property for fragments, to be introduced in Subsection 3.4 below.

1.4 Why Nelson-Oppen May Happen to Succeed

Our main goal is the combination of decision procedures for constraint satisfiability. The definition of an algebraic fragment we gave is sufficient to substantially reproduce Nelson-Oppen purification steps in our framework, hence we may freely suppose that combined constraints can be split into two sets of equisatisfiable pure constraints. The interpreted algebraic fragments Φ_1, Φ_2 to be combined share some interpreted algebraic subfragment Φ_0 in the common subsignature and a Nelson-Oppen style fair exchange protocol can then be used. In Subsection 4.2 we describe it in a detailed way, assuming only the availability of two abstractly axiomatized sound 'residue enumerators' for the positive Φ_0-clauses which are entailed by the current Φ_1- (resp. Φ_2-) constraints. The procedure terminates in case certain noetherianity conditions are satisfied (this is always the case whenever Φ_0 is locally finite, namely whenever there are only finitely many non Φ_0-equivalent terms in a fixed finite number of variables).

The main problem we are facing now is the completeness of the procedure: can we infer that the input combined constraint is satisfiable in case the system halts with a saturation message? In general this *cannot* be the case: we already mentioned that taking combined fragments may lead to undecidability, even in the very simple context of Subsection 1.1. The point is that a 'saturation-found' run of our procedure means at the semantic level (see Proposition 4.12 below) that there exist Φ_i-structures $\mathcal{A}_i$ $(i = 1, 2)$, whose Φ_0-reducts are only $\Phi_0(\underline{c}_0)$-equivalent (i.e. such Φ_0-reducts satisfy the same closed Φ_0-atoms in a signature augmented with finitely many free constants $\underline{c}_0$): this is far from being enough to build a combined structure satisfying the union of the pure constraints.

In order to analyze hypotheses under which Nelson-Oppen approach nevertheless succeeds, let us turn to the original Nelson-Oppen case. Here, signatures are one-sorted, first-order, and disjoint; in addition, the stably infiniteness hypotheses on the component theories says that we can safely limit ourselves to consider only infinite structures. Now if we are given *infinite* structures $\mathcal{A}_i$ for the input signatures whose Φ_0-reducts satisfy the same equations among the free constants in $\underline{c}_0$, then it is indeed possible to build a structure for the combined signature out of them. The reasons for this are the following: (i) the theory of an infinite set in the pure equality signature enjoys quantifier elimination, hence from the fact that $\mathcal{A}_1$ and $\mathcal{A}_2$ are $\Phi_0(\underline{c}_0)$-equivalent it follows that they are also elementarily equivalent, as far as the language $\mathcal{L}_0(\underline{c}_0)$ of $\Phi_0(\underline{c}_0)$ is concerned (this is due to standard model-theoretic arguments, see the proof of Theorem 5.7 below); (ii) by Keisler-Shelah theorem [Chang and Keisler 1990], applying ultrapowers to $\mathcal{A}_1$ and $\mathcal{A}_2$, we can make them $\mathcal{L}_0(\underline{c}_0)$-isomorphic and this is clearly sufficient to build the combined structure. Roughly speaking (see Section 5 for details), let us call *isomorphism theorem* a theorem saying that the application of a certain semantic operation makes two $\Phi_0(\underline{c}_0)$-equivalent structures $\mathcal{L}_0(\underline{c}_0)$-isomorphic. The above outlined argument says that the existence of a suitable isomorphism theorem is sufficient for our combined procedure to be complete: this is formally stated in our Main Decidability Transfer Theorem 5.6.

Of course, isomorphisms theorems are quite peculiar and rare. However, another isomorphism theorem justifies the completeness of the procedure in the case of global satisfiability problems for combination - alias fusions - of modal/description logics (this argument applies more generally to combined relativized satisfiability in abstract description systems in the sense of [Baader et al. 2002]). In that case, the shared fragment $\Phi_0(\underline{c}_0)$ is included into the one-variable fragment of first-order classical logic. Here a structure is specified up to $\Phi_0(\underline{c}_0)$-equivalence once we are given the information about emptiness of subsets definable through Boolean combinations of the finitely many unary predicates $\underline{c}_0$. If $\mathcal{A}_1$ and $\mathcal{A}_2$ are $\Phi_0(\underline{c}_0)$-equivalent, then they can be made $\mathcal{L}_0(\underline{c}_0)$-isomorphic by taking disjoint unions, provided that the set indexing the cardinality of the disjoint copies is sufficiently large. Now, if the input modal fragments are interpreted in a semantic class closed under disjoint unions, Theorem 5.6 applies.

Since in the above argument involving disjoint unions, there is nothing very specific to fragments obtained through standard translations of description languages, it is evident that we can get analogous transfer results starting from interpreted guarded and packed guarded fragments [Andréka et al. 1998; Grädel 1999; Marx 2001], which are also preserved under taking disjoint copies of the same structure. Of course, guarded and packed guarded fragments should be designed in such a way that all predicate symbols, except unary ones, are taken to be constants: in other words, second order variables (not to be used in guards) should be used for unary predicates and not for relations (in this way shared fragments are still contained into the monadic fragment of first-order classical logic). But further possible combinations arise, for instance because first-order formulae without equality are also preserved under taking disjoint copies, provided the operation of taking disjoint copies is defined in a proper alternative way (that we will call 'conglomeration' in Subsection 5.3). Theorem 5.21 summarize these new decidability transfer results in a unique statement, referring to the notion of a *monadically suitable* fragment. We also show in Theorem 5.11 how to get decidability transfer results for the combination of an A-Box and of a stably infinite first-order theory, operating on disjoint signatures.

A further application shows how to analyze *monodic modal/temporal fragments* (in the sense of [Wolter and Zakharyaschev 2001; Gabbay et al. 2003]) as combinations of extensional first-order fragments and standard translations of one variable modal/temporal fragments. Since a suitable isomorphism theorem (based on disjoint copies and fiberwise disjoint copies) holds here too, our procedure is complete and justifies rather general decidability transfer results: however, since in order to apply Theorem 5.6 to this case, specific 'descent' techniques are needed, we prefer to leave details for a separated paper (readers may get full information from the Technical Report [Ghilardi et al. 2005] or from the PhD thesis [Nicolini 2006]).

1.5 Related and Future Work

There has been remarkable amount of work in recent literature on decidability transfer results for various kinds of combined satisfiability problems, both within the automated reasoning community and within the modal/description logic community. The closest paper is of course [Ghilardi 2004] where completeness results for first-order non-disjoint Nelson-Oppen procedure are established. The methods in [Ghilardi 2004] gave rise to further contributions. In [Ganzinger et al. 2005] a general schema for combination of decision procedures for first-order theories is proposed: in such a schema, every decision procedure is formalized as an inference system, and their combination is formalized through the so-called inference modules. Applications to fusion decidability transfer in modal logic are already mentioned in [Ghilardi 2004] and are further developed in [Ghilardi and Santocanale 2003]. Additional applications (sometimes involving non trivial extensions of the method as well as integration with other work) concern transfer of decidability of global consequence relation to $\mathcal{E}$-connections [Baader and Ghilardi 2005b; Baader and Ghilardi 2005a], as well as transfer of decidability of local consequence relation to fusions [Baader et al. 2006]. The latter result was rather remarkable and also surprising, not only because the generality of the formulation solved an open problem in modal logic, but also because Nelson-Oppen method is not designed itself to solve word problems and conditions for decidability transfer of word problems were previously formulated in a completely different way for non-disjoint signatures (see [Baader and Tinelli 2002; Fiorentini and Ghilardi 2003]).

Thus, most of previously existing decidability results on fusions of modal logics (for instance those in [Wolter 1998]) were recaptured and sometimes also improved by using general automated reasoning methods based on Nelson-Oppen ideas. However, the standard approach to decision problems in modal/temporal/description logics is directly based on Kripke models (see for instance [Baader et al. 2002; Gabbay et al. 2003]), without the mediation of an algebraic formalism, whereas the mediation of the formalism of Boolean algebras with operators is essential in the approach of papers like [Ghilardi 2004; Ghilardi and Santocanale 2003; Baader et al. 2006; Baader and Ghilardi 2005b]. The appeal to the algebraic formulation of decision problems on one side produces proofs which are much smoother and apply also to semantically incomplete propositional logics, but on the other side it limits the method to the cases in which such a purely algebraic counterpart of semantic decision problems can be identified. For instance, whenever A-Boxes are involved, the results of the present paper go beyond the cases analyzed in [Ghilardi 2004; Ghilardi and Santocanale 2003; Baader et al. 2006; Baader and Ghilardi 2005b].

The main concern of this paper are higher-order signatures, whereas [Ghilardi 2004] deals only with first-order theories, this is the most evident difference between the two papers; however a closer comparison to [Ghilardi 2004] from a strictly technical point of view reveals further substantial novelties. There is a lemma in this paper (Lemma 4.14 below), whose proof closely follows the proof of a similar result (Lemma 9.4) from [Ghilardi 2004]; however the completeness argument for our main Theorem 5.6 is now based on the existence of an isomorphism theorem, instead on Robinson's joint consistency theorem. Moreover, termination is now obtained on the basis of a weaker noetherianity hypothesis which replaces local finiteness from [Ghilardi 2004]. This weaker hypothesis first requires the formulation of a suitable weak compactness property for truly higher-order fragments; in addition, noetherianity is useless if it is not accompanied by suitable devices called *residue enumerators*. In Subsection 3.3, we develop topics concerning residue enumerators from an automated reasoning perspective, in the style of partial theory reasoning. Residue enumerators are eventually integrated within our combination procedure thanks to Proposition 4.8. Notice also that the purification preprocessing steps required by the Nelson-Oppen procedure reveal some specific subtleties in this paper, due to the generality of our context, as shown in Subsection 4.1.

The area of combination methods in automated reasoning is currently receiving increasing interest: modularity and re-usability of existing software give *remarkable additional value* to decision and semi-decision procedures obtained through combination methods, especially in case the combination schema is a rather simple exchange protocol among specialized reasoners. Our higher-order logic approach to combination problems seems to be fruitful, in the sense that it encompasses relevant known results and suggests new applications. We believe that our framework is flexible and open enough to support further substantial extensions: decidability transfer results for $\mathcal{E}$-connections [Kutz et al. 2004] and for description logics with concrete domains [Baader and Hanschke 1991] are just few examples in this sense. In a larger perspective, we hope our framework might also contribute to the *integration* of fully automatized specialized reasoners into *higher-order proof assistants*: indeed, the main point of this paper consists in emphasizing that a rather simple higher-order interface can handle black boxes modules for disparate decision problems.

2. FORMAL PRELIMINARIES

We fix our notation for higher-order syntax, by adopting a type theory in Church's style (see, e.g., [Andrews 2002; Andrews 2001; Lambek and Scott 1988]).

Signatures. Given a set $\mathtt{S}$ (the set of *sorts*), the set of *types over* $\mathtt{S}$ is the set recursively defined as follows: (i) every sort $S \in \mathtt{S}$ is also a type; (ii) Ω is a type (this is called the *truth-values* type); (iii) if τ_1, τ_2 are types, so is $(\tau_1 \to \tau_2)$.

As usual external brackets are omitted; moreover, we shorten the expression $\tau_1 \to (\tau_2 \to \dots (\tau_n \to \tau))$ into $\tau_1 \dots \tau_n \to \tau$. In the following, we use the notation $\mathcal{T}(\mathtt{S})$ or simply $\mathcal{T}$ to indicate a *types set*, i.e. the totality of types that can be built up from the set of sorts $\mathtt{S}$. In this way, $\mathtt{S}$ is sometimes left implicit in the notation, however we always reserve to sorts the letters $S_1, S_2, \dots$ (as opposed to the letters τ, υ, etc. which are used for arbitrary types).

A *signature* is a triple $\mathcal{L} = \langle \mathcal{T}, \Sigma, a \rangle$, where $\mathcal{T}$ is a types set, Σ is a set of constants and a is an *arity map*, namely a map $a : \Sigma \longrightarrow \mathcal{T}$; we write $f : \tau_1 \dots \tau_n \to \tau$ to express that f is a *constant of type* $\tau_1 \dots \tau_n \to \tau$, i.e. that $a(f) = \tau_1 \dots \tau_n \to \tau$. We require *special constants* to be always present in a signature; the special constants are (i) the symbols $\top, \bot, \wedge, \vee, \neg$ denoting the Boolean operations on Ω (their types are the obvious ones); (ii) the equality symbols $=_\tau$ of type $\tau\,\tau \to \Omega$ (we have one such symbol for every type τ, but we write it as '$=$' omitting the subscript τ).

The *proper symbols* of a signature are its sorts and its non special constants. A signature is *one-sorted* iff its set of sorts is a singleton. A signature $\mathcal{L}$ is *first-order* if for any proper $f \in \Sigma$, we have that $a(f) = S_1 \dots S_n \to \tau$, where τ is a sort or it is Ω. A first-order signature is called *relational* iff any proper $f \in \Sigma$ is a *first-order relation*, that is we have $a(f) = S_1 \dots S_n \to \Omega$. By contrast, a first-order signature is called *functional* iff any proper $f \in \Sigma$ has arity $S_1 \dots S_n \to S$.

Terms. Given a signature $\mathcal{L} = \langle \mathcal{T}, \Sigma, a \rangle$ and a type $\tau \in \mathcal{T}$, we define the notion of an $\mathcal{L}$-*term* (or just *term*) *of type* τ - written $t : \tau$ or also t^τ - as follows (for the definition we need countable pairwise disjoint sets V_τ of variables of type τ):

– $x : \tau$ (for $x \in V_\tau$) is an $\mathcal{L}$-term of type τ;
– $c : \tau$ (for $c \in \Sigma$ and $a(c) = \tau$) is an $\mathcal{L}$-term of type τ;
– if $t : \upsilon \to \tau$ and $u : \upsilon$ are $\mathcal{L}$-terms of types $\upsilon \to \tau$ and υ, respectively, then $val_\upsilon(t, u) : \tau$ (also written as $val(t, u) : \tau$ or as $t(u) : \tau$) is an $\mathcal{L}$-term of type τ;
– if $t : \tau$ is an $\mathcal{L}$-term of type τ and $x \in V_\upsilon$ is a variable of type υ, $\lambda x^\upsilon\, t : \upsilon \to \tau$ is an $\mathcal{L}$-term of type $\upsilon \to \tau$.

If it can be deduced from the context, *the specification of the type of a term may be omitted*; moreover we shorten $val(\cdots(val(t, u_1), \cdots), u_n)$ to $t(u_1, \ldots, u_n)$. Terms of type Ω are also called *formulae*; given a formula φ, we write $\{x^\tau \mid \varphi\}$ for $\lambda x^\tau\, \varphi$.

For formulae $\varphi, \varphi_1, \varphi_2$ and for a type υ, we define the formulae $\forall x^\upsilon\, \varphi$, $\exists x^\upsilon\, \varphi$, $\varphi_1 \to \varphi_2$, and $\varphi_1 \leftrightarrow \varphi_2$ to be $\{x^\upsilon \mid \varphi\} = \{x^\upsilon \mid \top\}$, $\neg\forall x^\upsilon \neg\varphi$, $\neg\varphi_1 \vee \varphi_2$, and $(\varphi_1 \to \varphi_2) \wedge (\varphi_2 \to \varphi_1)$, respectively (but notice that $\varphi_1 \leftrightarrow \varphi_2$ can be defined in a semantically equivalent way also as $\varphi_1 = \varphi_2$).

By the above definitions, first-order formulae *can be considered as a subset* of the higher-order formulae defined in this section. More specifically, when we speak of *first-order terms*, we mean variables $x : S$, constants $c : S$ and terms of the kind $f(t_1, \ldots, t_n) : S$, where $t_1, \ldots, t_n$ are (inductively given) first-order terms and $a(f) = S_1 \cdots S_n \to S$. Now *first-order formulae* are obtained from formulae of the kind $\top : \Omega, \bot : \Omega, P(t_1, \ldots, t_n) : \Omega$ (where $t_1, \ldots, t_n$ are first-order terms and $a(P) = S_1 \cdots S_n \to \Omega$) by applying $\exists x^S, \forall x^S, \wedge, \vee, \neg, \to, \leftrightarrow$.

Substitutions and Conversions. Free and bound occurrences of a variable are defined in the usual way. If E is a term or a set of terms, by $fvar(E)$ we mean the set of the variables that occur free in E, whereas $fvar_\tau(E)$ is the set of variables of type τ that occur free in E; the notation $E[x_1, \ldots, x_n]$ means that $fvar(E) \subseteq \{x_1, \ldots, x_n\}$. A term without free variables is called a *closed* term and a formula without free variables is called a *sentence*. Two terms are said to be equivalent modulo *α-conversion* iff they differ only by a bound variables renaming; in the following, we shall identify α-equivalent terms, i.e. we consider terms as representatives of their equivalence class modulo α-conversion.

Let $\mathcal{V}$ be the union of the disjoint sets of variables V_τ ($\tau \in \mathcal{T}$). We define the notion of substitution as usual: a *substitution* is a map $\sigma : \mathcal{V} \to T$ (from the set $\mathcal{V}$ of the variables into the set T of the terms) that respects types (i.e. if $x \in V_\tau$ then $x\sigma$ is a term of type τ) and such that the set $\{x \mid x \not\equiv x\sigma\}$ is finite.[4] The set $dom(\sigma) := \{x \mid x \not\equiv x\sigma\}$ is called the *domain* of the substitution σ. A substitution σ will be written as $x_1 \mapsto x\sigma_1, \ldots, x_n \mapsto x_n\sigma$, where $dom(\sigma) \subseteq \{x_1, \ldots, x_n\}$. A substitution is a *renaming* iff it is a variable permutation.

Substitutions can be extended in the domain from variables to all terms in the usual way; notice however that, when defining inductively the term $t\sigma$, it might happen that α-conversions must be applied before actual replacements, in order to avoid clashes. If $\sigma = \{x_1 \mapsto u_1, \ldots, x_n \mapsto u_n\}$ and $fvar(t) \subseteq \{x_1, \ldots, x_n\}$, the term $t\sigma$ can also be written as $t[u_1, \ldots, u_n]$.

We assume the reader is familiar with the notion of *$\beta\eta$-equivalence* $\sim_{\beta\eta}$ between

[4]Since the equality symbol '=' is present in the object language, we prefer to use '$\equiv$' in the metalanguage for coincidence of syntactic expressions.

terms and of *(long)-$\beta\eta$-normal form* of a term (see, e.g., [Dowek 2001] for a quick brief account to what is needed for this paper).

Models. In order to introduce our computational problems, we need to recall the notion of an interpretation of a type-theoretic signature; for the purposes of this paper, we just need standard set-theoretic interpretations.

If we are given a map that assigns to every sort $S \in \mathtt{S}$ a set $[\![S]\!]$, we can inductively extend it to all types over $\mathtt{S}$, by taking $[\![\tau \to \upsilon]\!]$ to be the set of functions from $[\![\tau]\!]$ to $[\![\upsilon]\!]$. Given a signature $\mathcal{L} = \langle \mathcal{T}, \Sigma, a \rangle$, a *$\mathcal{L}$-structure* (or just a *structure*) $\mathcal{A}$ is a pair $\langle [\![-]\!]_{\mathcal{A}}, \mathcal{I}_{\mathcal{A}} \rangle$, where:

(i) $[\![-]\!]_{\mathcal{A}}$ is a function assigning to a sort $S \in \mathcal{T}$, a set $[\![S]\!]_{\mathcal{A}}$;

(ii) $\mathcal{I}_{\mathcal{A}}$ is a function assigning to a constant $c \in \Sigma$ of type τ, an element $\mathcal{I}_{\mathcal{A}}(c^\tau) \in [\![\tau]\!]_{\mathcal{A}}$ (here $[\![-]\!]_{\mathcal{A}}$ has been extended from sorts to types as explained above).

In every structure $\mathcal{A}$, we finally require also that $[\![\Omega]\!]_{\mathcal{A}} = \{0, 1\}$, that $\mathcal{I}_{\mathcal{A}}(\bot) = 0$, that $\mathcal{I}_{\mathcal{A}}(\top) = 1$, that $\mathcal{I}_{\mathcal{A}}(=_\tau)$ is the characteristic function of the identity relation on $[\![\tau]\!]_{\mathcal{A}}$, and that $\mathcal{I}_{\mathcal{A}}(\neg), \mathcal{I}_{\mathcal{A}}(\vee), \mathcal{I}_{\mathcal{A}}(\wedge)$ are the usual truth tables functions (notice that, in these and similar passages, we implicitly use the isomorphisms $(X^Y)^Z \simeq X^{Y \times Z}$ in order to treat in the natural way curryfied binary function symbols).

Given a $\mathcal{L}$-structure $\mathcal{A} = \langle [\![-]\!]_{\mathcal{A}}, \mathcal{I}_{\mathcal{A}} \rangle$, let $\mathcal{L}^{\mathcal{A}}$ be the signature enriched by a constant $\bar{a}$ of type τ for every $a \in [\![\tau]\!]_{\mathcal{A}}$; $\mathcal{A}$ can be canonically considered as a $\mathcal{L}^{\mathcal{A}}$-structure once $\mathcal{I}_{\mathcal{A}}$ is extended to the new constants by stipulating that $\mathcal{I}_{\mathcal{A}}(\bar{a}) := a$. By induction, it is now possible to extend $\mathcal{I}_{\mathcal{A}}$ to all closed $\mathcal{L}^{\mathcal{A}}$-terms t as follows:

– $\mathcal{I}_{\mathcal{A}}(val(t, u)) = \mathcal{I}_{\mathcal{A}}(t)(\mathcal{I}_{\mathcal{A}}(u))$ (this is set-theoretic functional application);
– $\mathcal{I}_{\mathcal{A}}(\lambda x^\tau\, t)$ is the function that maps each element $a \in [\![\tau]\!]_{\mathcal{A}}$ into $\mathcal{I}_{\mathcal{A}}(t[\bar{a}])$.

From now on, we shall not distinguish for simplicity between a and its name $\bar{a}$. A $\mathcal{L}^{\mathcal{A}}$-sentence φ is *true* in $\mathcal{A}$ (in symbols $\mathcal{A} \models \varphi$) iff $\mathcal{I}_{\mathcal{A}}(\varphi) = 1$.

To introduce the notion of satisfiability we use finite assignments. Let $\mathcal{A} = \langle [\![-]\!]_{\mathcal{A}}, \mathcal{I}_{\mathcal{A}} \rangle$ be a $\mathcal{L}$-structure and let $\underline{x}$ be a finite set of variables; an *$\underline{x}$-assignment* (or simply an assignment if $\underline{x}$ is clear from the context) α is a map associating with every variable $x^\tau \in \underline{x}$ an element $\alpha(x) \in [\![\tau]\!]_{\mathcal{A}}$. An $\mathcal{L}$-formula φ is *satisfied* in $\mathcal{A}$ under the $\underline{x}$-assignment α (here $\underline{x} \supseteq fvar(\varphi)$) iff $\mathcal{I}^\alpha_{\mathcal{A}}(\varphi) = 1$, where $\mathcal{I}^\alpha_{\mathcal{A}}(\varphi)$ is the $\mathcal{L}^{\mathcal{A}}$-sentence obtained by replacing in φ the variables $x \in \underline{x}$ by (the names of) $\alpha(x)$. We usually write $\mathcal{A} \models_\alpha \varphi$ for $\mathcal{I}^\alpha_{\mathcal{A}}(\varphi) = 1$.

A formula is *satisfiable* iff it is satisfied under some assignment and a set of formulae Θ (containing altogether only finitely many variables) is satisfiable iff for some assignment α we have that $\mathcal{A} \models_\alpha \varphi$ holds for each $\varphi \in \Theta$ (of course, for this to make sense, α must be an $\underline{x}$-assignment for some $\underline{x} \supseteq fvar(\Theta)$ - and one can even assume $\underline{x} = fvar(\Theta)$ without loss of generality).

3. FRAGMENTS

General type theory is very hard to attack from a computational point of view, this is why we are basically interested only in more tractable fragments and in combinations of them. Fragments are defined as follows:

Definition 3.1. A *fragment* is a pair $\langle \mathcal{L}, T \rangle$ where $\mathcal{L} = \langle \mathcal{T}, \Sigma, a \rangle$ is a signature and T is a recursive set of $\mathcal{L}$-terms. ⊣

3.1 Algebraic Fragments

We want to use fragments as ingredients of larger and larger combined fragments: a crucial notion in this sense is that of an algebraic fragment.

Definition 3.2. A fragment $\langle \mathcal{L}, T \rangle$ is said to be an *algebraic fragment* iff T satisfies the following conditions:

(i) T *is closed under composition*, i.e. if $u[x_1, \dots, x_n] \in T$, then $u\,\sigma \in T$, where $\sigma : \{x_1 \mapsto t_1, \dots, x_n \mapsto t_n\}$ is a substitution such that $t_i \in T$ for all $i = 1, \dots, n$;

(ii) T *contains domain variables*, i.e. if τ is a type such that some variable of type τ occurs free in a term $t \in T$, then every variable of type τ belongs to T;

(iii) T *contains codomain variables*, i.e. if $t : \tau$ belongs to T, then every variable of type τ belongs to T. $\dashv$

Observe that from the above definition it follows that T is closed under renamings, i.e. that if $t \in T$ and σ a renaming, then $t\sigma \in T$. The role of Definition 3.2(i) is that of making fragment combinations non trivial, whereas the other conditions of Definition 3.2 will be needed in order to apply preprocessing purification steps to combined constraints.[5]

Quite often, one is interested in interpreting the terms of a fragment not in the class of all possible structures for the signature of the fragment, but just in some selected ones (e.g., when checking satisfiability of some temporal formulae, one might be interested only in checking satisfiability in particular flows of time, those which are for instance discrete or continuous). This is the reason for 'interpreting' fragments:

Definition 3.3. An *interpreted algebraic fragment* (to be shortened as i.a.f.) is a triple $\Phi = \langle \mathcal{L}, T, \mathcal{S} \rangle$, where $\langle \mathcal{L}, T \rangle$ is an algebraic fragment and $\mathcal{S}$ is a class of $\mathcal{L}$-structures closed under isomorphisms. $\dashv$

The notion of isomorphism we used in Definition 3.3 is the expected one: two $\mathcal{L}$-structures $\mathcal{A}_1 = \langle [\![-]\!]_{\mathcal{A}_1}, \mathcal{I}_{\mathcal{A}_1} \rangle$ and $\mathcal{A}_2 = \langle [\![-]\!]_{\mathcal{A}_2}, \mathcal{I}_{\mathcal{A}_2} \rangle$ are said to be *isomorphic* iff there are bijections $\iota_\tau : [\![\tau]\!]_{\mathcal{A}_1} \longrightarrow [\![\tau]\!]_{\mathcal{A}_2}$ (varying $\tau \in \mathcal{T}$) such that $\iota_\tau(\mathcal{I}_{\mathcal{A}_1}(c)) = \mathcal{I}_{\mathcal{A}_2}(c)$ holds for all $c : \tau \in \Sigma$ and such that $\iota_{\tau \to \upsilon}(h) = \iota_\upsilon \circ h \circ \iota_\tau^{-1}$ holds for all $h \in [\![\tau \to \upsilon]\!]_{\mathcal{A}_1}$ (notice that to give such an isomorphism it is sufficient to specify the bijections ι_S for all sorts S of $\mathcal{L}$).

The set of terms T in an i.a.f. $\Phi = \langle \mathcal{L}, T, \mathcal{S} \rangle$ is called the set of Φ-*terms* and the set of types τ such that $t : \tau$ is a Φ-term for some t is called the set of Φ-*types*. A Φ-*variable* is a variable $x : \tau$ such that τ is a Φ-type (or equivalently, a variable which is a Φ-term). It is also useful to *identify a (non-interpreted) algebraic fragment* $\langle \mathcal{L}, T \rangle$ with the interpreted algebraic fragment $\Phi = \langle \mathcal{L}, T, \mathcal{S} \rangle$, where $\mathcal{S}$ is taken to be the class of all $\mathcal{L}$-structures.

---

[5] We would like to draw the reader's attention to the fact that in Definition 3.2, when formulating the closure under composition requirement for the set of the terms T of an algebraic fragment, we asked that if $t[x_1, \dots, x_n] \in T$ and $u_1, \dots, u_n \in T$, then *precisely* the term $t[u_1, \dots, u_n]$ belongs to T (and not just some other term which is $\beta\eta$-equivalent to it, like for instance its $\beta\eta$-normal form). This strict requirement guarantees terms belonging to a combined fragment to be effectively decomposable into iterated compositions of pure terms (see Subsection 4.1).

Definition 3.4. Given an i.a.f. Φ, a Φ-*atom* is an equation $t_1 = t_2$ between Φ-terms t_1, t_2 of the same type; a Φ-*literal* is a Φ-atom or a negation of a Φ-atom, a Φ-*constraint* is a finite conjunction of Φ-literals, a Φ-*clause* is a finite disjunction of Φ-literals. Infinite sets of Φ-literals (representing an infinite conjunction) are called *generalized* Φ-*constraints* (provided they contain altogether only finitely many free variables). $\dashv$

Some Conventions. Without loss of generality, we may assume that $\top$ *is a* Φ-*atom* in every i.a.f. Φ (in fact, to be of any interest, a fragment should at least contain one term t and we can let $\top$ be $t = t$). As a consequence, $\bot$ will always be a Φ-literal; by convention, however, we shall *include* $\bot$ *among* Φ-*atoms* (hence a Φ-atom is either an equation among Φ-terms - $\top$ included - or it is $\bot$). Since we have $\bot$ as an atom, there is no need to consider the empty clause as a clause, so clauses will be disjunctions of *at least one* literal. The reader should keep in mind these slightly non standard conventions for the whole paper.

A Φ-clause is said *positive* if only Φ-atoms occur in. A Φ-atom $t_1 = t_2$ is closed if and only if t_i is closed ($i \in \{1, 2\}$); the definition of closed Φ-literals, -constraints and -clauses is analogous. For a finite set $\underline{x}$ of variables and an i.a.f. Φ, a $\Phi(\underline{x})$-atom (-term, -literal, -clause, -constraint) is a Φ-atom (-term, -literal, -clause, -constraint) A such that $fvar(A) \subseteq \underline{x}$.

We deal in this paper mainly with the *constraint satisfiability problem* for an interpreted algebraic fragment $\Phi = \langle \mathcal{L}, T, \mathcal{S} \rangle$: this is *the problem of deciding whether a* Φ-*constraint is satisfiable in some structure* $\mathcal{A} \in \mathcal{S}$. On the other hand, the *word problem* for Φ is the problem of deciding if the universal closure of a given Φ-atom is true in every structure $\mathcal{A} \in \mathcal{S}$.

Remark. More generally, one may consider the problem of deciding satisfiability of *Boolean combinations* of Φ-atoms: for this problem, one should develop 'DPLL modulo Φ' techniques analogous to recent 'DPLL modulo theory' techniques [Tinelli 2002; Nieuwenhuis et al. 2005] (see also [Bozzano et al. 2005], among others, for efficient implementations). If we take this direction, it might be useful to allow the possibility of restricting Φ-atoms occurring in Φ-constraints to a subset S_Φ of the whole set of the equations among Φ-terms: the only condition to be imposed is that any Φ-atom is equivalent in all Φ-structures to a Boolean combination of an atom from S_Φ.

3.2 Examples

We give here a list of examples of i.a.f.'s; we shall mainly concentrate on those examples which play a central role in the positive results of the paper. In all cases, the easy proof that the properties of Definition 3.2 are satisfied is just sketched or entirely left to the reader.

Example 3.5 (Simply Typed λ-Calculus). This is the i.a.f. Φ that one gets by keeping only the terms that can be built by 'omitting any reference to the type Ω'. According to Friedman theorem [Friedman 1975], this i.a.f. has decidable word problem because $\beta\eta$-normalization can decide equality of Φ-terms in all interpretations (remember that, when no semantic class $\mathcal{S}$ is mentioned in the definition of an i.a.f., it is intended that $\mathcal{S}$ consists of all possible interpretations for the signature). However, constraint satisfiability problem in this fragment is not decidable.

Example 3.6 (First-order equational fragments). Let us consider a first-order signature $\mathcal{L} = \langle \mathcal{T}, \Sigma, a\rangle$ (for simplicity, we also assume that $\mathcal{L}$ is one-sorted). Let T be the set of the first-order $\mathcal{L}$-terms and let $\mathcal{S}$ consists of the $\mathcal{L}$-structures which happen to be models of a certain first-order theory in the signature $\mathcal{L}$. Obviously, the triple $\Phi = \langle \mathcal{L}, T, \mathcal{S}\rangle$ is an i.a.f.. The Φ-atoms will be equalities between Φ-terms, i.e. first-order atomic formulae of the kind $t_1 = t_2$. Word problem in Φ=$\langle \mathcal{L}, T, \mathcal{S}\rangle$ is standard uniform word problem (as defined for the case of equational theories for instance in [Baader and Nipkow 1998]), whereas constraint satisfiability problem is the problem of deciding satisfiability of a finite set of equations and disequations.

Example 3.7 (Universal first-order fragments). The previous example disregards the relational symbols of the first-order signature $\mathcal{L}$. To take also them into consideration, it is sufficient to make some slight adjustments: besides first-order terms, also atomic formulae ($\top, \bot$ included), as well as propositional variables - namely variables having type Ω - will be terms of the fragment (notice that propositional variables are added to the set of terms in order for closure under codomain variables to be satisfied, see Definition 3.2). The semantic class $\mathcal{S}$ where the fragment is to be interpreted is again the class of the models of some first-order theory. Then, for Φ=$\langle \mathcal{L}, T, \mathcal{S}\rangle$ so defined, the constraint satisfiability problem is *essentially* the problem of deciding the satisfiability of an arbitrary finite set of $\mathcal{L}$-literals in the models belonging to $\mathcal{S}$ (the complementary problem is equivalent to the problem of deciding validity of a universal first-order formula in $\mathcal{S}$). We said 'essentially' because we need little minor observations here: first, notice that $\mathcal{L}$-atomic formulae A (resp. negated $\mathcal{L}$-atomic formulae $\neg A$) can be identified to the Φ-atoms $A = \top$ (resp. $A = \bot$). Second, in a constraint also equations $A = B$ and disequations $A \neq B$ (among $\mathcal{L}$-atomic formulae and/or propositional variables) are included. However, for instance, $A = B$ is satisfiable iff $A \wedge B$ is satisfiable or $\neg A \wedge \neg B$ is satisfiable: this means that, by case splitting, we can anyway reduce satisfiability of Φ-constraints to satisfiability of conjunctions of $\mathcal{L}$-atomic and negated $\mathcal{L}$-atomic formulae (this is a case in which one may apply the approach suggested in the last Remark of the previous subsection).

We now define different kinds of i.a.f.'s starting from the set F of first-order formulae of a first-order signature $\mathcal{L}$; for simplicity, let us suppose also that $\mathcal{L}$ is relational and one-sorted (call W the unique sort of $\mathcal{L}$).

Example 3.8 (Full First-Order Language, plain version). We take T to be the union of F with the sets of the individual variables and of the propositional variables. Of course, Φ=$\langle \mathcal{L}, T\rangle$ so defined is an algebraic fragment, whose types are W and Ω. By Church theorem, both word and constraint satisfiability problem are undecidable here (the two problems reduce to satisfiability of a first-order formula with equality); they may be decidable in case the fragment is interpreted into some specific semantic class $\mathcal{S}$. If $\mathcal{S}$ is an elementary class (i.e. it is the class of the models of a first-order theory), then the i.a.f. $\Phi = \langle \mathcal{L}, T, \mathcal{S}\rangle$ is called a *first-order* fragment.

In the next example, we build formulae (out of the symbols of our fixed first-order relational one-sorted signature $\mathcal{L}$) by using at most N (free or bound) individual variables; we are allowed to use also second order variables of arity at most K:

Example 3.9 (Full First-Order Language, NK-version). Fix cardinals $K \leq N \leq \omega$ and consider, instead of F, the set F_{NK} of formulae φ that contains at most N (free or bound) individual variables and that are built up by applying boolean connectives and individual quantifiers to atomic formulae of the following two kinds:

– $P(x_{i_1}, \dots, x_{i_n})$, where P is a relational constant and $x_{i_1}, \dots, x_{i_n}$ are individual variables (since at most $x_1, \dots, x_N$ can be used, we require that $i_1, \dots, i_n \leq N$);
– $X(x_{i_1}, \dots, x_{i_n})$, where $i_1, \dots, i_n \leq N$, and X is a variable of type $W^n \to \Omega$ with $n \leq K$ (here W^n abbreviates $W \cdots W$, n-times).

The terms in the algebraic fragment $\Phi^{\mathcal{L}}_{NK} = \langle \mathcal{L}, T^{\mathcal{L}}_{NK}\rangle$ are now the terms t such that $t \sim_{\beta\eta} \{x_1, \dots, x_n \mid \varphi\}$, for some $n \leq K$ and for some $\varphi \in F_{NK}$, with $fvar_W(\varphi) \subseteq \{x_1, \dots, x_n\}$.[6]

[6] We need to use $\beta\eta$-equivalence here to show that the properties of Definition 3.2 (namely closure

Types in such $\Phi^{\mathcal{L}}_{NK}$ are now $W^n \to \Omega$ ($n \leq K$) and this fact makes a big difference with the previous example (the difference will become apparent later when combined fragments enter into the picture). Constraint satisfiability problems still reduce to satisfiability problems for sentences: in fact, once second order variables are replaced by the names of the subsets assigned to them by some assignment α in a $\mathcal{L}$-structure, $\Phi^{\mathcal{L}}_{NK}$-atoms like $\{\underline{x} \mid \varphi\} = \{\underline{x} \mid \psi\}$ are equivalent to first-order sentences $\forall \underline{x}(\varphi \leftrightarrow \psi)$ and conversely any first-order sentence θ (with at most N bound individual variables) is equivalent to the $\Phi^{\mathcal{L}}_{NK}$-atom $\theta = \top$.

The cases $N = 1, 2$ are particularly important, because in these cases the satisfiability problem for sentences (and hence also constraint satisfiability problems in our fragments) is decidable, see [Börger et al. 2001] and the references quoted therein.

We mention that the previous two examples admit very important weaker versions in which some of the first-order operators are omitted. For instance, if universal quantifiers and negations are omitted, constraint satisfiability in the $\omega\omega$-version becomes the problem of deciding whether a geometric sequent is entailed by a finitely axiomatized geometric theory (for this terminology, see for example [Baader and Ghilardi 2005b]). Further examples can be obtained by using the large information contained in the textbook [Börger et al. 2001] (see also [Fermüller et al. 1993]). We introduce now fragments that arise from research in knowledge representation area, especially in connection to modal and description logics.

Example 3.10 (Modal/Description Logic Fragments, global version). A *modal signature* is a set O_M, whose elements are called unary 'Diamond' modal operators (the case of n-ary modal operators does not create special difficulties and it is left to the reader). O_M-concepts are built up from a countable set of atomic concepts $\mathrm{x}, \mathrm{y}, \mathrm{z}, \ldots$ by applying $\top, \bot, \neg, \sqcap, \sqcup$ as well as the diamond operators $\Diamond_k \in O_M$ (we prefer to use here modal notation instead of the equivalent description logic notation of Subsection 1.2).

With every modal signature O_M we associate the first-order signature $\mathcal{L}_M$, containing a unique sort W and, for every $\Diamond_k \in O_M$, a relational constant R_k of type $WW \to \Omega$. Suppose we are given a bijective correspondence $\mathrm{x} \longmapsto X$ between atomic concepts and second order variables of type $W \to \Omega$. Given an O_M-modal concept C and a variable w of type W, the standard translation $ST(C, w)$ is the $\mathcal{L}_M$-term of type Ω inductively defined as follows:

$$ST(\top, w) = \top; \qquad ST(\bot, w) = \bot;$$
$$ST(\mathrm{x}, w) = X(w); \qquad ST(\neg D, w) = \neg ST(D, w);$$
$$ST(D_1 \sqcap D_2, w) = ST(D_1, w) \wedge ST(D_2, w), \qquad ST(D_1 \sqcup D_2, w) = ST(D_1, w) \vee ST(D_2, w);$$
$$ST(\Diamond_k D, w) = \exists v(R_k(w, v) \wedge ST(D, v)),$$

where v is a variable of type W (different from w). Let T_M be the set of those $\mathcal{L}_M$-terms t for which there exists a concept C s.t. $t \sim_{\beta\eta} \{w \mid ST(C, w)\}$. The pair $\langle \mathcal{L}_M, T_M \rangle$ is an algebraic fragment and it becomes an i.a.f. $\Phi_M = \langle \mathcal{L}_M, T_M, \mathcal{S}_M \rangle$ if we specify also a class $\mathcal{S}_M$ of $\mathcal{L}_M$-structures closed under isomorphisms (notice that $\mathcal{L}_M$-structures, usually called *Kripke frames* in modal logic, are just sets endowed with a binary relation R_k for every $\Diamond_k \in O_M$). Using Boolean connectives, Φ_M-constraints can be equivalently represented in the form

$$\{w \mid ST(D, w)\} = \{w \mid \top\} \wedge \{w \mid ST(D_1, w)\} \neq \{w \mid \bot\} \wedge \cdots \wedge \{w \mid ST(D_n, w)\} \neq \{w \mid \bot\}.$$

Thus constraint satisfiability problem becomes, in the description logics terminology, just the (simultaneous) satisfiability problem of concepts with respect to a given T-Box (see Subsection 1.2).

Example 3.11 (Modal/Description Logic Fragments, local version). If we want to capture A-Box reasoning too, we need to build a slightly different fragment. The type-theoretic

under composition and under domain/codomain variables) are satisfied, see also Footnote 5.

signature $\mathcal{L}_{ML}$ of our fragment $\langle \mathcal{L}_{ML}, T_{ML} \rangle$ is again $\mathcal{L}_M$, but T_{ML} now contains: a) the 'concept assertions', i.e. the terms which are $\beta\eta$-equivalent to terms of the kind $ST(C, w)$; b) the 'role assertions', i.e. the terms of the kind $R_k(v, w)$; c) the variables of type W, Ω and $W \to \Omega$.

The pair $\langle \mathcal{L}_{ML}, T_{ML} \rangle$ is an algebraic fragment and it becomes an interpreted algebraic fragment $\Phi_{ML} = \langle \mathcal{L}_{ML}, T_{ML}, \mathcal{S}_{ML} \rangle$ if we specify also a class $\mathcal{S}_{ML}$ of $\mathcal{L}_{ML}$-structures closed under isomorphisms. Let us now analyze constraints in this fragment: as in Example 3.7, we can eliminate (by Boolean case splitting) atoms of the kind $ST(\varphi, w) = ST(\psi, v)$, $ST(\varphi, w) = R(v_1, v_2)$, etc. (and their negations), in favor of plain concept assertions and role assertions. In addition we have: a) identities among individual names (i.e. among variables of type W); b) identities among atomic concepts (i.e. among second order variables of type $W \to \Omega$); c) propositional variables (i.e. variables of type Ω); d) negations of identities among atomic concepts; e) negations of propositional variables; f) negations of role assertions; g) negations of identities among individual names.

Now, a)-b)-c)-d)-e) can be eliminated without loss of generality: in fact, (i) all variable identities can be eliminated by replacements; (ii) negations of identities among atomic concepts can be replaced by concept assertions involving fresh variables; (iii) propositional variables and their negations do not interact with the remaining part of the constraint and can be ignored. In conclusion, Φ_{LM}-constraints are just standard A-Boxes with, in addition, negations of role assertions and of identities among individual names (notice that traditional A-Boxes automatically include all negations of identities among distinct individual variables by the so-called 'unique name assumption', see [Baader et al. 2003]). Let us call *A-Boxes* these slightly more general constraints and let us reserve the name of *positive A-Boxes* to conjunctions of concept assertions and role assertions.

Example 3.12 (Modal/Description Logic Fragments, full version). To deal with satisfiability of a whole knowledge basis, it is sufficient to join the two previous fragments. More precisely, we can build the fragments $\Phi_{MF} = \langle \mathcal{L}_{MF}, T_{MF}, \mathcal{S}_{MF} \rangle$, where $\mathcal{L}_{MF} = \mathcal{L}_M$ and $T_{MF} = T_M \cup T_{ML}$. Types in this fragment are W, Ω and $W \to \Omega$; constraints are conjunctions of a T-Box and an A-Box.

Example 3.13 (Modal/Description Logic Fragments, non-normal case). If we want to consider the case in which some of the operators in O_M are non-normal, we can use higher-order constants $f_k : (W \to \Omega) \to (W \to \Omega)$ (instead of binary relations $R_k : WW \to \Omega$) and define a different translation. Such a translation $NT(C, w)$ differs from $ST(C, w)$ for the inductive step relative to modal operators which now reads as follows:

$$NT(\Diamond_k D, w) = f_k(\{w \mid NT(D, w)\})(w).$$

Now global, local and full algebraic fragments can be defined as in the normal case. If the easy extension to n-ary non normal operators is included and if we also interpret the resulting fragments, we get precisely the abstract description systems of [Baader et al. 2002].

Example 3.14 (μ-calculus). We show how to build a truly second-order fragment out of a modal signature O_M (in the sense of Example 3.10). In the syntax of μ-calculus [Kozen 1983], we are allowed to apply the minimum fixed point operator $\mu \mathrm{x}\, D$ to a concept D provided x occurs only positively or only negatively in D. According to well-known fixed point characterization, we can extend the translation ST from Example 3.10, by using the second-order formulae

$$ST(\mu \mathrm{x}\, D, w) := \forall Y((\{w \mid ST(D, w)\}[Y \mapsto X] \subseteq Y) \to Y(w))$$

Armed by this translation, we can easily design suitable μ-fragments.

Guarded and packed guarded fragments were introduced as generalizations of modal fragments [Andréka et al. 1998; Grädel 1999; Marx 2001]: in fact, they form classes of formulae which are remarkably large but still inherit relevant syntactic and semantic features of the more restricted class of formulae which are standard translations of modal concepts. In particular, guarded and packed guarded formulae are decidable for satisfiability (with the appropriate settings, decision procedures can also be obtained by running standard superposition provers [Ganzinger and de Nivelle 1999]). For simplicity, we give here the instructions on how to build only one version of the packed guarded fragment with equality (other versions can be built by following the methods we used above for the first-order and the modal cases). We notice that packed guarded fragments without equality are also important: to built them it is sufficient to erase any reference to the equality predicate in the relevant definitions.

Example 3.15 (Packed Guarded Fragments). Let us consider a first-order one-sorted relational signature $\mathcal{L}_G$.[7] A *guard* π is a $\mathcal{L}_G$-formula like $\bigwedge_{i=1}^k \pi_i$, where:

– π_i is obtained by applying existential quantifiers to atomic formulae $P_i(x_{i1}, \dots, x_{in_i})$ where the P_i are constants of type $W^{n_i} \to \Omega$ and $x_{i1}, \dots, x_{in_i}$ are variables of type W;
– for all $x_1, x_2 \in fvar(\pi)$, there exists an $i \in \{1, \dots, k\}$ such that $\{x_1, x_2\} \subseteq fvar(\pi_i)$.

We define the *packed guarded formulae* as follows:

– if $X : W \to \Omega$ and $x : W$ are variables, $X(x)$ is a packed guarded formula;
– if $P : W^n \to \Omega$ is a constant and $y_1 : W, \dots, y_n : W$ are variables, $P(y_1, \dots, y_n)$ is a packed guarded formula;
– if φ is a packed guarded formula, $\neg\varphi$ is a packed guarded formula;
– if φ_1 and φ_2 are packed guarded formulae, $\varphi_1 \wedge \varphi_2$ and $\varphi_1 \vee \varphi_2$ are packed guarded formulae;
– if φ is a packed guarded formula and π is a guard such that $fvar_W(\varphi) \subseteq fvar(\pi)$, then $\forall \underline{y}(\pi[\underline{x}, \underline{y}] \to \varphi[\underline{x}, \underline{y}])$ and $\exists \underline{y}(\pi[\underline{x}, \underline{y}] \wedge \varphi[\underline{x}, \underline{y}])$ are packed guarded formulae.[8]

Guarded formulae are obtained by limiting the guards in the above definition to atomic formulae. Notice that we used second order variables of type $W \to \Omega$ only (and not of type $W^n \to \Omega$ for $n > 1$): the reason, besides the applications to combined decision problems we have in mind, is that we want constraint problems to be equivalent to sentences which are still packed guarded, see below. Packed guarded formulae not containing variables of type $W \to \Omega$ are called *elementary* (or first-order) packed guarded formulae.

If we let T_G be the set of $\mathcal{L}_G$-terms t such that t is $\beta\eta$-equivalent to a term of the kind $\{w \mid \varphi\}$ (where φ is a packed guarded formula such that $fvar_W(\varphi) \subseteq \{w\}$), then the pair $\langle \mathcal{L}_G, T_G \rangle$ is an algebraic fragment. The only type in this fragment is $W \to \Omega$ and constraint satisfiability problem in this fragment is equivalent to satisfiability of guarded sentences: this is because, in case φ_1, φ_2 are packed guarded formulae with $fvar_W(\varphi_i) \subseteq \{w\}$ (for $i = 1, 2$), then $\{w \mid \varphi_1\} = \{w \mid \varphi_2\}$ is equivalent to $\forall w(\varphi_1 \leftrightarrow \varphi_2)$ which is packed guarded (just use $w = w$ as a guard).

3.3 Reduced Fragments and Residues

Let Φ be an i.a.f.; we shall use Greek letters $\Gamma, \Delta, \dots$ for Φ-constraints (i.e. for finite sets of Φ-literals) and letters $\Theta, \Lambda, \dots$ for finite sets of Φ-clauses. If Θ is a finite set of Φ-clauses and $C \equiv L_1 \vee \cdots \vee L_k$ is a Φ-clause, we say that C is a Φ-*consequence of* Θ (written $\Theta \models_\Phi C$) iff $\Theta \cup \{\neg L_1, \dots, \neg L_k\}$ is not Φ-satisfiable (i.e. iff such a set is not satisfiable in any $\mathcal{A} \in \mathcal{S}$).[9]

The notion of Φ-consequence is too strong for certain applications; for instance, when we simply need to delete certain deductively useless data, a weaker notion of redundancy (based, for example, on subsumption) is preferable. We give here an abstract axiomatization for a notion of redundancy (to understand properly the

[7] There exist some variations to the packed guarded fragment presented here, for a quick comparison among them see [Hirsch and Hodkinson 2002], Chapter 19.
[8] If $\underline{y} = \{y_1, \dots, y_n\}$, then $\forall \underline{y}$ means $\forall y_1 \cdots \forall y_n$ and $\exists \underline{y}$ means $\exists y_1 \cdots \exists y_n$. Notice that second order variables do not appear in guards.
[9] Notice that variables in clauses are treated here 'rigidly' (i.e. they are not implicitly universally quantified).

definitions of this subsection, recall that we conventionally included $\top$ and $\bot$ among Φ-atoms for any i.a.f. Φ):

Definition 3.16. A *redundancy notion* for a fragment Φ is a *recursive* binary relation Red_Φ between finite sets of Φ-clauses Θ and Φ-clauses C satisfying the following properties:

(i) $Red_\Phi(\Theta, C)$ implies $\Theta \models_\Phi C$ (soundness);
(ii) $Red_\Phi(\emptyset, \top)$ and $Red_\Phi(\{\bot\}, C)$ both hold;
(iii) $Red_\Phi(\Theta, C)$ and $\Theta \subseteq \Theta'$ imply $Red_\Phi(\Theta', C)$ (monotonicity);
(iv) $Red_\Phi(\Theta, C)$ and $Red_\Phi(\Theta \cup \{C\}, D)$ imply $Red_\Phi(\Theta, D)$ (transitivity);
(v) if C is subsumed by some $C' \in \Theta$ (i.e. if every literal occurring in C' occurs also in C), then $Red_\Phi(\Theta, C)$ holds.

Whenever a redundancy notion Red_Φ is fixed, we say that C is Φ-redundant with respect to Θ when $Red_\Phi(\Theta, C)$ holds. $\dashv$

For example, the *minimum* redundancy notion is obtained by stipulating that $Red_\Phi(\Theta, C)$ holds precisely when ($\bot \in \Theta$ or $C \equiv \top$ or $C \equiv \top \vee D$ or C is subsumed by some $C' \in \Theta$). In contrast, if the constraint solving problem for Φ is decidable, there is a maximum redundancy notion, called the *full* redundancy notion, given by the Φ-consequence relation: in fact, by introducing case-splitting for disjunctions, a recursive procedure for Φ-constraint solving can be turned into a recursive procedure deciding $\Theta \models_\Phi C$.

Let $\Phi = \langle \mathcal{L}, T, \mathcal{S} \rangle$ be an i.a.f. on the signature $\mathcal{L} = \langle \mathcal{T}, \Sigma, a \rangle$ and let $\mathcal{L}_0 = \langle \mathcal{T}_0, \Sigma_0, a_0 \rangle$ be a subsignature of $\mathcal{L}$ (this means that the proper symbols of $\mathcal{L}_0$ are included in the proper symbols of $\mathcal{L}$ and that a_0 is the restriction of a). The i.a.f. *restricted to* $\mathcal{L}_0$ is the i.a.f. $\Phi_{|\mathcal{L}_0} = \langle \mathcal{L}_0, T_{|\mathcal{L}_0}, \mathcal{S}_{|\mathcal{L}_0} \rangle$ that is so defined:

– $T_{|\mathcal{L}_0}$ is the set of terms obtained by intersecting T with the set of $\mathcal{L}_0$-terms;
– $\mathcal{S}_{|\mathcal{L}_0}$ consists of the structures $\mathcal{B}$ for which there exists some $\mathcal{A} \in \mathcal{S}$ such that $\mathcal{B} \simeq \mathcal{A}_{|\mathcal{L}_0}$. Here $\mathcal{A}_{|\mathcal{L}_0}$ is the $\mathcal{L}_0$-*reduct* of $\mathcal{A}$, i.e. it is the structure obtained from $\mathcal{A}$ by forgetting the interpretation of the proper symbols of $\mathcal{L}$ which are not proper symbols of $\mathcal{L}_0$.

An i.a.f. $\Phi_0 = \langle \mathcal{L}_0, T_0, \mathcal{S}_0 \rangle$ is said to be a $\mathcal{L}_0$-*subfragment* (or simply a subfragment, leaving the subsignature $\mathcal{L}_0 \subseteq \mathcal{L}$ as understood) of $\Phi = \langle \mathcal{L}, T, \mathcal{S} \rangle$ iff $T_0 \subseteq T_{|\mathcal{L}_0}$ and $\mathcal{S}_0 \supseteq \mathcal{S}_{|\mathcal{L}_0}$. In this case, we may also say that Φ is an *expansion* (or an *extension*) of Φ_0.

Recall that a $\Phi(\underline{x})$-constraint is a constraint in which at most the variables $\underline{x}$ occur free. Given such a $\Phi(\underline{x})$-constraint Γ and a redundancy notion Red_{Φ_0} on a subfragment Φ_0 of Φ, we call Φ_0-*basis for* Γ a set Θ of $\Phi_0(\underline{x}_0)$-clauses such that (here $\underline{x}_0$ collects those variables among the $\underline{x}$ which happen to be Φ_0-variables):

(i) all clauses $D \in \Theta$ are positive and are such that $\Gamma \models_\Phi D$;
(ii) every positive $\Phi_0(\underline{x}_0)$-clause C such that $\Gamma \models_\Phi C$ is Φ_0-redundant with respect to Θ.

Since we will be interested in in exchanging information concerning consequences over shared signatures, we need a notion of a residue, like in partial theory reasoning (see, e.g., [Baumgartner et al. 1992] for comprehensive information on the subject and the relevant pointers to the literature). Again, we prefer an abstract approach and we treat residues as clauses which are recursively enumerated by a suitable device (the device may for instance be an enumerator of certain proofs of a calculus, but there is no need to think of it in this way):[10]

Definition 3.17. Suppose we are given a subfragment Φ_0 of a fragment Φ. A *positive residue* Φ*-enumerator for* Φ_0 (often shortened as a Φ*-enumerator*) is a recursive function mapping a finite set $\underline{x}$ of Φ-variables, a $\Phi(\underline{x})$-constraint Γ and a natural number i to a Φ_0-clause $Res^{x}_{\Phi}(\Gamma, i)$ (to be written simply as $Res_{\Phi}(\Gamma, i)$) in such a way that:

– $Res_{\Phi}(\Gamma, i)$ is a positive clause;
– $fvar(Res_{\Phi}(\Gamma, i)) \subseteq \underline{x}$;
– $\Gamma \models_{\Phi} Res_{\Phi}(\Gamma, i)$ (soundness).

Any Φ_0-clause of the kind $Res_{\Phi}(\Gamma, i)$ (for some $i \geq 0$) will be called a Φ_0*-residue* of Γ. $\dashv$

Having also a redundancy notion for Φ_0 at our disposal, we can axiomatize the notion of an 'optimized' (i.e. of a non-redundant) Φ-enumerator for Φ_0. The version of the Nelson-Oppen combination procedure we give in Subsection 4.2 has non-redundant enumerators as main ingredients and it is designed to be automatically terminating in the relevant cases when termination follows from our results. These are basically the noetherian and the locally finite cases mentioned in Subsection 3.4, where enumerators which are non redundant with respect to the full redundancy notion usually exist and enjoy the termination property below.

Definition 3.18. A Φ-enumerator Res_{Φ} for Φ_0 is said to be *non-redundant* (with respect to a redundancy notion Red_{Φ_0}) iff it satisfies also the following properties for every $\underline{x}$, for every $\Phi(\underline{x})$-constraint Γ of and for every $i \geq 0$ (we write $\Gamma_{|\Phi_0}$ for the set of literals in Γ which are Φ_0-literals):

(i) if $Res_{\Phi}(\Gamma, i)$ is Φ_0-redundant with respect to $\Gamma_{|\Phi_0} \cup \{Res_{\Phi}(\Gamma, j) \mid j < i\}$, then $Res_{\Phi}(\Gamma, i)$ is either $\bot$ or $\top$;
(ii) if $\bot$ is Φ_0-redundant with respect to $\Gamma_{|\Phi_0} \cup \{Res_{\Phi}(\Gamma, j) \mid j < i\}$, then $Res_{\Phi}(\Gamma, i)$ is equal to $\bot$;
(iii) if $Res_{\Phi}(\Gamma, i)$ is equal to $\top$, then $\Gamma_{|\Phi_0} \cup \{Res_{\Phi}(\Gamma, j) \mid j < i\}$ is a Φ_0-basis for Γ. $\dashv$

Definition 3.19. A non-redundant Φ-enumerator for Φ_0 is said to be *complete* iff for every $\underline{x}$, for every $\Phi(\underline{x})$-constraint Γ and for every positive $\Phi_0(\underline{x})$-clause C, we have that $\Gamma \models_{\Phi} C$ implies that C is Φ_0-redundant with respect to $\Gamma_{|\Phi_0} \cup \{Res_{\Phi}(\Gamma, j) \mid j \leq i\}$ for some i. A non-redundant Φ-enumerator Res_{Φ} is said to be *terminating* iff for for every $\underline{x}$ and for every for every $\Phi(\underline{x})$-constraint Γ there is an i such that $Res_{\Phi}(\Gamma, i)$ is equal to $\bot$ or to $\top$. $\dashv$

[10] In [Ghilardi et al. 2005] we introduced residues operating on finite sets of clauses, here we prefer to restrict to residues operating on finite sets of literals (i.e. on constraints), because they are sufficient for our combination purposes.

Let us make a few comments on Definition 3.18: first, only non redundant residues can be produced at each step (condition (i)), if possible. If this is not possible, this means that all the relevant information has been accumulated (a Φ_0-basis has been reached). In this case, if the inconsistency $\perp$ is discovered (in the sense that it is perceived as redundant), then the residue enumeration in practice stops, because it becomes constantly equal to $\perp$ (condition (ii)). The tautology $\top$ has the special role of marking the opposite outcome: it is the residue that is returned precisely when Γ is consistent and a Φ_0-basis has been produced, meaning that all relevant semantic consequences of Γ have been discovered (conditions (ii)-(iii)).

If the redundancy notion we use is trivial (i.e. it is the minimum one), then only very mild corrections are needed for any Φ-enumerator for Φ_0 to become non-redundant: apart from minor ad hoc modifications,[11] we only need to make it constantly equal to $\perp$, as soon as $\perp$ becomes redundant in the enumeration. This observation shows that, in practice, any Φ-enumerator for Φ_0 can be made non-redundant and can consequently be used as input of our combined decision procedure.

In certain special situations (typically exemplified in computational algebra, see Subsection 3.4 below), we have an effective procedure which is able to recognize whether a given set of positive Φ_0-clauses forms a Φ_0-basis for a constraint Γ with respect to the full redundancy notion: if these *full* Φ_0*-bases* for Γ can be effectively recognized and if also Φ_0-consequence is decidable, we can always turn a complete Φ-enumerator for Φ_0 into a complete and non-redundant one with respect to the full redundancy notion. The advantage of this optimization is that the combined decision procedure of Subsection 4.2, after getting $\top$ or $\perp$ as residues, automatically recognizes that the residue exchange is over and halts.

3.4 Noetherian, Locally Finite and Convex Fragments

The above mentioned optimization for enumerators usually apply to the cases in which the 'small' fragment Φ_0 is noetherian: this important notion is borrowed from Algebra. Noetherianity conditions known from Algebra [MacLane and Birckhoff 1988] say that there are no infinite ascending chains of congruences. In finitely presented algebras, congruences are represented as sets of equations among terms, hence noetherianity can be expressed there by saying that there are no infinite ascending chains of sets of atoms, modulo logical consequence. If we translate this into our general setting, we get the following definition.

Definition 3.20. An i.a.f. Φ_0 is called *noetherian* if and only if for every finite set of variables $\underline{x}$, every infinite ascending chain

$$\Delta_1 \subseteq \Delta_2 \subseteq \cdots \subseteq \Delta_n \subseteq \cdots$$

[11]These modifications are possible provided that there are countably many closed Φ_0-atoms equivalent to $\top$ but syntactically different from it: if there are such infinitely many closed Φ_0-atoms which are 'copies' of $\top$, then we can replace $Res_\Phi(\Gamma, i)$ by one of them in case $Res_\Phi(\Gamma, i)$ (but not $\perp$) is redundant with respect to $\Gamma_{|\Phi_0} \cup \{Res_\Phi(\Gamma, j) \mid j < i\}$. By using this trick, conditions (i) and (iii) of Definition 3.18 can be forced, if the underlying redundancy notion for Φ_0 is the *minimum* one. The hypothesis that Φ_0 is endowed with such infinitely many 'copies' of $\top$ is not really restrictive and can be always obtained by slight modifications of Φ_0.

of sets of $\Phi_0(\underline{x})$-atoms is eventually constant for Φ_0-consequence (meaning that there is an n such that for all m and $A \in \Delta_m$, we have $\Delta_n \models_{\Phi_0} A$). ⊣

The following effective local finiteness notion is often used in order to make Nelson-Oppen procedures terminating [Ghilardi 2004; Baader et al. 2006; Baader and Ghilardi 2005b]:[12]

Definition 3.21. An i.a.f. Φ_0 is said to be *effectively locally finite* iff

(i) the set of Φ_0-types is recursive and constraint satisfiability problem for Φ_0 is decidable;

(ii) for every finite set of Φ_0-variables $\underline{x}$, there are finitely many computable $\Phi_0(\underline{x})$-terms $t_1, \ldots, t_n$ such that for every further $\Phi_0(\underline{x})$-term u one of the literals $t_1 \neq u, \ldots, t_n \neq u$ is not Φ_0-satisfiable (that is, in the class of the structures in which Φ_0 is interpreted, every $\Phi_0(\underline{x})$-term is equal, as an interpreted function, to one of the t_i).

The terms $t_1, \ldots, t_n$ in (ii) are called the $\underline{x}$-representative terms of Φ_0. ⊣

Noetherianity clearly implies local finiteness and it is strictly weaker than it (as shown by Examples 3.27, 3.28, 3.29 below); nevertheless, it is sufficient for termination of our combined procedure, once it is accompanied by a suitable effectiveness condition. In order to formulate such an effectiveness condition, we need the following Proposition:

PROPOSITION 3.22. *In a noetherian fragment Φ_0 every infinite ascending chain of sets of positive $\Phi_0(\underline{x})$-clauses is eventually constant for Φ_0-consequence.*

PROOF. Suppose not; in this case, an equivalent formulation of the negation of the statement of the Proposition says that there are infinitely many positive $\Phi_0(\underline{x})$-clauses $C_1, C_2, \ldots$,such that for all i, the clause C_i is not a Φ_0-consequence of $\{C_k \mid k < i\}$.

Let us build a chain of trees $T_0 \subseteq T_1 \subseteq T_2 \subseteq \cdots$, whose nodes are labeled by $\Phi_0(\underline{x})$-atoms as follows. T_0 consists of the root only, which is labeled $\top$. Suppose T_{i-1} is already built and consider the clause $C_i \equiv B_1 \vee \cdots \vee B_m$. To build T_i, do the following for every leaf K of T_{i-1} (let the branch leading to K be labeled by $A_1, \ldots, A_k$): append new sons to K labeled $B_1, \ldots, B_m$, respectively, if C_i is not a Φ_0-consequence of $\{A_1, \ldots, A_k\}$ (if it is, do nothing for the leaf K).

Consider now the union tree $T := \bigcup T_i$: since, whenever a node labeled A_{k+1} is added, A_{k+1} is not a Φ_0-consequence of the formulae labeling the predecessor nodes, by the noetherianity of Φ_0, all branches are finite and by König lemma the whole tree is itself finite. This means that for some index j, the examination of clauses C_i (for $i > j$) did not yield any modification of the already built tree. Now, C_{i+1} is not a Φ_0-consequence of $\{C_1, \ldots, C_i\}$: this means that there is a structure $\mathcal{A}$ in the class of the structures in which Φ_0 is interpreted, in which under some assignment α, all atoms of C_{i+1} are false and the $C_1, \ldots, C_i$ are all true. By induction on $j = 0, \ldots, i$, it is easily seen that there is a branch in T_j whose labeling atoms are

[12]Notice that the definition of local finiteness below becomes slightly redundant in the first-order universal case considered in these papers.

true in $\mathcal{A}$: this contradicts the fact that the tree T_i has not been modified in step $i+1$. $\square$

Suppose that Φ_0 is noetherian, that Φ_0-constraint satisfiability is decidable, and that Φ is an expansion of Φ_0: by the above Proposition, it is immediate to see that every $\Phi(\underline{x})$-constraint Γ has a finite full Φ_0-basis (i.e. there is a finite Φ_0-basis for Γ with respect to the full redundancy notion). The following noetherianity requirement for a enumerator is intended to be nothing but an effectiveness requirement for the computation of finite full Φ_0-bases.

Definition 3.23. A Φ-enumerator Res_Φ for a noetherian subfragment Φ_0 is said to be *noetherian* iff it is non redundant with respect to the full redundancy notion for Φ_0. ⊣

An immediate consequence of Proposition 3.22 is that:

PROPOSITION 3.24. *A noetherian Φ-enumerator Res_Φ for Φ_0 is terminating and also complete.*

The following proposition is also easy, but let us fix it for future reference:

PROPOSITION 3.25. *If Φ_0 is effectively locally finite and Φ is any extension of it having decidable constraint satisfiability problem, then there always exists a noetherian Φ-enumerator for Φ_0.*

PROOF. Once a $\Phi(\underline{x})$-constraint Γ is given, first check Γ for consistency: if it is inconsistent, the residue enumeration just returns $\bot$. If it is consistent, test the finitely many $\Phi_0(\underline{x}_0)$-positive clauses built up from the $\underline{x}_0$-representative terms of Φ_0 for being a Φ-consequence of Γ (here $\underline{x}_0$ are those variables among the $\underline{x}$'s which are Φ_0-variables). To build the desired Φ-enumerator, it is then sufficient to list (up to Φ_0-redundancy, which can be effectively checked) the clauses whose test is positive and to give $\top$ as a final output. $\square$

We shall see that, when dealing with noetherian enumerators over a noetherian shared fragment, the combination procedure of Subsection 4.2 becomes automatically *terminating.* If noetherianity is the essential requirement for the termination of Nelson-Oppen combination procedure, convexity is the crucial property for efficiency, as it makes the combination procedure deterministic [Oppen 1980]. The following Definition is an adaptation to our context of an analogous notion introduced in [Tinelli 2003]:

Definition 3.26. Let Φ_0 be a subfragment of Φ; we say that Φ is Φ_0-*convex* iff every finite set Γ of Φ-literals having as a Φ-consequence the disjunction of $n > 1$ Φ_0-atoms, actually has as a Φ-consequence one of them. When we say that a fragment Φ is *convex* tout court, we mean that it is Φ-convex. ⊣

As an example, notice that fragments $\Phi = \langle \mathcal{L}, T, \mathcal{S} \rangle$ analyzed in Example 3.6 are convex in case $\mathcal{S}$ is the class of the models of a first-order Horn theory.

A Φ-enumerator Res_Φ for Φ_0 is Φ_0-*convex* iff $Res_\Phi(\Gamma, i)$ is always an atom (recall that by our conventions, this includes the case in which it is $\top$ or $\bot$). Any complete non-redundant Φ-enumerator for Φ_0 can be turned into a Φ_0-convex complete non-redundant Φ-enumerator for Φ_0, in case Φ is Φ_0-convex. Thus the combination procedure of Subsection 4.2 is designed in such a way that it becomes automatically *deterministic* if the component fragments are both convex with respect to the shared fragment.

EXAMPLE 3.27 (K-ALGEBRAS). Given a field K, let us consider the one-sorted language $\mathcal{L}_{Kalg}$, whose signature contains the constants $0, 1$ of type V (V is the unique sort of $\mathcal{L}_{Kalg}$), the two binary function symbols $+$, $\circ$ of type $VV \to V$, the unary function symbol $-$ of type $V \to V$ and a K-indexed family of unary function symbols g_k of type $V \to V$. We consider the i.a.f. $\Phi_{Kalg} = \langle \mathcal{L}_{Kalg}, T_{Kalg}, \mathcal{S}_{Kalg} \rangle$ where T_{Kalg} is the set of first-order terms in the above signature (we shall use infix notation for $+$ and write kv, v_1v_2 for $g_k(v)$, $\circ(v_1, v_2)$, respectively). Furthermore, the class $\mathcal{S}_{Kalg}$ consists of the structures which happen to be models for the theory of (commutative, for simplicity) K-algebras [MacLane and Birckhoff 1988]: these are structures having both a commutative ring with unit and a K-vector space structure (the two structures are related by the equations $k(v_1v_2) = (kv_1)v_2 = v_1(kv_2)$). It is clear that Φ_{Kalg} is an interpreted algebraic fragment, which is also convex and noetherian. Constraint satisfiability problem in this fragment is equivalent to the ideal membership problem and hence it is solved by Buchberger algorithm computing Gröbner bases (see, e.g., [Cox et al. 1997] for a textbook on this topic).

EXAMPLE 3.28 (K-VECTOR SPACES). As a subfragment of Φ_{Kalg} we can consider the interpreted algebraic fragment corresponding to the theory of K-vector spaces (this is also convex and noetherian, although still not locally finite). In order to obtain a noetherian Φ_{Kalg}-enumerator for Φ_K, we need a condition that is satisfied by common admissible term orderings, namely that membership of a linear polynomial to a finitely generated ideal to be decided only by linear reduction rules of a Gröbner basis. If this happens, we get a noetherian Φ_{Kalg}-enumerator for Φ_K *simply by listing the linear polynomials of a Gröbner basis* (see [Ghilardi et al. 2005] or [Nicolini 2006] for more details).

EXAMPLE 3.29. For general algebraic reasons [MacLane and Birckhoff 1988], the observations of the previous example concerning noetherianity and convexity of the i.a.f. $\Phi_K = \langle \mathcal{L}_K, T_K, \mathcal{S}_K \rangle$ applies also in the analogous case of the theory of modules over a noetherian ring K. This implies that the theory of abelian groups is a noetherian fragment and, since *integer* or *rational linear arithmetic* (namely the theory of the integers or of the rationals under addition) is an extension of the latter, it is noetherian too (however noetherianity is lost if we add the ordering to the language).

EXAMPLE 3.30 (K-VECTOR SPACES ENDOWED WITH AN ENDOMORPHISM). This is an expansion of the fragment in Example 3.28. We augment the signature $\mathcal{L}_K$ with a unary function symbol f and, in order to interpret the fragment, we take K-vector spaces endowed with a linear endomorphism (call this fragment $\Phi_{Kend} = \langle \mathcal{L}_{Kend}, T_{Kend}, \mathcal{S}_{Kend} \rangle$ and the structures in $\mathcal{S}_{Kend}$ f-K-vector spaces). A decision procedure for constraint satisfiability in this fragment can be obtained by modifying Buchberger completion algorithm; such completion algorithm is easily seen to provide also a *noetherian* Φ_{Kend}-enumerator for the subfragment Φ_K (see again [Ghilardi et al. 2005] or [Nicolini 2006] for more details).

4. COMBINED FRAGMENTS

We give now the formal definition for the operation of combining fragments.

Definition 4.1. Let $\Phi_1 = \langle \mathcal{L}_1, T_1, \mathcal{S}_1 \rangle$ and $\Phi_2 = \langle \mathcal{L}_2, T_2, \mathcal{S}_2 \rangle$ be i.a.f.'s on the signatures $\mathcal{L}_1$ and $\mathcal{L}_2$ respectively; we define the *shared fragment of* Φ_1, Φ_2 as the i.a.f. $\Phi_0 = \langle \mathcal{L}_0, T_0, \mathcal{S}_0 \rangle$, where:

– $\mathcal{L}_0 := \mathcal{L}_1 \cap \mathcal{L}_2$;

– $T_0 := T_{1|\mathcal{L}_0} \cap T_{2|\mathcal{L}_0}$;

– $\mathcal{S}_0 := \mathcal{S}_{1|\mathcal{L}_0} \cup \mathcal{S}_{2|\mathcal{L}_0}$. ⊣

Thus the Φ_0-terms are the $\mathcal{L}_0$-terms that are both Φ_1-terms and Φ_2-terms, whereas the Φ_0-structures are the $\mathcal{L}_0$-structures which are reducts either of a Φ_1- or of a Φ_2-structure. According to the above definition, Φ_0 is a subfragment of both Φ_1 and Φ_2.

Definition 4.2. The *combined fragment* of the i.a.f.'s Φ_1 and Φ_2 is the i.a.f.

$$\Phi_1 \oplus \Phi_2 = \langle \mathcal{L}_1 \oplus \mathcal{L}_2, T_1 \oplus T_2, \mathcal{S}_1 \oplus \mathcal{S}_2 \rangle$$

so defined:

(i) $\mathcal{L}_1 \oplus \mathcal{L}_2$ is the union of the signatures $\mathcal{L}_1$ and $\mathcal{L}_2$ (i.e. it is the signature obtained by joining the proper symbols of $\mathcal{L}_1$ and $\mathcal{L}_2$);

(ii) $T_1 \oplus T_2$ is the smallest set of $\mathcal{L}_1 \oplus \mathcal{L}_2$-terms which includes $T_1 \cup T_2$, is closed under composition and contains domain and codomain variables;

(iii) $\mathcal{S}_1 \oplus \mathcal{S}_2 = \{\mathcal{A} \mid \mathcal{A}$ is a $\mathcal{L}_1 \oplus \mathcal{L}_2$-structure s.t. $\mathcal{A}_{|\mathcal{L}_1} \in \mathcal{S}_1$ and $\mathcal{A}_{|\mathcal{L}_2} \in \mathcal{S}_2\}$. ⊣

$T_1 \oplus T_2$ is defined in such a way that conditions (i)-(ii)-(iii) from Definition 3.2 are matched;[13] of course, since $\Phi_1 \oplus \Phi_2$-types turn out to be just the types which are either Φ_1- or Φ_2-types, closure under domain and codomain variables comes for free, i.e. it needs not to be postulated in (ii) above.

4.1 The Purification Steps

We say that a $\Phi_1 \oplus \Phi_2$-term is *pure* iff it is a Φ_i-term ($i = 1$ or $i = 2$) and that a $\Phi_1 \oplus \Phi_2$-constraint Γ is *pure* iff it for each literal $L \in \Gamma$ there is $i = 1$ or $i = 2$ such that L is a Φ_i-literal. Constraints in combined fragments can be purified, as we shall see. Before giving the related procedure, we first have a better look to terms in a combined fragment $\Phi_1 \oplus \Phi_2 = \langle \mathcal{L}_1 \oplus \mathcal{L}_2, T_1 \oplus T_2, \mathcal{S}_1 \oplus \mathcal{S}_2 \rangle$. For a $\mathcal{L}_1 \oplus \mathcal{L}_2$-term t and for a natural number n, the relation $\delta(t, n)$ (written as $\delta(t) \leq n$) holds whenever one of the following non mutually exclusive conditions apply:

- $n \geq 0$ and t is a shared variable (i.e. a Φ_0-variable);
- $n \geq 1$ and $t \in T_1 \cup T_2$;
- $n \geq 2$ and there are $r, s > 0$, there are terms $u[x_1, \dots, x_k], t_1, \dots, t_k$ such that $n = r + s$, $\delta(u) \leq r$, $\delta(t_1) \leq s, \dots, \delta(t_k) \leq s$ and t is equal to $u[t_1, \dots, t_k]$.

Notice that if $\delta(t) \leq n$ holds and if $n \leq m$, then $\delta(t) \leq m$ holds too. The *degree* $\delta(t)$ of a $\mathcal{L}_1 \oplus \mathcal{L}_2$-term t is the minimum d such that $\delta(t) \leq d$ holds - provided such a d exists, otherwise the degree of t is said to be infinite. It turns out that terms having degree 0 are just shared variables and terms having degree 1 are pure Φ_i-terms which are not shared variables. The following Lemma is easily proved by induction:

LEMMA 4.3. *$\mathcal{L}_1 \oplus \mathcal{L}_2$-terms t satisfying $\delta(t) \leq n$ are closed under substitutions mapping variables into variables.*

LEMMA 4.4. *A term $t \in \mathcal{L}_1 \oplus \mathcal{L}_2$ belongs to $T_1 \oplus T_2$ iff it has a finite degree.*

[13] In Corollary 4.5 below, we prove that $T_1 \oplus T_2$ is recursive, given that T_1 and T_2 are recursive.

PROOF. Let us show that terms having finite degree are closed under composition: take terms $u[x_1, \dots, x_k]$ and $t_1, \dots, t_k$ (all having finite degree) and suppose that types are compatible for substitution. We must show that $u[t_1, \dots, t_k]$ has finite degree: this is immediate, as we can suppose that $\delta(u[x_1, \dots, x_k]) \leq r$, $\delta(t_1), \dots, \delta(t_k) \leq s$, for some $r, s > 0$.

Since terms having finite degree contain domain and codomain variables (essentially because T_1, T_2 contain their domain and codomain variables), we proved that terms of finite degree satisfy conditions (i)-(ii)-(iii) of Definition 3.2. Viceversa, if $\delta(t) \leq n$, then it is immediate to see by induction on n that t belongs to any set of $\mathcal{L}_1 \oplus \mathcal{L}_2$-terms containing $T_1 \cup T_2$ and satisfying such conditions. □

COROLLARY 4.5. *$T_1 \oplus T_2$ is recursive.*

PROOF. This is an effective procedure (based on Lemma 4.4) that determines whether a given term $t \in \mathcal{L}_1 \oplus \mathcal{L}_2$ belongs to the combined fragment. We associate with t the complexity measure $\rho(t)$ given by the sum of the size of t and of the number of occurrences of constants in t. We first check whether t is a pure Φ_i-term; if not, in order for t to belong to $T_1 \oplus T_2$, the degree of t (namely the *smallest* n such that $\delta(t) \leq n$ holds) must be some $n > 1$, which means that we can split it as $u[t_1, \dots, t_k]$, where $\delta(u) \leq r$, $\delta(t_1), \dots, \delta(t_k) \leq s$, $r + s = n$, and $r, s > 0$. Since $n > r$, by Lemma 4.3 it follows that at least one of the t_i is not a variable and u cannot be a variable too because $n > s$; but this means that $\rho(u) < \rho(t), \rho(t_1) < \rho(t), \dots, \rho(t_k) < \rho(t)$, i.e. that we can recursively check whether $u, t_1, \dots, t_k$ have finite degree. □

The membership problem $t \in T_1 \oplus T_2$ might be computationally hard: since we basically have to guess a subtree of the position tree of the term t, *the procedure we outlined in Corollary 4.5 is in $\mathcal{NP}$*. For instance, if we combine $T_1 = \{f_1(f_0^{2n+1}(x)) \mid n \geq 0\}$ with $T_2 = \{f_0^{2n+1}(f_2(x)) \mid n \geq 0\}$,[14] then it is evident that in order to get a good splitting of $f_1(f_0^4(f_2(x)))$ one might need to backtrack from a first inappropriate attempt like

$$f_1(f_0^2(y)) \quad \text{and} \quad y \mapsto f_0^2(f_2(x)).$$

Notice however that these complications in complexity (with respect to the plain Nelson-Oppen case) are due to our level of generality and that they disappear in customary situations where don't know non-determinism can be avoided by looking for 'alien' subterms, see [Baader and Tinelli 2002] for a thorough discussion of the problem in standard first-order cases.

Let Γ be now any $\Phi_1 \oplus \Phi_2$-constraint: we shall provide finite sets Γ_1, Γ_2 of Φ_1- and Φ_2-literals, respectively, such that Γ is $\Phi_1 \oplus \Phi_2$-satisfiable iff $\Gamma_1 \cup \Gamma_2$ is $\Phi_1 \oplus \Phi_2$-satisfiable. The purification process is obtained by iterated applications of the Purification Rule of Figure 1 (to understand the Rule, recall that, from our conventions in Section 2, the notation $A[y, \underline{x}]$ means that $fvar(A) \subseteq \{y, \underline{x}\}$ and $A[t, \underline{x}]$ means the formula obtained by applying to A the substitution $y \mapsto t$).

[14]To complete the settings for this example, we may assume that $a(f_1) = S_0 \to S_1$, $a(f_2) = S_2 \to S_0$, $a(f_0) = S_0 \to S_0$ (f_0 is the unique shared symbol). Suitable variables should also be added to T_1, T_2 to formally fulfill the conditions of Definition 3.2.

The meaning of the Purification Rule is that *we are allowed to simultaneously abstract out in a constraint one or more occurrences of a non-variable subterm t*, provided we still produce a $\Phi_1 \oplus \Phi_2$-constraint (for termination, we also take care of not introducing variable equations).

$$\frac{\Gamma', A[t, \underline{x}]}{\Gamma',\ A[y, \underline{x}],\ y = t}$$

where (we use notations like $\Gamma', A[t, \underline{x}]$ for the constraint $\Gamma' \cup \{A[t, \underline{x}]\}$)

- t is a non-variable term (let τ be its type);
- y is a variable of type τ occurring in $A[y, \underline{x}]$ but not occurring in $\Gamma', A[t, \underline{x}]$.
- the literal $A[y, \underline{x}]$ is not an equation between variables;
- $\Gamma',\ A[y, \underline{x}],\ y = t$ is a $\Phi_1 \oplus \Phi_2$-constraint (this means that it still consists of equations and disequations among $\Phi_1 \oplus \Phi_2$-terms).

Fig. 1. **The Purification Rule.**

Proposition 4.6. *An application of Purification Rule produces an equisatisfiable constraint.*

Proof. The constraint $\Gamma', A[t, \underline{x}]$ is satisfied in a $\mathcal{L}_1 \oplus \mathcal{L}_2$-structure $\mathcal{A} \in \mathcal{S}_1 \oplus \mathcal{S}_2$ under the (finite) assignment α iff the constraint $\Gamma',\ A[y, \underline{x}],\ y = t$ produced by the rule is satisfied in $\mathcal{A}$ under the assignment obtained by incrementing α with $y \mapsto \mathcal{I}^{\alpha}_{\mathcal{A}}(t)$. $\square$

The *purification process* takes as input an arbitrary $\Phi_1 \oplus \Phi_2$-constraint Γ and applies the Purification Rule as far as possible. The Purification Rule can be applied in a don't care non deterministic way; however recall that in order to apply the rule one must check in advance that the constraint produced by it still consists of $\Phi_1 \oplus \Phi_2$-literals, hence don't know non-determinism may arise inside a single application of the rule.

Proposition 4.7. *The purification process terminates and returns a set $\Gamma_1 \cup \Gamma_2$, where Γ_i is a set of Φ_i-literals ($i = 1, 2$).*

Proof. The termination property is proved as follows. First notice that, after an application of the Purification Rule, the number N of the non variable subterm positions of the current constraint cannot increase. New equations are added by the rule, but these are only equations between a variable and a non-variable term occurring in the constraint, so that the overall number of equations that can be added during the purification process does not exceed N (notice that, after the rule has produced $\Gamma',\ A[y, \underline{x}], y = t$, the new position in which the subterm t is now is not available for another purification step, since purification steps cannot produce variables equations).

Let us now show that if the Purification Rule does not apply to Γ, then Γ splits into two pure Φ_i-constraints. We first claim that, since Purification Rule does not apply to Γ, any term t in a literal $t = v$ or $t \neq v$ of Γ has degree at most 1 (i.e. it is either in T_1 or in T_2): otherwise we have $t \equiv u[t_1, \dots, t_k]$, with $u[x_1, \dots, x_k], t_1, \dots, t_k$ all having lower degree than t. Since the degree of u and of the t_i's is lower than the degree of t, both u and at least one of the t_i are not a variable (see Lemma 4.3); suppose for instance that t_1 is not a variable and that the constraint Γ is $\Gamma', u[t_1, \dots, t_k] = v$. Contrary to the assumption, the Purification Rule applies to Γ and produces the constraint

$$\Gamma',\, u[x_1, t_2, \dots, t_k] = v,\, x_1 = t_1 \tag{1}$$

(x_1 can be renamed, if needed):[15] in fact fragments are closed under domain/codomain variables, hence the variable we need is at our disposal, so that (1) is a $\Phi_1 \oplus \Phi_2$-constraint (notice that $u[x_1, t_2, \dots, t_k]$ has a degree, hence it is a $\Phi_1 \oplus \Phi_2$-term by Lemma 4.4).

Having established that terms in Γ are all pure, we wonder whether there are impure (in)equations. This is also impossible, because the Purification Rule can replace e.g., $t_1 = t_2$ by $t_1 = x \wedge x = t_2$ in case $t_1 \in T_1, t_2 \in T_2$ are non-variable terms (since fragments are closed under codomain variables, if $t_1 : \tau_1 \in T_1$, $t_2 : \tau_2 \in T_2$ and $\tau := \tau_1 = \tau_2$, then the type τ is shared and $t_i = x$ is a Φ_i-atom for every variable $x : \tau$). $\square$

Actually, one can prove that the Purification Rule (if exhaustively applied) can bring the current constraint not only into a pure form, but also in a form in which negative literals just contain variables and positive literals do not contain equations among two non-variable terms.

4.2 The Combination Procedure

In this subsection, we develop a procedure which is designed to solve constraint satisfiability problems in combined fragments: the procedure is sound and we shall investigate afterwards sufficient conditions for it to be terminating and complete. Let us fix (once and for all) relevant notation for the involved data.

Assumptions/Notational Conventions. We suppose that we are given two i.a.f.'s $\Phi_1 = \langle \mathcal{L}_1, T_1, \mathcal{S}_1 \rangle$ and $\Phi_2 = \langle \mathcal{L}_2, T_2, \mathcal{S}_2 \rangle$, with shared fragment $\Phi_0 = \langle \mathcal{L}_0, T_0, \mathcal{S}_0 \rangle$. We suppose also that a redundancy notion Red_{Φ_0} for Φ_0 and two Φ_i-enumerators for Φ_0 (call them Res_{Φ_1}, Res_{Φ_2}) are available; Res_{Φ_1} and Res_{Φ_2} are assumed to be both non-redundant with respect to Red_{Φ_0}. We also fix a purified $\Phi_1 \oplus \Phi_2$-constraint $\Gamma_1 \cup \Gamma_2$ to be tested for $\Phi_1 \oplus \Phi_2$-consistency and we indicate by $\underline{x}_i$ the free variables occurring in Γ_i ($i = 1, 2$). We let Γ_0 be the set of those literals among $\Gamma_1 \cup \Gamma_2$ which happen to be Φ_0-literals and we let $\underline{x}_0$ be a tuple containing those variables among $\underline{x}_1 \cup \underline{x}_2$ which happen to be Φ_0-variables. Notice that we can freely suppose that $\Gamma_0 = \Gamma_1 \cap \Gamma_2$ and that $\underline{x}_0 = \underline{x}_1 \cap \underline{x}_2$: otherwise, Φ_0-literals and/or trivial equations like $x = x$ can be added to Γ_1 and Γ_2 till this holds.

The Procedure TCOMB. We introduce our combined procedure in a tableau form: our tableau procedure generates a tree whose internal nodes are labeled by sets of $\Phi_0(\underline{x}_0)$-atoms; *the root of the tree is labeled by the empty set, leaves are the unique*

[15] This might be needed to fulfill the Purification Rule requirement that x_1 does not occur in Γ.

nodes whose label set contains either $\top$ *or* $\bot$. Leaves whose label set contains $\top$ are called *saturated* and leaves whose label set contains $\bot$ are called *unsatisfiable*.

We have just one Tableau Expansion Rule, which is explained in Figure 2: the rule applies to any node whose label set does not contain $\bot$ or $\top$ and attaches son nodes to it as shown in the consequent of the Tableau Expansion Rule itself.

$$\frac{\Delta}{\Delta, A_1 \parallel \cdots \parallel \Delta, A_k}$$

where the positive clause $C :\equiv A_1 \vee \cdots \vee A_k$ satisfies the following requirements:

(i) C is a Φ_i-consequence of $\Gamma_i \cup \Delta$, for $i = 1$ or $i = 2$;
(ii) if $\bot$ is Φ_0-redundant with respect to $\Gamma_0 \cup \Delta$, then C is $\bot$;
(iii) if C is Φ_0-redundant with respect to $\Gamma_0 \cup \Delta$, then C is $\top$ or $\bot$.

Fig. 2. **The Tableau Expansion Rule.**

The tableau branches which are infinite or end with a saturated leaf are called *open*, whereas the branches ending with an unsatisfiable leaf are called *closed*. The procedures stops (and the generation of the above tree is interrupted) iff all branches are closed or if there is an open finite branch. The tableau returns *"saturated"* iff it has a saturated leaf and it returns *"unsatisfiable"* iff all branches are closed (of course termination is not guaranteed in the general case).

A tableau is non deterministically generated in the sense that many instances of the Tableau Expansion Rule can be applied to a given node, in order to produce its successors: a strategy (i.e. a 'selection function', see below) for application of the Tableau Expansion Rule is needed in order to concretely implement our procedure. Not all strategies are equally productive, in order for a strategy to be really productive, it must fulfill a fairness requirement.

A tableau is *fair* iff the following happens for every open branch $\Delta_0 \subseteq \Delta_1 \subseteq \cdots$: if $C \equiv Res_{\Phi_i}(\Gamma_i \cup \Delta_k, l)$ for some $i = 1, 2$ and for some $k, l \geq 0$, then C is Φ_0-redundant with respect to $\Gamma_0 \cup \Delta_n$ for some n (roughly, *residues with respect to* Φ_i *of an open branch are redundant with respect to the atoms in the branch*).

We first show how to effectively build a fair tableau (under the current assumptions/notational conventions):

Proposition 4.8. *There exists a fair tableau.*

Proof. A *selection function* is a recursive function returning to a finite list Δ of $\Phi_0(\underline{x}_0)$-atoms a $\Phi_0(\underline{x}_0)$-positive clause $\textsc{Choose}(\Delta)$ matching the requirements (i)-(ii)-(iii) from Figure 2. A selection function is said to be *fair* iff the tableau generated according to it is fair; thus, we simply need to find a fair selection function.[16]

[16]Notice that, since a selection function takes as input a *list* (and not just a set) of atoms, it might be sensible not only to the atoms labeling the current node, but also to the order in which such atoms have been produced along the branch.

For a finite set Δ of $\Phi_0(\underline{x}_0)$-literals and for a list Θ of $\Phi_0(\underline{x}_0)$-clauses, let us define the auxiliary procedure $\textsc{LMin}(\Delta, \Theta)$. We have that

– $\textsc{LMin}(\Delta, [C]) = C$;
– $\textsc{LMin}(\Delta, [C|\Theta']) = \textsc{LMin}(\Delta, \Theta')$, if $Red_{\Phi_0}(\Delta \cup \{D\}, C)$ holds for some $D \in \Theta'$;
– $\textsc{LMin}(\Delta, [C|\Theta']) = C$, otherwise

(roughly, the procedure takes the leftmost Red_{Φ_0}-maximal element of the list Θ, using the set Δ as a parameter).

Fix now a surjective recursive function

$$\delta = (\delta_1, \delta_2, \delta_3) : \mathbb{N} \longrightarrow \{1, 2\} \times \mathbb{N} \times \mathbb{N}$$

such that $n \geq \delta_2(n)$ holds for every n (this function can be easily built by using a recursive encoding of pairs, see, for example, [Odifreddi 1989]).

For $\Delta = [A_1, \dots, A_n]$, define now $\textsc{Choose}(\Delta)$ to be

$$\textsc{LMin}(\Gamma_0 \cup \Delta, [D_1, D_2, D_3]), \tag{2}$$

where D_1 is $Res_{\Phi_{\delta_1(n)}}(\Gamma_{\delta_1(n)} \cup \{A_1, \dots, A_{\delta_2(n)}\}, \delta_3(n))$, whereas D_2, D_3 are the clauses $Res_{\Phi_1}(\Gamma_1 \cup \Delta, 0)$ and $Res_{\Phi_2}(\Gamma_2 \cup \Delta, 0)$, respectively. The intuitive explanation of this definition is as follows: for $i = 1, 2$ and $j \leq n$, the residues $Res_{\Phi_i}(\Gamma_i \cup \{A_1, \dots, A_j\}, k)$ can be disposed into two matrices having n infinite rows ($Res_{\Phi_i}(\Gamma_i \cup \{A_1, \dots, A_j\}, k)$ is the k-th entry of the j-th row in the i-th matrix). Now our selection function explores the rows of these two matrices by a diagonal path, but before making the final choice it checks whether in the first entries of the two last rows there is anything more informative from the redundancy viewpoint.

Clearly $\textsc{Choose}(\Delta)$ is a Φ_i-consequence of $\Gamma_i \cup \Delta$, for $i = 1$ or $i = 2$, because of the soundness condition of Definition 3.17; moreover if $\textsc{Choose}(\Delta)$ is redundant with respect to $\Gamma_0 \cup \Delta$, then (by Definition 3.16 (iii)) it must be equal to $Res_{\Phi_2}(\Gamma_2 \cup \Delta, 0)$, hence it is $\top$ or $\bot$, according to Definition 3.18 (i) (and it is $\bot$, if $\bot$ is redundant with respect to $\Gamma_0 \cup \Delta$, by Definition 3.18 (ii)). Thus conditions (i)-(iii) from Figure 2 are satisfied.

To show fairness of the tableau, pick an open branch labeled by the increasing sets of Φ_0-atoms $\Delta_0 \subseteq \Delta_1 \subseteq \cdots$ and suppose that $C \equiv Res_{\Phi_i}(\Gamma_i \cup \Delta_k, l)$ for some $i = 1, 2$ and for some $k, l \geq 0$. Let us distinguish the case in which the open branch is finite and the case in which it is infinite.

If it is finite, it ends with a saturated label, which means that for some n, we have that $\top$ is a disjunct of $\textsc{Choose}(\Delta_n)$. From (2) and Definition 3.16 (ii)-(iii)-(iv), we must have that $\textsc{Choose}(\Delta_n) \equiv \top$ and that both $Res_{\Phi_1}(\Gamma_1 \cup \Delta_n, 0)$ and $Res_{\Phi_2}(\Gamma_2 \cup \Delta_n, 0)$ are equal to $\top$. To show this, notice that: (a) a residue equal to $\top$ or to $\top \vee D$ selected by the function $\textsc{Choose}(\Delta_n)$ according to (2) cannot be either $Res_{\Phi_{\delta_1(n)}}(\Gamma_{\delta_1(n)} \cup \{A_1, \dots, A_{\delta_2(n)}\}, \delta_3(n))$ or $Res_{\Phi_1}(\Gamma_1 \cup \Delta_n, 0)$, because $\top$ and $\top \vee D$ are always redundant; (b) hence it must be $Res_{\Phi_2}(\Gamma_2 \cup \Delta_n, 0)$, which implies however (by the way the procedure $\textsc{LMin}$ is defined) that $Res_{\Phi_1}(\Gamma_1 \cup \Delta_n, 0)$ is redundant with respect to $\Gamma_0 \cup \Delta_n \cup \{Res_{\Phi_2}(\Gamma_2 \cup \Delta_n, 0)\}$ (which is equal to $\Gamma_0 \cup \Delta_n \cup \{\top \vee D\}$ or to $\Gamma_0 \cup \Delta_n \cup \{\top\}$) and hence with respect to $\Gamma_0 \cup \Delta_n$ by transitivity. All this implies that $Res_{\Phi_1}(\Gamma_1 \cup \Delta_n, 0)$ and $Res_{\Phi_2}(\Gamma_2 \cup \Delta_n, 0)$ are both redundant w.r.t. $\Gamma_0 \cup \Delta_n$ and consequently they are both equal to $\top$ by Definition 3.18 (i).[17]

Since $Res_{\Phi_1}(\Gamma_1 \cup \Delta_n, 0)$ and $Res_{\Phi_2}(\Gamma_2 \cup \Delta_n, 0)$ are both equal to $\top$, by Definition 3.18 (iii), we conclude that $\Gamma_0 \cup \Delta_n$ is a Φ_0-basis for both $\Gamma_1 \cup \Delta_n$ and $\Gamma_2 \cup \Delta_n$, which means that the Φ_i-residue C is redundant with respect to $\Gamma_0 \cup \Delta_n$: in fact, since $C \equiv Res_{\Phi_i}(\Gamma_i \cup \Delta_k, l)$, by Definition 3.17, C is a Φ_i-consequence of $\Gamma_i \cup \Delta_k$ (for $\Delta_k \subseteq \Delta_n$) and hence also of $\Gamma_i \cup \Delta_n$, thus the definition of a Φ_0-basis applies.

If the branch is infinite, for some n, we have $\delta_1(n) = i, \delta_2(n) = k, \delta_3(n) = l$. Hence, either C has been selected, or some better choice (from the redundancy point of view) has been made according to (2). Since this better choice D cannot be $\top$ or $\bot$ because the branch is infinite, some atom of D (or of C, if C has been directly selected) is in Δ_{n+1}: this means that C is redundant with respect to $\Gamma_0 \cup \Delta_{n+1}$ because of Definition 3.16(iii)-(iv)-(v). □

We remark that the fair selection function given in (2) above can be optimized in specific situations, where extra information on the input residue enumerators is available; however, the existence of a uniform schema for defining a fair tableau is an interesting property of our combination procedure. From now on, *when we refer to the procedure* TCOMB*, we refer to any deterministic version of it producing fair tableaux.*

4.3 Soundness and Termination

One possible exit of our procedure is when it generates a finite tree whose leaves are all unsatisfiable: this is precisely the case in which the whole procedure returns *"unsatisfiable"*.

PROPOSITION 4.9 (SOUNDNESS). *If the procedure* TCOMB *returns "unsatisfiable", then the purified constraint $\Gamma_1 \cup \Gamma_2$ is $\Phi_1 \oplus \Phi_2$-unsatisfiable.*

PROOF. All branches of the tree generated by the execution of the procedure described in Section 4.2 are closed: hence, an easy inductive argument shows that, if a node in the tree is labeled by Δ, then $\Gamma_1 \cup \Gamma_2 \cup \Delta$ is $\Phi_1 \oplus \Phi_2$-unsatisfiable. As a consequence, $\Gamma_1 \cup \Gamma_2$ itself is $\Phi_1 \oplus \Phi_2$-unsatisfiable, because the root is labeled by the empty set of atoms. □

Next, we identify a relevant termination case:

PROPOSITION 4.10 (TERMINATION). *If Φ_0 is noetherian and Red_{Φ_0} is the full redundancy notion, then the procedure* TCOMB *terminates on the purified constraint $\Gamma_1 \cup \Gamma_2$.*

PROOF. Let us consider the tree T generated by the execution of the procedure TCOMB as described in Section 4.2. Recalling that T is finite iff the procedure terminates, we now suppose that the procedure does not terminate. In this way T, which is a finitely branching tree by construction, is not finite and has an infinite

[17] They cannot be equal to $\bot$, because of redundancy with respect $\Gamma_0 \cup \Delta_n$: recall from Definition 3.18(ii) that the residue is $\bot$, if $\bot$ is redundant (and our residue cannot be $\bot$ because the branch is open).

branch by König lemma. This means that there is a infinite chain of sets of $\Phi_0(\underline{x}_0)$-atoms

$$\Delta_1 \subset \Delta_2 \subset \cdots \subset \Delta_n \subset \cdots ,$$

where Δ_i is the label of a node that belongs to that infinite path, $\Delta_{i+1} = \Delta_i \cup \{A_i\}$ and $Red_{\Phi_0}(\Gamma_0 \cup \Delta_i, A_i)$ does not hold by Definitions 3.16(iv)-(v) and by condition (iii) from Figure 2. Since Red_{Φ_0} is the full redundancy notion, we obtained an infinite sequence $A_1, A_2, A_3, \ldots$ such that $\Gamma_0 \cup \{A_j \mid j < i\} \not\models_{\Phi_0} A_i$ for every i. This contradicts noetherianity of Φ_0. □

4.4 Towards Completeness

Completeness of the procedure TCOMB cannot be achieved easily, heavy conditions are needed. In this section, we nevertheless identify what is the 'semantic meaning' of a run of the procedure that either does not terminate or terminates with a saturation message.

Since our investigations are taking a completeness-oriented route, it is quite obvious that we must consider from now on only the case in which the input Φ_i-enumerators are *complete* (see Definition 3.19). In addition we need a compactness-like assumption. We say that an i.a.f. Φ is Φ_0-*compact* (where Φ_0 is a subfragment of Φ) iff, given a Φ-constraint Γ and a generalized Φ_0-constraint Γ_0, we have that $\Gamma \cup \Gamma_0$ is Φ-satisfiable if and only if for all finite $\Delta_0 \subseteq \Gamma_0$, we have that $\Gamma \cup \Delta_0$ is Φ-satisfiable.

PROPOSITION 4.11. *Any extension Φ of a locally finite i.a.f. Φ_0 is Φ_0-compact.*

PROOF. Recall that, according to Definition 3.4, a generalized Φ_0-constraint Γ_0 is an infinite set of Φ_0-literals in which only finitely many Φ_0-variables (call them $\underline{x}$) occur free. Since Φ_0 is locally finite, there exist finitely many $\Phi_0(\underline{x})$-terms representing all $\Phi_0(\underline{x})$-terms up to Φ_0-equivalence: for this reason, a generalized $\Phi_0(\underline{x})$-constraint Γ_0 is equivalent to the constraint in which all terms have been replaced by their representatives. □

The above Proposition means that, if we assume effective local finiteness in order to guarantee termination, Φ_0-compactness is guaranteed too. Notice that only special kinds of generalized Φ-constraints are involved in the definition of Φ_0-compactness, namely those that contain finitely many proper Φ-literals; thus, Φ_0-compactness is a rather weak condition. Finally, it goes without saying that, by the compactness theorem for first-order logic, Φ_0-compactness is guaranteed whenever Φ is a first-order fragment from Examples 3.6, 3.7, 3.8.

PROPOSITION 4.12. *Suppose that Φ_1, Φ_2 are both Φ_0-compact, that the two Φ_i-enumerators Res_{Φ_i} are complete, and that the procedure* TCOMB *does not return "unsatisfiable" when applied to the purified constraint $\Gamma_1 \cup \Gamma_2$. Then there are $\mathcal{L}_i$-structures $\mathcal{M}_i \in \mathcal{S}_i$ and $\mathcal{L}_i$-assignments α_i $(i = 1, 2)$ such that:*

(i) *$\mathcal{M}_1 \models_{\alpha_1} \Gamma_1$ and $\mathcal{M}_2 \models_{\alpha_2} \Gamma_2$;*
(ii) *for every $\Phi_0(\underline{x}_0)$-atom A, we have that $\mathcal{M}_1 \models_{\alpha_1} A$ iff $\mathcal{M}_2 \models_{\alpha_2} A$.*

PROOF. A set of positive $\Phi_0(\underline{x}_0)$-clauses $\Theta_0^\star$ is *saturated* if and only if it is closed under the two rules:

$$\begin{array}{lcl}\Gamma_1 \cup \Theta_0^\star \models_{\Phi_1} C & \Rightarrow & C \in \Theta_0^\star \\ \Gamma_2 \cup \Theta_0^\star \models_{\Phi_2} C & \Rightarrow & C \in \Theta_0^\star,\end{array}$$

for every positive $\Phi_0(\underline{x}_0)$-clause C.

Let us suppose that Res_{Φ_1} and Res_{Φ_2} are complete enumerators and suppose that the fair tableau generated by TCOMB for the purified constraint $\Gamma_1 \cup \Gamma_2$ does not return *"unsatisfiable"*: by König's lemma, this tableau must contain an open (finite or infinite) branch. Let this open branch be labeled by $\Delta_0 \subseteq \Delta_1 \subseteq \cdots$ and let us take $\Delta := \bigcup_j \Delta_j$; we define $\Theta_0^\star$ to be $\{C \mid C$ is a positive $\Phi_0(\underline{x}_0)$-clause s.t. $\Gamma_1 \cup \Delta \models_{\Phi_1} C\}$ (we remark that $\Delta \subseteq \Theta_0^\star$). $\Theta_0^\star$ is saturated and (for $i = 1, 2$) $\Gamma_i \cup \Theta_0^\star$ is Φ_i-satisfiable, as shown by Lemma 4.13. Thus, Lemma 4.14 applies and there are two $\mathcal{L}_i$-structures $\mathcal{M}_i \in \mathcal{S}_i$ satisfying $\Gamma_i \cup \Theta_0^\star$ under assignments α_i, such that $\mathcal{M}_1, \alpha_1$ and $\mathcal{M}_2, \alpha_2$ satisfy the same $\Phi_0(\underline{x}_0)$-atoms. □

LEMMA 4.13. *The set $\Theta_0^\star$ defined above is saturated, $\Gamma_1 \cup \Theta_0^\star$ is Φ_1-satisfiable and $\Gamma_2 \cup \Theta_0^\star$ is Φ_2-satisfiable.*

PROOF. To prove that $\Theta_0^\star$ is saturated, we need to show that

$$\Gamma_2 \cup \Theta_0^\star \models_{\Phi_2} C \qquad \Rightarrow \qquad C \in \Theta_0^\star$$

where C is a positive $\Phi_0(\underline{x}_0)$-clause. We first prove that $\Gamma_1 \cup \Delta \models_{\Phi_1} C$ implies $\Gamma_2 \cup \Delta \models_{\Phi_2} C$ (and conversely, but the proof of the converse is the same).

$\Gamma_1 \cup \Delta \models_{\Phi_1} C$ iff there exists n such that $\Gamma_1 \cup \Delta_n \models_{\Phi_1} C$ (by Φ_0-compactness of Φ_1) iff there exists k such that[18] $Red_{\Phi_0}(\Gamma_0 \cup \Delta_n \cup \{C_0, \ldots, C_k\}, C)$ holds, where $C_j \equiv Res_{\Phi_1}(\Gamma_1 \cup \Delta_n, j)$ (by the completeness of the Φ_1-enumerator). By the fairness requirement on our tableau, there exist m_j's such that $Red_{\Phi_0}(\Gamma_0 \cup \Delta_{m_j}, C_j)$ holds ($j \in \{1, \ldots, k\}$), hence by monotonicity of redundancy there exists $m \geq n, m \geq m_j$ such that $Red_{\Phi_0}(\Gamma_0 \cup \Delta_m, C_j)$ holds for each $j \in \{1, \ldots, k\}$; by transitivity of redundancy we have $Red_{\Phi_0}(\Gamma_0 \cup \Delta_m, C)$ and consequently also $\Gamma_0 \cup \Delta_m \models_{\Phi_0} C$. Thus $\Gamma_2 \cup \Delta_m \models_{\Phi_2} C$ and finally $\Gamma_2 \cup \Delta \models_{\Phi_2} C$.

We showed that $\Gamma_1 \cup \Delta \models_{\Phi_1} C$ holds iff $\Gamma_2 \cup \Delta \models_{\Phi_2} C$ holds; it follows that:

$$\Theta_0^\star = \{C \text{ is a positive } \Phi_0(\underline{x}_0)\text{-clause} \mid \Gamma_1 \cup \Delta \models_{\Phi_1} C\} \tag{3}$$

$$\Theta_0^\star = \{C \text{ is a positive } \Phi_0(\underline{x}_0)\text{-clause} \mid \Gamma_2 \cup \Delta \models_{\Phi_2} C\}, \tag{4}$$

that is $\Gamma_2 \cup \Theta_0^\star \models_{\Phi_2} C \Rightarrow C \in \Theta_0^\star$.

We finally prove that $\Gamma_i \cup \Theta_0^\star$ is Φ_i-satisfiable for $i = 1, 2$. To this aim notice that $\Gamma_i \cup \Theta_0^\star$ is Φ_i-satisfiable iff $\Gamma_i \cup \tilde{\Delta}$ is Φ_i-satisfiable for each $\tilde{\Delta} \subseteq \Delta$, $\tilde{\Delta}$ finite (by Φ_0-compactness of Φ_i and by (3)-(4)). For each such $\tilde{\Delta}$, there exists an index n such that $\tilde{\Delta} \subseteq \Delta_n$, thus it is sufficient to prove that $\Gamma_i \cup \Delta_n$ is Φ_i-satisfiable for each n. But if $\Gamma_i \cup \Delta_n \models_{\Phi_i} \perp$ for some n, then $\perp$ appears as a residue of $\Gamma_i \cup \Delta_n$ (by completeness of Res_{Φ_i} and by Definition 3.18 (ii)) and by the fairness of our tableau, $\perp$ is then redundant with respect to some $\Gamma_0 \cup \Delta_m$: this implies that $\perp \in \Delta_{m+1}$ by condition (ii) of Figure 2, contrary to fact that the branch is not closed. □

[18]Recall that $\Gamma_0 = \Gamma_{1|\Phi_0} = \Gamma_{2|\Phi_0}$ according to the 'Notational Conventions' at the beginning of Subsection 4.2.

LEMMA 4.14. *Suppose that we are given a saturated set of positive $\Phi_0(\underline{x}_0)$-clauses $\Theta_0^\star$, such that $\Gamma_1 \cup \Theta_0^\star$ is Φ_1-satisfiable and $\Gamma_2 \cup \Theta_0^\star$ is Φ_2-satisfiable. Then there are structures $\mathcal{M}_1 \in \mathcal{S}_1$, $\mathcal{M}_2 \in \mathcal{S}_2$ and two assignments α_1, α_2 such that $\mathcal{M}_1 \models_{\alpha_1} \Gamma_1 \cup \Theta_0^\star$ and $\mathcal{M}_2 \models_{\alpha_2} \Gamma_2 \cup \Theta_0^\star$. Moreover, for every $\Phi_0(\underline{x}_0)$-atom A, $\mathcal{M}_1 \models_{\alpha_1} A$ holds if and only if $\mathcal{M}_2 \models_{\alpha_2} A$ holds.*

PROOF. A set Ξ of $\Phi_0(\underline{x}_0)$-literals is *exhaustive* iff for each $\Phi_0(\underline{x}_0)$-atom A, $A \in \Xi$ or $\neg A \in \Xi$.

Let us consider any terminating strict total order on $\Phi_0(\underline{x}_0)$-atoms (it exists by the well ordering principle) and let us extend it to a terminating strict total order on multisets of $\Phi_0(\underline{x}_0)$-atoms.[19] We use such an ordering to define increasing subsets Ξ_C, varying C among positive $\Phi_0(\underline{x}_0)$-clauses in $\Theta_0^\star$ (positive clauses are identified here with the multiset of their atoms).

We say that the $\Phi_0(\underline{x}_0)$-clause $C \equiv A \vee A_1 \vee \cdots \vee A_n$ from $\Theta_0^\star$ is *productive* (and produces the $\Phi_0(\underline{x}_0)$-atom A) iff $\{A\} > \{A_1, \ldots, A_n\}$ and $A_1, \ldots, A_n \notin \Xi^+_{<C}$ where $\Xi^+_{<C} := \bigcup_{D<C, D\in\Theta_0^\star} \Xi^+_D$. If C is productive and produces A, then we put $\Xi^+_C := \Xi^+_{<C} \cup \{A\}$, otherwise we put $\Xi^+_C := \Xi^+_{<C}$.

Let us define $\Xi^+ := \bigcup_{C\in\Theta_0^\star} \Xi^+_C$ and $\Xi := \Xi^+ \cup \{\neg A \mid A \text{ is a } \Phi_0(\underline{x}_0)\text{-atom and } A \notin \Xi^+\}$. By construction, $\Xi \models_{\Phi_0} \Theta_0^\star$ (because Ξ contains a $\Phi_0(\underline{x}_0)$-atom for every $\Phi_0(\underline{x}_0)$-clause in $\Theta_0^\star$).

We need to show that $\Gamma_1 \cup \Xi$ is Φ_1-satisfiable and $\Gamma_2 \cup \Xi$ is Φ_2-satisfiable. First of all, we *claim* that if a $\Phi_0(\underline{x}_0)$-clause $C \equiv A \vee A_1 \vee \cdots \vee A_n$ is productive and $\{A\} > \{A_1, \ldots, A_n\}$, then $A_1, \ldots, A_n \notin \Xi^+$. To show this, recall that, by definition, $A_i \in \Xi^+$ ($i \in \{1, \ldots, n\}$) iff A_i belongs to a productive $\Phi_0(\underline{x}_0)$-clause C_i and A_i is the maximum atom in it, thus $C_i < C$ (by multisets ordering): however none of the A_i can be in $\Xi^+_{<C}$, because C is productive, thus justifying our claim.

We suppose now that $\Gamma_1 \cup \Xi$ is Φ_1-unsatisfiable. By Φ_0-compactness of the i.a.f. Φ_1, there are $\Phi_0(\underline{x}_0)$-atoms $B_1, \ldots, B_m \notin \Xi^+$ and productive $\Phi_0(\underline{x}_0)$-clauses

$$\begin{array}{rcl} C_1 & \equiv & A_1 \vee A_{11} \vee \cdots \vee A_{1k_1} \\ & & \cdots \\ C_n & \equiv & A_n \vee A_{n1} \vee \cdots \vee A_{nk_n} \end{array}$$

(with maximum $\Phi_0(\underline{x}_0)$-atoms $A_1, \ldots, A_n$ respectively) s.t. the constraint $\Gamma_1 \cup \{A_1, \ldots, A_n, \neg B_1, \ldots, \neg B_m\}$ is Φ_1-unsatisfiable. It follows that

$$\Gamma_1 \cup \{C_1, \ldots, C_n\} \cup \{\neg A_{11}, \cdots, \neg A_{nk_n}, \neg B_1, \ldots, \neg B_m\}$$

is also unsatisfiable. As $C_1, \ldots, C_n$ are positive $\Phi_0(\underline{x}_0)$-clauses in $\Theta_0^\star$ and $\Theta_0^\star$ is saturated, the positive $\Phi_0(\underline{x}_0)$-clause

$$D \equiv \bigvee_{i,j} A_{ij} \vee B_1 \vee \cdots \vee B_m$$

is also in $\Theta_0^\star$.[20] By construction, some of the atoms of this positive clause belongs

[19] We are using basic information on multiset orderings that can be found in textbooks like [Baader and Nipkow 1998].

[20] Notice that we cannot have $k_1 = \cdots = k_n = m = 0$, because $\Gamma_1 \cup \{C_1, \ldots, C_n\} \subseteq \Gamma_1 \cup \Theta_0^\star$ and the latter is Φ_1-satisfiable by hypothesis.

to Ξ^+: $A_{11}, \ldots, A_{nk_n}$ cannot be there because the $C_1, \ldots, C_n$ are productive (see the above claim), thus at least one of the B_j's is in Ξ^+: contradiction. The proof of the Φ_2-satisfiability of $\Gamma_2 \cup \Xi$ is analogous.

We finally show that, given two structures $\mathcal{M}_1 \in \mathcal{S}_1$, $\mathcal{M}_2 \in \mathcal{S}_2$ and two assignments α_1, α_2 such that $\mathcal{M}_1 \models_{\alpha_1} \Gamma_1 \cup \Xi$ and $\mathcal{M}_2 \models_{\alpha_2} \Gamma_2 \cup \Xi$, we have that $\mathcal{M}_1, \alpha_1$ and $\mathcal{M}_2, \alpha_2$ satisfy the same $\Phi_0(\underline{x}_0)$-atoms. This is clear, because Ξ an exhaustive set of $\Phi_0(\underline{x}_0)$-literals. $\square$

5. ISOMORPHISM THEOREMS AND COMPLETENESS

Proposition 4.12 explains what is the main problem for completeness: we would like an open branch to produce Φ_i-structures ($i = 1, 2$) whose $\mathcal{L}_0$-reducts are isomorphic and we are only given Φ_i-structures whose $\mathcal{L}_0$-reducts are $\Phi_0(\underline{x}_0)$-equivalent (in the sense that they satisfy the same $\Phi_0(\underline{x}_0)$-atoms). Hence we need a powerful criterion which states *when* $\Phi_0(\underline{x}_0)$*-equivalence is sufficient for the existence of an* $\mathcal{L}_0$*-isomorphism*: this criterion will be called an isomorphism theorem. The precise formulation of what we mean by an isomorphism theorem needs some preparation. First of all, we introduce fragments extended with free constants; in fact, whereas the use of variables in constraints was precious so far as it emphasized the role of substitutions (and substitutions act on variables, not on free constants), now it is better to have at hand also the formalism of free constants, otherwise standard model-theoretic results would get an unnatural formulation.

Given an i.a.f. $\Phi = \langle \mathcal{L}, T, \mathcal{S} \rangle$, we denote by $\Phi(\underline{c}) = \langle \mathcal{L}(\underline{c}), T(\underline{c}), \mathcal{S}(\underline{c}) \rangle$ the following i.a.f.: (i) $\mathcal{L}(\underline{c}) := \mathcal{L} \cup \{\underline{c}\}$ is obtained by adding to $\mathcal{L}$ finitely many new constants $\underline{c}$ (the types of these new constants must be types of Φ); (ii) $T(\underline{c})$ contains the terms of the kind $t[\underline{c}, \underline{y}]$ for $t[\underline{x}, \underline{y}] \in T$; (iii) $\mathcal{S}(\underline{c})$ contains precisely the $\mathcal{L}(\underline{c})$-structures whose $\mathcal{L}$-reduct is in $\mathcal{S}$. Fragments like $\Phi(\underline{c})$ are called *finite expansions* of Φ.

Let $\Phi(\underline{c})$ be a finite expansion of $\Phi = \langle \mathcal{L}, T, \mathcal{S} \rangle$ and let $\mathcal{A}, \mathcal{B}$ be $\mathcal{L}(\underline{c})$-structures. We say that $\mathcal{A}$ is $\Phi(\underline{c})$*-equivalent* to $\mathcal{B}$ (written $\mathcal{A} \equiv_{\Phi(\underline{c})} \mathcal{B}$) iff for every closed $\Phi(\underline{c})$-atom A we have that $\mathcal{A} \models A$ iff $\mathcal{B} \models A$. By contrast, we say that $\mathcal{A}$ is $\Phi(\underline{c})$-*isomorphic* to $\mathcal{B}$ (written $\mathcal{A} \simeq_{\Phi(\underline{c})} \mathcal{B}$) iff there is an $\mathcal{L}(\underline{c})$-isomorphism from $\mathcal{A}$ onto $\mathcal{B}$.

We can now specify what we mean by a structural operation on an i.a.f. $\Phi_0 = \langle \mathcal{L}_0, T_0, \mathcal{S}_0 \rangle$. We will be very liberal here and define *structural operation on* Φ_0 any family of correspondences $O = \{O^{\underline{c}_0}\}$ associating with any finite set of free constants $\underline{c}_0$ and with any $\mathcal{A} \in \mathcal{S}_0(\underline{c}_0)$ some $O^{\underline{c}_0}(\mathcal{A}) \in \mathcal{S}_0(\underline{c}_0)$ such that $\mathcal{A} \equiv_{\Phi_0(\underline{c}_0)} O^{\underline{c}_0}(\mathcal{A})$. If no confusion arises, we omit the indication of $\underline{c}_0$ in the notation $O^{\underline{c}_0}(\mathcal{A})$ and write it simply as $O(\mathcal{A})$.

A collection $\mathcal{O}$ of structural operations on Φ_0 is said to *admit a* Φ_0*-isomorphism theorem* if and only if, for every $\underline{c}_0$, for every $\mathcal{A}, \mathcal{B} \in \mathcal{S}_0(\underline{c}_0)$, if $\mathcal{A} \equiv_{\Phi_0(\underline{c}_0)} \mathcal{B}$ then there exist $O_1, O_2 \in \mathcal{O}$ such that $O_1(\mathcal{A}) \simeq_{\Phi_0(\underline{c}_0)} O_2(\mathcal{B})$.

Example 5.1 (Ultrapowers). Ultrapowers [Chang and Keisler 1990] are basic constructions in the model theory of first-order logic. An ultrapower $\prod_{\mathcal{U}}$ (technically, an ultrafilter $\mathcal{U}$ over a set of indices is needed to describe the operation) transforms a first-order structure $\mathcal{A}$ into a first-order structure $\prod_{\mathcal{U}} \mathcal{A}$ which is elementarily equivalent to it (meaning that $\mathcal{A}$ and $\prod_{\mathcal{U}} \mathcal{A}$ satisfy the same first-order sentences). Hence if we take a first-order fragment $\Phi_0 = \langle \mathcal{L}_0, T_0, \mathcal{S}_0 \rangle$ in the sense of Examples 3.6, 3.7, or 3.8, then $\prod_{\mathcal{U}}$ is a structural operation on Φ_0. If Φ_0 is a first-order fragment from Example 3.8, the following deep result in classical model theory (known as the *Keisler-Shelah isomorphism theorem* [Chang and Keisler 1990]) gives here a Φ_0-isomorphism theorem in our sense:

Theorem 5.2 (Keisler-Shelah Isomorphism Theorem). *Let $\mathcal{L}$ be a first-order signature and let $\mathcal{A}, \mathcal{B}$ be $\mathcal{L}$-structures. Then $\mathcal{A}$ is elementarily equivalent to $\mathcal{B}$ iff there is an ultrafilter $\mathcal{U}$ such that the ultrapowers $\prod_{\mathcal{U}} \mathcal{A}$ and $\prod_{\mathcal{U}} \mathcal{B}$ are $\mathcal{L}$-isomorphic.*

We shall mainly be interested into operations that can be extended to a preassigned expanded fragment. Here is the related definition. Let an i.a.f. $\Phi = \langle \mathcal{L}, T, \mathcal{S} \rangle$ extending $\Phi_0 = \langle \mathcal{L}_0, T_0, \mathcal{S}_0 \rangle$ be given; a structural operation O on Φ_0 is Φ-*extensible* if and only if for every $\underline{c}$ and every $\mathcal{A} \in \mathcal{S}(\underline{c})$ there exist $\mathcal{B} \in \mathcal{S}(\underline{c})$ such that

$$\mathcal{B}_{|\mathcal{L}_0(\underline{c}_0)} \simeq_{\Phi_0(\underline{c}_0)} O(\mathcal{A}_{|\mathcal{L}_0(\underline{c}_0)}) \quad \text{and} \quad \mathcal{B} \equiv_{\Phi(\underline{c})} \mathcal{A},$$

(where $\underline{c}_0$ denotes the set of those constants in $\underline{c}$ whose type is a Φ_0-type).

Example 5.3. Taking the reduct of a first-order structure to a smaller signature commutes with ultrapowers; consequently, if we are given two first-order fragments $\Phi = \langle \mathcal{L}, T, \mathcal{S} \rangle$ and $\Phi_0 = \langle \mathcal{L}_0, T_0, \mathcal{S}_0 \rangle$ from Examples 3.6, 3.7, or 3.8 such that Φ is an extension of Φ_0, then the Φ_0-structural operation $\prod_{\mathcal{U}}$ is Φ-extensible (the structure $\mathcal{B}$ required in the definition of Φ-extensibility is again $\prod_{\mathcal{U}} \mathcal{A}$, where the ultrapower is now taken at the level of $\mathcal{L}$-structures).

Sometimes an isomorphism theorem does not hold precisely for a fragment $\Phi_0 = \langle \mathcal{L}_0, T_0, \mathcal{S}_0 \rangle$, but for a variation (called specialization) of it. A *specialization* of Φ_0 is an i.a.f. $\Phi_0^\star = \langle \mathcal{L}_0, T_0, \mathcal{S}_0^\star \rangle$, which has the same signature and the same terms as Φ_0, but whose class of $\mathcal{L}_0$-structures is a smaller class $\mathcal{S}_0^\star \subseteq \mathcal{S}_0$ satisfying the following condition: for every $\underline{c}_0$ and for every $\mathcal{A} \in \mathcal{S}_0(\underline{c}_0)$, there exists $\mathcal{A}^\star \in \mathcal{S}_0^\star(\underline{c}_0)$ such that $\mathcal{A} \equiv_{\Phi_0(\underline{c}_0)} \mathcal{A}^\star$. Thus, the condition simply means that Φ_0 and its specialization $\Phi_0^\star$ satisfy the same generalized constraints.

Given an i.a.f. $\Phi = \langle \mathcal{L}, T, \mathcal{S} \rangle$ extending Φ_0, we say that Φ is *compatible* with respect to a given specialization $\Phi_0^\star = \langle \mathcal{L}_0, T_0, \mathcal{S}_0^\star \rangle$ of Φ_0 iff $\Phi^\star = \langle \mathcal{L}, T, \mathcal{S}^\star \rangle$ is a specialization of Φ, where $\mathcal{S}^\star$ is defined by $\mathcal{S}^\star := \{\mathcal{A} \in \mathcal{S} \mid \mathcal{A}_{|\mathcal{L}_0} \in \mathcal{S}_0^\star\}$.

This Φ_0-compatibility notion is intended to recapture, in our general setting, T_0-compatibility as introduced in [Ghilardi 2004]. The latter generalizes, in its turn, the standard stable infiniteness requirement of Nelson-Oppen procedure:

Example 5.4 (Stably Infinite First-Order Theories). Let $\Phi = \langle \mathcal{L}, T, \mathcal{S} \rangle$ be an i.a.f. of the kind considered in Example 3.6 or in Example 3.7: we say that Φ is *stably infinite* iff every satisfiable Φ-constraint is satisfiable in some infinite $\mathcal{L}$-structure $\mathcal{A} \in \mathcal{S}$. To see that this is a Φ_0-compatibility requirement, consider the i.a.f. $\Phi_0 = \langle \mathcal{L}_0, T_0, \mathcal{S}_0 \rangle$ so specified: (i) $\mathcal{L}_0$ is the empty one-sorted signature; (ii) T_0 contains only the individual variables; (iii) $\mathcal{S}_0$ is the totality of $\mathcal{L}_0$-structures (i.e. the totality of sets). A specialization $\Phi_0^\star$ of Φ_0 is obtained by restricting $\mathcal{S}_0$ to the class $\mathcal{S}_0^\star$ formed by the infinite sets. By a compactness argument, it is easily seen that Φ is stably infinite iff it is compatible with respect to the specialization $\Phi_0^\star$ of Φ_0.

5.1 The Main Combination Result

We are now ready to formulate a sufficient condition for our combined procedure to be complete:

Proposition 5.5. *Suppose that Φ_1, Φ_2 are both Φ_0-compact and Φ_0-compatible with respect to a specialization $\Phi_0^\star$ of Φ_0; suppose also that there is a collection $\mathcal{O}$ of structural operations on $\Phi_0^\star$ which are all $\Phi_1^\star$- and $\Phi_2^\star$-extensible and admit a $\Phi_0^\star$-isomorphism theorem. In this case, if the Φ_i-enumerators Res_{Φ_i} are complete and* *the procedure* TCOMB *does not return "unsatisfiable" once applied to the purified constraint* $\Gamma_1 \cup \Gamma_2$, *then such a constraint is* $\Phi_1 \oplus \Phi_2$*-satisfiable.*

PROOF. By Proposition 4.12, there are two structures $\mathcal{M}_1$, $\mathcal{M}_2$ and two assignments α_1, α_2 such that: (i) $\mathcal{M}_1 \in \mathcal{S}_1$, $\mathcal{M}_2 \in \mathcal{S}_2$; (ii) $\mathcal{M}_1 \models_{\alpha_1} \Gamma_1$ and $\mathcal{M}_2 \models_{\alpha_2} \Gamma_2$; (iii) $\mathcal{M}_1, \alpha_1$ and $\mathcal{M}_2, \alpha_2$ satisfy the same $\Phi_0(\underline{x}_0)$-atoms. If we put variables into bijective correspondence with free constants, we may identify the pairs $(\mathcal{M}_i, \alpha_i)$ with structures in $\mathcal{S}_i(\underline{c}_i)$, for finite sets of free constants $\underline{c}_i$. Thus we can say that there are structures $\mathcal{N}_1 \in \mathcal{S}(\underline{c}_1), \mathcal{N}_2 \in \mathcal{S}(\underline{c}_2)$ satisfying $\Gamma_1[\underline{c}_1], \Gamma_2[\underline{c}_2]$, respectively, such that $\mathcal{N}_{1|\mathcal{L}_0(\underline{c}_0)} \equiv_{\Phi_0(\underline{c}_0)} \mathcal{N}_{2|\mathcal{L}_0(\underline{c}_0)}$ (where $\underline{c}_0 = \underline{c}_1 \cap \underline{c}_2$ are the free constants whose types are Φ_0-types). Now we show that there is a $\mathcal{L}_1(\underline{c}_1) \cup \mathcal{L}_2(\underline{c}_2)$-structure $\mathcal{M}$ such that $\mathcal{M}_{|\mathcal{L}_i} \in \mathcal{S}_i$ and $\mathcal{M} \models \Gamma_i[\underline{c}_i]$ $(i = 1, 2)$.

By Φ_0-compatibility with respect to $\Phi_0^\star$, we may assume that $\mathcal{N}_{1|\mathcal{L}_0(\underline{c}_0)}$ and $\mathcal{N}_{2|\mathcal{L}_0(\underline{c}_0)}$ are in a class $\mathcal{S}_0^\star$, over which the collection of structural operations $\mathcal{O}$ admits an isomorphism theorem. Thus there are two structural operations $O_1, O_2 \in \mathcal{O}$ such that $O_1(\mathcal{N}_{1|\mathcal{L}_0(\underline{c}_0)}) \simeq_{\Phi_0(\underline{c}_0)} O_2(\mathcal{N}_{2|\mathcal{L}_0(\underline{c}_0)})$. Since O_1, O_2 are $\Phi_1^\star$- and $\Phi_2^\star$-extensible, there exist two structures $\mathcal{B}_1 \in \mathcal{S}_1^\star(\underline{c}_1)$ and $\mathcal{B}_2 \in \mathcal{S}_2^\star(\underline{c}_2)$ such that $\mathcal{B}_{1|\mathcal{L}_0(\underline{c}_0)} \simeq_{\Phi_0(\underline{c}_0)} O_1(\mathcal{N}_{1|\mathcal{L}_0(\underline{c}_0)})$ and $\mathcal{B}_{2|\mathcal{L}_0(\underline{c}_0)} \simeq_{\Phi_0(\underline{c}_0)} O_2(\mathcal{N}_{2|\mathcal{L}_0(\underline{c}_0)})$, $\mathcal{B}_1 \equiv_{\Phi_1(\underline{c}_1)} \mathcal{N}_1$ and $\mathcal{B}_2 \equiv_{\Phi_2(\underline{c}_2)} \mathcal{N}_2$. Thus, $\mathcal{B}_1$ satisfies $\Gamma_1[\underline{c}_1]$ and $\mathcal{B}_2$ satisfies $\Gamma_2[\underline{c}_2]$. Moreover, $\mathcal{B}_{1|\mathcal{L}_0(\underline{c}_0)} \simeq_{\Phi_0(\underline{c}_0)} \mathcal{B}_{2|\mathcal{L}_0(\underline{c}_0)}$; we can now easily build the desired $\mathcal{M}$ in two steps.

In the first step, we define $\mathcal{B}'_2$ such that $\mathcal{B}_{1|\mathcal{L}_0(\underline{c}_0)} = \mathcal{B}'_{2|\mathcal{L}_0(\underline{c}_0)}$ and $\mathcal{B}_2 \simeq_{\Phi_2(\underline{c}_2)} \mathcal{B}'_2$ (notice that $\mathcal{B}'_2 \in \mathcal{S}_2(\underline{c}_2)$ by the closure under isomorphisms of $\mathcal{S}_2$, see Definition 3.3). Let ι be the isomorphism $\mathcal{B}_{1|\mathcal{L}_0(\underline{c}_0)} \longrightarrow \mathcal{B}_{2|\mathcal{L}_0(\underline{c}_0)}$; to define $\mathcal{B}'_2$, we interpret $\mathcal{L}_0$-sorts as in $\mathcal{B}_1$ and $\mathcal{L}_2 \setminus \mathcal{L}_0$-sorts as in $\mathcal{B}_2$. Put now $\iota'_S := \iota_S$ for $S \in \mathcal{L}_0$ and let ι'_S be the identity for $\mathcal{L}_2 \setminus \mathcal{L}_0$-sorts; by taking standard inductive extension to all $\mathcal{L}_2$-types, we get a family of bijections $\iota' = \{\iota'_\tau : [\![\tau]\!]_{\mathcal{B}'_2} \longrightarrow [\![\tau]\!]_{\mathcal{B}_2}\}$ (indexed by the $\mathcal{L}_2$-types) that can be used in order to complete the definition of $\mathcal{B}'_2$ (in the sense that we define the $\mathcal{B}'_2$-interpretation of every constant $d : \tau$ of $\mathcal{L}_2(\underline{c}_2)$ as $(\iota'_\tau)^{-1}(\mathcal{I}_{\mathcal{B}_2}(d))$). It is easily seen that the $\mathcal{L}_2(\underline{c}_2)$-structure $\mathcal{B}'_2$ matches the desired requirements.

Since the $\mathcal{L}_0(\underline{c}_0)$-reducts of $\mathcal{B}_1$ and $\mathcal{B}'_2$ are now just the same structure, it is easy to define (through a trivial join of both sorts and constants interpretations) a $\mathcal{L}_1(\underline{c}_1) \cup \mathcal{L}_2(\underline{c}_2)$-structure $\mathcal{M}$ such that $\mathcal{M}_{|\mathcal{L}_1(\underline{c}_1)} = \mathcal{B}_1$ and $\mathcal{M}_{|\mathcal{L}_2(\underline{c}_2)} = \mathcal{B}'_2$. Thus, the $\mathcal{L}_1 \oplus \mathcal{L}_2$-reduct of $\mathcal{M}$ belongs to $\mathcal{S}_1 \oplus \mathcal{S}_2$ and satisfies $\Gamma_1[\underline{x}_1] \cup \Gamma_2[\underline{x}_2]$. $\square$

The facts we established so far can be collected into our main combination theorem:

THEOREM 5.6 (MAIN DECIDABILITY TRANSFER THEOREM). *Suppose that:*

(1) *the interpreted algebraic fragments* Φ_1, Φ_2 *have decidable constraint satisfiability problems;*

(2) *the shared fragment* Φ_0 *is effectively locally finite (or, more generally,* Φ_1, Φ_2 *are both* Φ_0*-compact,* Φ_0 *is noetherian and there exist noetherian positive residue* Φ_1*- and* Φ_2*-enumerators for* Φ_0*);*

(3) Φ_1 *and* Φ_2 *are both* Φ_0*-compatible with respect to a specialization* $\Phi_0^\star$ *of* Φ_0*;*

(4) *there is a collection* $\mathcal{O}$ *of structural operations on* $\Phi_0^\star$ *which are all* $\Phi_1^\star$*- and* $\Phi_2^\star$*-extensible and admit a* $\Phi_0^\star$*-isomorphism theorem.*

Then the tableau procedure TComb *(together with the preprocessing Purification Rule) decides constraint satisfiability in the combined fragment* $\Phi_1 \oplus \Phi_2$.

Proof. From Propositions 4.6, 4.7, 4.8,4.9, 3.25, 4.10, 4.11, 3.24, and 5.5. $\square$

Remark. Theorem 5.6 cannot be used to transfer decidability of word problems to our combined fragments: the reason is that, in case the procedure TComb is initialized with only a single negative literal, constraints containing positive literals are generated during the execution (and also by the Purification Rule). However, since negative literals are never run-time generated, Theorem 5.6 can be used to transfer decidability of *conditional word problems*, namely of satisfiability problems for constraints containing just one negative literal. Notice that in convex fragments, conditional word problems and constraint satisfiability problems are inter-reducible.

In the next two subsections we investigate families of concrete applications of Theorem 5.6, based on suitable isomorphism theorems.

5.2 Applications: Decidability Transfer through Ultrapowers

We shall use the isomorphism Theorem 5.2 to get the transfer decidability results of [Ghilardi 2004] as a special case of Theorem 5.6. Let $\Phi_1 = \langle \mathcal{L}_1, T_1, \mathcal{S}_1 \rangle$ and $\Phi_2 = \langle \mathcal{L}_2, T_2, \mathcal{S}_2 \rangle$ be universal first-order i.a.f.'s (i.e. i.a.f.'s of the kind considered in Example 3.7) and let $\Phi_0 = \langle \mathcal{L}_0, T_0, \mathcal{S}_0 \rangle$ be their shared fragment. The hypothesis for the decidability transfer result of [Ghilardi 2004] are (equivalent to) the following:

(C1) there is an $\mathcal{L}_0$-theory $T_0^\star$ admitting elimination of quantifiers such that, taking $\mathcal{S}_0^\star$ to be the class of all models of $T_0^\star$, we have that $\mathcal{S}_0^\star \subseteq \mathcal{S}_0$ and that $\Phi_0^\star = \langle \mathcal{L}_0, T_0, \mathcal{S}_0^\star \rangle$ is a specialization of Φ_0;

(C2) Φ_1 and Φ_2 are both compatible with respect to $\Phi_0^\star$;

(C3) Φ_0 is effectively locally finite.

Theorem 5.7 [Ghilardi 2004]. *Suppose that* Φ_1 *and* Φ_2 *are first-order universal i.a.f.'s satisfying conditions* (C1)-(C3) *above; if constraint satisfiability problems are decidable in* Φ_1 *and* Φ_2*, then they are decidable in* $\Phi_1 \oplus \Phi_2$ *too.*

Proof. We check the conditions of Theorem 5.6. Since we have two decision procedures for the constraint satisfiability problems in Φ_1 and Φ_2, 5.6 (1) holds; (C3) guarantees 5.6 (2) and 5.6 (3) follows from (C2). To check the remaining condition 5.6 (4), we use ultrapowers and the isomorphism Theorem 5.2.

Let us first check that for every $\underline{c}_0$, if $\mathcal{A}, \mathcal{B} \in \mathcal{S}_0^\star(\underline{c}_0)$ are such that $\mathcal{A} \equiv_{\Phi_0(\underline{c}_0)} \mathcal{B}$, then $\mathcal{A}$ and $\mathcal{B}$ are $\mathcal{L}_0(\underline{c}_0)$-elementarily equivalent. In fact a first-order $\Phi_0(\underline{c}_0)$-sentence $\varphi[\underline{c}_0]$ which is true in $\mathcal{A}$ is true also in $\mathcal{B}$, for the following reason. Since both structures are models of $T_0^\star$ and since $T_0^\star$ eliminates quantifiers, $\varphi[\underline{c}_0]$ is equivalent modulo $T_0^\star$ to some quantifier-free $\varphi'[\underline{c}_0]$: the latter holds in $\mathcal{A}$ iff it holds in $\mathcal{B}$, because $\mathcal{A} \equiv_{\Phi_0(\underline{c}_0)} \mathcal{B}$ (recall that atomic formulae are atoms in the fragments of the kind of Example 3.7).

Since, within the class $\mathcal{S}_0^\star(\underline{c}_0)$, we have that $\equiv_{\Phi_0(\underline{c}_0)}$ is the same as elementary equivalence, Keisler-Shelah Theorem implies that ultrapowers are a collection of structural operations on $\Phi_0^\star$ admitting a $\Phi_0^\star$-isomorphism theorem.

It remains to check extensibility; to this aim, recall that $\mathcal{S}_i$ is an elementary class (by the conditions from Example 3.7), hence $\mathcal{S}_i^\star$ is an elementary class too (we have $\mathcal{A} \in \mathcal{S}_i^\star$ iff both $\mathcal{A} \in \mathcal{S}_i$ and $\mathcal{A} \models T_0^\star$). For elementary classes of structures, extensibility of ultrapowers as structural operations has been already observed in Example 5.3. □

As a special case, suppose that the signatures $\mathcal{L}_1, \mathcal{L}_2$ are disjoint (i.e. they share just their unique sort and no other proper symbol) and that $\mathcal{S}_1, \mathcal{S}_2$ are the classes of the models of two (consistent) stably infinite first-order theories T_1, T_2; if we take $T_0^\star$ to be the theory of an infinite set, then it is immediately seen that conditions (C1)-(C3) are satisfied (see the discussion in Example 5.4). Thus, Theorem 5.7 *implies the standard Nelson-Oppen result* [Nelson and Oppen 1979; Oppen 1980; Tinelli and Harandi 1996] *concerning stably infinite theories over disjoint signatures.*

We recall from [Ghilardi 2004] that among relevant examples of theories to which Theorem 5.7 is easily seen to apply, we have the (convex) theories axiomatizing varieties of Boolean algebras with operators: thus, decidability of conditional word problem transfers from two such theories to their union (provided only Boolean operators are shared). This result, proved in [Wolter 1998] by specific techniques, is the algebraic version of the *fusion transfer of decidability of global consequence relation in modal logic.*

We remark that condition (C3) can be weakened to

(C3′). Φ_0 is noetherian and there exist noetherian positive residue Φ_1- and Φ_2-enumerators for Φ_0

(as suggested by Theorem 5.6 (2)): we give an example of an application of Theorem 5.7 under this weaker condition.

Example 5.8 (A Combination of noetherian fragments). Consider the combined fragment $\Phi_1 \oplus \Phi_2$, where Φ_1 is the fragment Φ_{Kalg} of the Example 3.27 and Φ_2 is the fragment Φ_{Kend} of Example 3.30 (here, however, we require K-algebras to be non degenerate, i.e. to satisfy the condition $0 \neq 1$). From Definition 4.2, it follows that the class $\mathcal{S}_1 \oplus \mathcal{S}_2$ consists of the models of the theory of the non degenerate K-algebras endowed with a linear endomorphism (i.e. endowed with a function f preserving sum and scalar multiplication). The class $\mathcal{S}_0$ of the structures of the shared fragment Φ_0 consists of the models of the theory T_0 of K-vector spaces. As $T_0^\star$, we take the theory $T_0 \cup \{\exists x\,(x \neq 0)\}$, if K is an infinite field, and the theory $T_0 \cup \{\exists x_1 \cdots \exists x_n \bigwedge_{i \neq j} x_i \neq x_j\}_{n \in \mathbb{N}}$, otherwise: in both cases, the models of $T_0^\star$ are just infinite K-vector spaces. Thus conditions (C1) is easily seen to be satisfied. Since every non degenerate K-algebra (resp. every f-K-vector space) can be embedded into an infinite K-algebra (resp. into an infinite f-K-vector space), condition (C2) holds too. Condition (C3′) is also satisfied, as pointed out in Subsection 3.4 when discussing Examples 3.27, 3.28 and 3.30. Hence the combination procedure TComb decides conditional word problems for the theory of (non degenerate) K-algebras endowed with a linear endomorphism.

As another application of Theorem 5.6 based on Keisler-Shelah isomorphism theorem, we show *how to include a first-order equational theory within an A-Box.*

An *A-Box fragment* is an i.a.f. of the kind $\Phi_{ML} = \langle \mathcal{L}_{ML}, T_{ML}, \mathcal{S}_{ML} \rangle$, where $\langle \mathcal{L}_{ML}, T_{ML} \rangle$ is defined (out of a modal signature O_M) as in Example 3.11 and $\mathcal{S}_{ML}$ is a class of $\mathcal{L}_{ML}$-structures closed under isomorphisms and disjoint I-copies. The latter operation is defined as follows:

Definition 5.9 *(Disjoint I-copy).* Consider a first-order one-sorted relational signature $\mathcal{L}$ and a (non empty) index set I. The operation $\sum_I$, defined on $\mathcal{L}$-structures and called *disjoint I-copy*, associates with an $\mathcal{L}$-structure $\mathcal{M} = \langle [\![-]\!]_{\mathcal{M}}, \mathcal{I}_{\mathcal{M}} \rangle$ the $\mathcal{L}$-structure $\sum_I \mathcal{M}$ such that $[\![W]\!]_{\sum_I \mathcal{M}}$ is the disjoint union of I-copies of $[\![W]\!]_{\mathcal{M}}$

(here W is the unique sort of $\mathcal{L}$). The interpretation of relational predicates is defined as follows (elements of the disjoint union of I-copies of a set S are represented as s^i - meaning that s^i is the i-th copy of $s \in S$)

$$\sum_I \mathcal{M} \models P(d_1^{i_1}, \ldots, d_n^{i_n}) \quad \Longleftrightarrow \quad i_1 = i_2 = \cdots = i_n \text{ and } \mathcal{M} \models P(d_1, \ldots, d_n) \quad (5)$$

for every n-ary predicate P. $\dashv$

Disjoint I-copy is a special case of a more general disjoint union operation: the latter is defined again by (5) and applies to any I-indexed family of structures (which may not coincide with each other). Our specific interest for disjoint I-copies is motivated by the following Lemma, concerning satisfiability of packed guarded formulae:[21]

LEMMA 5.10. *Consider a first-order one-sorted relational signature $\mathcal{L}$, the $\mathcal{L}$-structure $\mathcal{M}$ and its disjoint I-copy $\sum_I \mathcal{M}$. The following statements hold:*

(i) *for every elementary packed guarded formula $\varphi[x_1, \ldots, x_n]$ ($n \geq 0$), for every $d_1, \ldots, d_n$ in the support of $\mathcal{M}$ and for every index $i \in I$, we have that*

$$\sum_I \mathcal{M} \models \varphi[d_1^i, \ldots, d_n^i] \quad \Longleftrightarrow \quad \mathcal{M} \models \varphi[d_1, \ldots, d_n];$$

(ii) *a packed guarded elementary sentence is satisfiable in $\mathcal{M}$ iff it is satisfiable in $\sum_I \mathcal{M}$.*

PROOF. We check the first claim by induction on φ (the second claim follows immediately for the case $n = 0$). If φ is atomic, just apply (5), and the case of Boolean connectives is immediate. Suppose now that φ is the packed guarded existential quantification

$$\exists y_1 \cdots \exists y_m (\pi[x_{i_1}, \ldots, x_{i_k}, y_1, \ldots, y_m] \wedge \psi[x_{i_1}, \ldots, x_{i_k}, y_1, \ldots, y_m])$$

where $x_{i_1}, \ldots, x_{i_k}$ are the variables among $x_1, \ldots, x_n$ that really occur free in $\varphi[x_1, \ldots, x_n]$ (notice that they must all occur free in the guard π, as well as the $y_1, \ldots, y_m$). That $\mathcal{M} \models \varphi[d_1, \ldots, d_n]$ implies $\sum_I \mathcal{M} \models \varphi[d_1^i, \ldots, d_n^i]$ is trivial; for the converse suppose that

$$\sum_I \mathcal{M} \models \pi[d_{i_1}^i, \ldots, d_{i_k}^i, e_1^{j_1}, \ldots, e_m^{j_m}] \wedge \psi[d_{i_1}^i, \ldots, d_{i_k}^i, e_1^{j_1}, \ldots, e_m^{j_m}]$$

for some $e_1^{j_1}, \ldots, e_m^{j_m}$. By (5) and the definition of a guard, all indices $j_1, \ldots, j_m$ must be equal to some j (and, if $k \neq 0$, then j must be i). As a consequence, $\mathcal{M} \models \pi[d_{i_1}, \ldots, d_{i_k}, e_1, \ldots, e_m] \wedge \psi[d_{i_1}, \ldots, d_{i_k}, e_1, \ldots, e_m]$ holds by induction hypothesis and by (5). $\square$

The following result combines an equational first-order i.a.f. $\Phi = \langle \mathcal{L}, T, \mathcal{S} \rangle$ from Example 3.6 and an A-Box fragment $\Phi_{ML} = \langle \mathcal{L}_{ML}, T_{ML}, \mathcal{S}_{ML} \rangle$:

[21] See Example 3.15 for the related definition. Lemma 5.10 can be seen as a special case of invariance under a suitable notion of 'guarded' bisimulation.

Theorem 5.11. *Suppose that we are given an equational first-order i.a.f.* $\Phi = \langle \mathcal{L}, T, \mathcal{S} \rangle$ *and an A-Box fragment* $\Phi_{ML} = \langle \mathcal{L}_{ML}, T_{ML}, \mathcal{S}_{ML} \rangle$*; suppose also that the signatures* $\mathcal{L}$ *and* $\mathcal{L}_{ML}$ *are disjoint, that* Φ *is stably infinite and that* $\mathcal{S}_{ML}$ *is an elementary class. Then decidability of constraint satisfiability problems transfers from* Φ *and* Φ_{ML} *to* $\Phi \oplus \Phi_{ML}$.

Proof. We preliminarily observe that formulae $ST(C, w)$ are packed guarded, hence if we replace in them the second order variables of type $W \to \Omega$ by free constants for subsets of W (which are first-order relational symbols), Lemma 5.10 (i) applies to them.

Since our data are all first-order, the argument in the proof of Theorem 5.7 works, provided conditions (C1)-(C3) hold. Notice that terms in the shared fragment are just first-order individual variables; we take as $T_0^\star$ the theory of an infinite set, so that we only have to check condition (C2) for both Φ and Φ_{ML}. For the former, the condition holds trivially by stably infiniteness; for the latter, given a $\mathcal{L}_{ML}(\underline{c})$-structure $\mathcal{A}(\underline{c})$, we must produce an infinite $\mathcal{L}_{ML}(\underline{c})$-structure $\mathcal{B}(\underline{c})$ such that $\mathcal{A}(\underline{c}) \equiv_{\Phi_{ML}(\underline{c})} \mathcal{B}(\underline{c})$. To this aim, let I be an infinite set and let us take the disjoint I-copy $\sum_I \mathcal{A}(\underline{c})$. However, Definition 5.9 tells us how to interpret binary and unary predicate symbols from $\mathcal{L}_{ML}(\underline{c})$, but not the free individual constants c that might occur in $\underline{c}$: we interpret them as $\mathcal{I}_{\mathcal{A}(\underline{c})}(c)^i$, where i is some arbitrarily chosen element of I (to be the same for all the free individual constants c that belong to $\underline{c}$). By Lemma 5.10(i), we can now conclude that $\mathcal{A}(\underline{c}) \equiv_{\Phi_{ML}(\underline{c})} \sum_I \mathcal{A}(\underline{c})$ holds, as desired. $\square$

The fragment $\Phi \oplus \Phi_{ML}$ of Theorem 5.11 is quite peculiar, because combined terms all arise from a single composition step (they all have degree 2, in the terminology of Lemma 4.4). To understand a possible meaning of such a fragment, notice that individual terms from Φ may be seen as functions that enrich the domain of a Φ_{ML}-structure with some more concrete algebraic operations (e.g., numerical operations, data structure operations, etc.). Thus the fragments $\Phi \oplus \Phi_{ML}$ of Theorem 5.11 may play a role similar to concrete domains in description logics [Baader and Hanschke 1991] (it should be noted, however, that description logics with concrete domains are two-sorted formalisms, whereas the combination schema of Theorem 5.11 is one-sorted).

5.3 Applications: Decidability Transfer through Disjoint Copies

Disjoint copies are the key tool for decidability transfer results in modal fragments too. Let O_M be a modal signature, as defined in Example 3.10. A *modal i.a.f. over* O_M is a fragment of the kind $\Phi_M = \langle \mathcal{L}_M, T_M, \mathcal{S}_M \rangle$, where $\mathcal{L}_M$ and T_M are as defined in Example 3.10, whereas $\mathcal{S}_M$ is a class of $\mathcal{L}_M$-structures *closed under isomorphisms and disjoint* I*-copies.* In the following, we indicate by O_{M_0} the empty modal signature.

Proposition 5.12. *Let* Φ_M *be a modal i.a.f. over the modal signature* O_M *and consider a modal subfragment* Φ_{M_0} *of it, based on the empty modal signature; the structural operations* $\{\sum_I\}_I$ *over* Φ_{M_0} *are* Φ_M*-extensible and form a collection admitting a* Φ_{M_0}*-isomorphism theorem.*

Proof. Recall from Example 3.10 that $W \to \Omega$ is the only type of the i.a.f.

Φ_M, hence the relevant free constants $\underline{c}$ in expanded signatures are constants for subsets of W (this means, in particular, that their interpretation can be extended to disjoint I-copies like any other relational first-order predicate symbol as shown in Definition 5.9).

That taking disjoint I-copies $\sum_I$ is a structural Φ_M-extensible operation is clear: to define $\mathcal{N} \in \mathcal{S}_M(\underline{c})$ which is $\Phi_M(\underline{c})$-equivalent to some $\mathcal{M} \in \mathcal{S}_M(\underline{c})$ and whose $\Phi_{M_0}(\underline{c})$-reduct is $\mathcal{L}_{M_0}(\underline{c})$-isomorphic to $\sum_I(\mathcal{M}_{|\mathcal{L}_{M_0}(\underline{c})})$ it is sufficient to take the $\mathcal{L}_M(\underline{c})$-structure $\sum_I \mathcal{M}$ as $\mathcal{N}$ and apply Lemma 5.10 (ii) (recall that constraints in Φ_M are equivalent to conjunctions of formulae of the kind $\forall w ST(C, w)$ and their negations). That taking disjoint I-copies is a structural operation (i.e. that a $\mathcal{L}_{M_0}(\underline{c})$-structure and its disjoint I-copy are $\Phi_{M_0}(\underline{c})$-equivalent) is clear by the same reasons.

To show that a Φ_{M_0}-isomorphism theorem holds, suppose that we are given free constants $\underline{c}_0 := \{P_1, \dots, P_n\}$ and two structures $\mathcal{M}_1$ and $\mathcal{M}_2$ in $\mathcal{S}_{M_0}(\underline{c}_0)$ such that $\mathcal{M}_1 \equiv_{\Phi_{M_0}(\underline{c}_0)} \mathcal{M}_2$; we show that $\sum_I \mathcal{M}_1 \simeq_{\Phi_{M_0}(\underline{c}_0)} \sum_I \mathcal{M}_2$ holds for some I.

Consider every boolean combination of the form $\varepsilon(w) = Q_1(w) \wedge \cdots \wedge Q_n(w)$ where $Q_j \equiv P_j$ or $Q_j \equiv \neg P_j$ (thus the number of such formulae is 2^n). For a given $\mathcal{L}_{M_0}(\underline{c}_0)$-structure $\mathcal{N}$, let $\varepsilon(\mathcal{N}) := \{a \in [\![W]\!]_{\mathcal{N}} \mid \mathcal{N} \models \varepsilon(a)\}$ and let us associate with $\mathcal{N}$ the 2^n cardinal invariants $a_\varepsilon(\mathcal{N}) := \sharp\varepsilon(\mathcal{N})$. Now two $\mathcal{L}_{M_0}(\underline{c}_0)$-structures $\mathcal{N}_1$ and $\mathcal{N}_2$ having the same invariants are isomorphic, because we can glue bijections $\varepsilon(\mathcal{N}_1) \longrightarrow \varepsilon(\mathcal{N}_2)$ to a $\mathcal{L}_{M_0}(\underline{c}_0)$-isomorphism $\mathcal{N}_1 \simeq \mathcal{N}_2$.

Finally, we note that $\mathcal{M}_1 \equiv_{\Phi_{M_0}(\underline{c}_0)} \mathcal{M}_2$ means that $\mathcal{M}_1 \models A$ holds iff $\mathcal{M}_2 \models A$ holds for every closed $\Phi_{M_0}(\underline{c}_0)$-atom A. In particular, $\mathcal{M}_1 \models \{w \mid \varepsilon(w)\} = \{w \mid \bot\}$ iff $\mathcal{M}_2 \models \{w \mid \varepsilon(w)\} = \{w \mid \bot\}$: thus $\varepsilon(\mathcal{M}_1) = \emptyset$ iff $\varepsilon(\mathcal{M}_2) = \emptyset$ holds for all ε. Let now consider a set I whose cardinality m is such that $m \geq \varepsilon(\mathcal{M}_i)$ for all ε and for $i \in \{1, 2\}$: we show that $\sum_I \mathcal{M}_1 \simeq_{\Phi_{M_0}(\underline{c}_0)} \sum_I \mathcal{M}_2$ proving that the two structures have the same invariants. In fact the cardinal identities $a_\varepsilon(\sum_I \mathcal{M}_1) = m \cdot a_\varepsilon(\mathcal{M}_1) = m = m \cdot a_\varepsilon(\mathcal{M}_2) = a_\varepsilon(\sum_I \mathcal{M}_2)$ hold for all ε. $\square$

We indicate by $O_{M_1 \oplus M_2}$ the disjoint union of the modal signatures O_{M_1} and O_{M_2} ($O_{M_1 \oplus M_2}$ is called the fusion of the modal signatures O_{M_1} and O_{M_2}). Given a modal i.a.f. Φ_{M_1} over O_{M_1} and a modal i.a.f. Φ_{M_2} over O_{M_2}, let us define their *fusion* as the modal i.a.f.

$$\Phi_{M_1 \oplus M_2} = \langle \mathcal{L}_{M_1 \oplus M_2}, T_{M_1 \oplus M_2}, \mathcal{S}_{M_1} \oplus \mathcal{S}_{M_2} \rangle.$$

Let the two modal i.a.f.'s $\Phi_{M_1} = \langle \mathcal{L}_{M_1}, T_{M_1}, \mathcal{S}_{M_1} \rangle$ and $\Phi_{M_2} = \langle \mathcal{L}_{M_2}, T_{M_2}, \mathcal{S}_{M_2} \rangle$ have disjoint modal signatures; the shared fragment $\Phi_{M_0} = \langle \mathcal{L}_{M_0}, T_{M_0}, \mathcal{S}_{M_0} \rangle$ is locally finite, because it is a modal i.a.f. over the empty modal signature (for any finite set of Φ_{M_0}-variables $\underline{x}_0$, the $\Phi_{M_0}(\underline{x}_0)$-terms are terms of the kind $\{w \mid \psi(w)\}$, where ψ is a boolean combination of the second order variables $\underline{x}_0$).

Now if Φ_{M_1} and Φ_{M_2} have decidable constraint satisfiability problems, then so does the combined i.a.f. $\Phi_{M_1} \oplus \Phi_{M_2}$: in fact, the hypotheses of Theorem 5.6 are satisfied by Proposition 5.12.[22] To infer the transfer decidability result for the fusion modal i.a.f., we need to clarify the relationship between $\Phi_{M_1 \oplus M_2}$ and $\Phi_{M_1} \oplus \Phi_{M_2}$. These two i.a.f.'s have the same signatures and the classes of structures in which

[22]We obviously take $\Phi_0^\star$ to be Φ_{M_0} in 5.6 (3).

they are interpreted are also the same; there is however a little difference between their sets of terms.

Given two i.a.f.'s $\Phi_1 = \langle \mathcal{L}_1, T_1, \mathcal{S}_1 \rangle$ and $\Phi_2 = \langle \mathcal{L}_2, T_2, \mathcal{S}_2 \rangle$, we say that they are *$\beta\eta$-equivalent* (written $\Phi_1 \sim_{\beta\eta} \Phi_2$) iff $\mathcal{L}_1 = \mathcal{L}_2$, $\mathcal{S}_1 = \mathcal{S}_2$ and moreover for every $t_1 \in T_1$ one can effectively compute some $t_2 \in T_2$ such that $t_1 \sim_{\beta\eta} t_2$, and vice versa. Clearly, $\beta\eta$-equivalent i.a.f.'s can be considered to be just the same. The following Lemma is easily proved by exploiting basic properties of $\beta\eta$-equivalence:

LEMMA 5.13. *If $\Phi_{M_1 \oplus M_2}$ and $\Phi_{M_1} \oplus \Phi_{M_2}$ are as above, we have that $\Phi_{M_1 \oplus M_2} \sim_{\beta\eta} \Phi_{M_1} \oplus \Phi_{M_2}$.*

PROOF. Since $T_{M_1} \oplus T_{M_2}$ is defined to be the minimum set of terms closed under substitutions and containing T_{M_1} and T_{M_2} and since $T_{M_1 \oplus M_2}$ enjoys these properties, clearly any $t \in T_{M_1} \oplus T_{M_2}$ belongs to $T_{M_1 \oplus M_2}$.

Conversely, let us take $t \in T_{M_1 \oplus M_2}$; then $t \sim_{\beta\eta} \{w \mid ST(C, w)\}$ for some $O_{M_1 \oplus M_2}$-modal concept C.[23] By induction on C, we define $u \in T_{M_1} \oplus T_{M_2}$ such that $u \sim_{\beta\eta} \{w \mid ST(C, w)\}$ (then $t \sim_{\beta\eta} u$ follows by transitivity). If C is a propositional variable we can take u to be $\{w \mid ST(C, w)\}$. If C is $D_1 \sqcap D_2$, by induction there are $u_1, u_2 \in T_{M_1} \oplus T_{M_2}$ such that $u_i \sim_{\beta\eta} \{w \mid ST(D_i, w)\}$ for $i = 1, 2$. Then $\{w \mid ST(C, w)\} = \{w \mid ST(D_1, w) \wedge ST(D_2, w)\} \sim_{\beta\eta} \{w \mid \{w \mid ST(D_1, w)\}(w) \wedge \{w \mid ST(D_2, w)\}(w)\} \sim_{\beta\eta} \{w \mid u_1(w) \wedge u_2(w)\}$. The latter is obtained by replacing in the term $\{w \mid ST(\mathrm{x}_1 \wedge \mathrm{x}_2, w)\} = \{w \mid X_1(w) \wedge X_2(w)\}$ the terms $u_1, u_2 \in T_{M_1} \oplus T_{M_2}$ for the second order variables X_1, X_2, respectively, hence it is a term that belongs to $T_{M_1} \oplus T_{M_2}$ too, because the latter is closed under substitution. The cases of $\sqcup, \neg, \Diamond_k$ are analogous. $\square$

We have so proved the following well-known decidability transfer result (see, e.g., [Baader et al. 2002] and the literature quoted therein):

THEOREM 5.14 (DECIDABILITY TRANSFER FOR MODAL I.A.F.'S). *If two modal i.a.f.'s Φ_{M_1} and Φ_{M_2} have decidable constraint satisfiability problems, so does their fusion $\Phi_{M_1 \oplus M_2}$.*

Fragments of the kind examined in Example 3.11 are not interesting for being combined with each other, because the absence of the type $W \to \Omega$ makes such combinations trivial. On the contrary, full modal fragments from Example 3.12 are quite interesting in this respect (we recall that they perform both A-Box and T-Box reasoning from the point of view of description logics). In fact very slight modifications are sufficient to get a result analogous to Theorem 5.14: we just sketch how to do it.

Let O_M be a modal signature; a *full modal i.a.f. over O_M* is a fragment of the kind $\Phi_{FM} = \langle \mathcal{L}_{FM}, T_{FM}, \mathcal{S}_{FM} \rangle$, where $\mathcal{L}_{FM}$ and T_{FM} are as defined in Example 3.12, whereas $\mathcal{S}_{FM}$ is again a class of $\mathcal{L}_M$-structures closed under isomorphisms and disjoint I-copies.

There is a little complication arising now: since W is a type of an i.a.f. like Φ_{FM}, when we expand signatures with free constants, we now get (besides constants of

[23]Notice that $\{w \mid ST(C, w)\}$ - hence also C - can be effectively computed because it is in long-$\beta\eta$-normal form and so it is the long-$\beta\eta$-normal form of t.

type $W \to \Omega$) also individual constants of type W. The interpretation of these constants is not defined in disjoint I-copies, because taking disjoint I-copies is an operation defined only for first-order relational signatures. We proceed like in the proof of Theorem 5.11: we take index sets I which are pointed, namely some $i_0 \in I$ is specified. Then, we define the interpretation of an individual constant c of type W in $\sum_I \mathcal{M}$ as $\langle \mathcal{I}_{\mathcal{M}}(c), i_0 \rangle$.

The definition of fusion for full modal i.a.f.'s is the obvious one and it leads to the following result [Baader et al. 2002]:

Theorem 5.15 (Decidability transfer for full modal i.a.f.'s) . *If two full modal i.a.f.'s have decidable constraint satisfiability problems, so does their fusion.*

Proof. We sketch the little modifications required to prove Proposition 5.12 in the present context (Lemma 5.13 extends trivially).

Let Φ_{FM} be a full modal i.a.f. over O_M and let Φ_{FM_0} be a subfragment of it on the empty modal signature. According to the considerations in Examples 3.10-3.11, when considering signatures expanded with free constants $\underline{c}$, closed $\Phi_{FM}(\underline{c})$-atoms are now of the kind $c_1 = c_2$, $R_k(c_1, c_2)$, $ST(D, c)$, and $\forall w ST(D, c)$ (where second order variables in D have been replaced by first-order unary predicate constants). From the pointed definition of disjoint I-copy given above and Lemma 5.10, it is then clear that the $\mathcal{L}_{FM}(\underline{c})$-structures $\mathcal{M}$ and $\sum_I \mathcal{M}$ still are $\Phi_{FM}(\underline{c})$-equivalent and this is all what matters in order to check that pointed disjoint I-copies are Φ_{FM}-extensible structural operations over Φ_{FM_0}.

For the Φ_{FM_0}-isomorphism theorem, we just need to add to the invariants of a $\mathcal{L}_{FM_0}(\underline{c})$-structure $\mathcal{N}$ considered in the proof of Proposition 5.12 also the indication about the truth/falsity in $\mathcal{N}$ of the ground atoms of the kind $\varepsilon(c)$ and $c_1 = c_2$, varying c, c_1, c_2 among the individual constants in $\underline{c}$. $\square$

The statement of Theorem 5.15 seems not to allow the decidability transfer of only *positive* A-Box satisfiability with respect to T-Box axioms; however this further decidability transfer result follows immediately once one realizes that the combined algorithm TComb never adds negative information to current constraints, so if non positive A-Boxes are not present from the very beginning, there won't be any call for a decision procedure involving them (see also the Remark following Theorem 5.6 for the same observation).

The decidability transfer theorem for the non-normal case of Example 3.13 (i.e. for the full strength of abstract description systems in the sense of [Baader et al. 2002]) requires a simple adaptation of Definition 5.9 and of Lemma 5.10. We can also extend our transfer results to fragments involving the μ-calculus fixed-points constructors of Example 3.14: in fact, these constructors are invariant under bisimulation, hence Lemma 5.10 still holds (notice also that fixed points can be eliminated from empty modal signatures, hence local finiteness of the shared fragment is not compromised, even in case we wish to combine two 'μ-fragments' with each other).

We now try to extend our decidability transfer results to cover also combinations of packed guarded and/or of two-variable fragments. However, to get positive results, we need to keep shared signatures under control. In addition, we still want to exploit the isomorphism theorem of Proposition 5.12 and for that reason we need the shared signature to be empty and second order variables appearing as terms in the fragments to be monadic only. The kind of combination that arise in this way can be considered as a form of fusion, that we shall call monadic fusion. We begin by identifying a class of fragments to which our techniques apply.

Let us call $\Phi_\emptyset = \langle \mathcal{L}_\emptyset, T_\emptyset, \mathcal{S}_\emptyset \rangle$ the following i.a.f.: (i) $\mathcal{L}_\emptyset$ is the empty one-sorted first-order signature (that is, $\mathcal{L}_\emptyset$ does not contain any proper symbol, except for its unique sort which is called W); (ii) $T_\emptyset$ is $T_{11}^{\mathcal{L}_\emptyset}$;[24] (iii) $\mathcal{S}_\emptyset$ contains all $\mathcal{L}_\emptyset$-structures.

Definition 5.16. A *monadically suitable* i.a.f. $\Phi = \langle \mathcal{L}, T, \mathcal{S} \rangle$ is an i.a.f. such that:

(i) $\mathcal{L}$ is a relational one-sorted first-order signature;

(ii) $T_{11}^{\mathcal{L}_\emptyset} \subseteq T \subseteq T_{\omega 1}^{\mathcal{L}}$;

(iii) the $\Phi_\emptyset$-structural operation of taking disjoint I-copies is Φ-extensible. ⊣

We give a couple of interesting examples of monadically suitable decidable fragments:

EXAMPLE 5.17. *Packed guarded fragments* are i.a.f.'s of the kind $\Phi_G = \langle \mathcal{L}_G, T_G, \mathcal{S}_G \rangle$, where T_G is as defined in Example 3.15, whereas $\mathcal{S}_G$ is a class of $\mathcal{L}_G$-structures closed under isomorphisms and disjoint I-copies. To see that these are monadically suitable fragments, recall Lemma 5.10: by this Lemma, it is easy to see that for every free constants $\underline{c}$ of type $W \to \Omega$, for every $\mathcal{A} \in \mathcal{S}_G(\underline{c})$ and for every non empty set of indices I, we have that $\mathcal{A} \equiv_{\Phi_G(\underline{c})} \sum_I \mathcal{A}$. Thus taking disjoint I-copies is trivially a Φ_G-extensible operation.

Before giving the second family of examples of monadically suitable fragments, we introduce an alternative construction for proving extensibility of the operation of taking disjoint I-copies. This construction is nicely behaved only for fragments without identity and is called I-conglomeration:

Definition 5.18 *(I-conglomeration)*. Consider a first-order one-sorted relational signature $\mathcal{L}$ and a (non empty) index set I. The operation $\sum^I$, defined on $\mathcal{L}$-structures and called *I-conglomeration*, associates with a given $\mathcal{L}$-structure $\mathcal{M} = \langle [\![-]\!]_\mathcal{M}, \mathcal{I}_\mathcal{M} \rangle$ the $\mathcal{L}$-structure $\sum^I \mathcal{M}$ such that $[\![W]\!]_{\sum^I \mathcal{M}}$ is the disjoint union of I-copies of $[\![W]\!]_\mathcal{M}$ (here W is the unique sort of $\mathcal{L}$). The interpretation of relational constants is defined in such a way that we have

$$\sum^I \mathcal{M} \models P(d_1^{i_1}, \dots, d_n^{i_n}) \quad \Longleftrightarrow \quad \mathcal{M} \models P(d_1, \dots, d_n)$$

for every n-ary relational predicate P different from equality. ⊣

Notice that I-conglomerations and disjoint I-copies *coincide* for relational first-order signatures having only unary predicates. The preservation Lemma 5.10 can be reformulated as follows:

LEMMA 5.19. *Consider a first-order one-sorted relational signature $\mathcal{L}$ and the $\mathcal{L}$-structures $\mathcal{M}$ and $\sum^I \mathcal{M}$. The following statements hold:*

(i) *for every first-order formula $\varphi[x_1, \dots, x_n]$* not containing the equality predicate, *for every $d_1, \dots, d_n$ in the support of $\mathcal{M}$ and for every indices $i_1, \dots, i_n \in$*

[24] See Example 3.9 for this notation and for other similar notation used below.

I, we have that

$$\sum^{I} \mathcal{M} \models \varphi[d_1^{i_1}, \dots, d_n^{i_n}] \quad \Longleftrightarrow \quad \mathcal{M} \models \varphi[d_1, \dots, d_n];$$

(ii) *a first-order formula not containing the equality predicate is satisfiable in $\mathcal{M}$ iff it is satisfiable in $\sum^{I} \mathcal{M}$.*

Example 5.20. For a first-order relational one-sorted signature $\mathcal{L}_{2V}$, a *two variables i.a.f.* over $\mathcal{L}_{2V}$ is a fragment of the kind $\Phi_{2V} = \langle \mathcal{L}_{2V}, T_{2V}, \mathcal{S}_{2V} \rangle$, where: (i) T_{2V} contains the terms *without identity* belonging to the set $T_{NK}^{\mathcal{L}_{2V}}$ of Example 3.9 for $K = 1$ and $N = 2$; (ii) $\mathcal{S}_{2V}$ is a class of $\mathcal{L}_{2V}$-structures closed under isomorphisms and I-conglomerations. Notice that the closure properties in (ii) are guaranteed if $\mathcal{S}_{2V}$ is axiomatized by first-order formulae without identity.

For two monadically suitable i.a.f.'s Φ_1 and Φ_2 operating on disjoint signatures, let us call the combined fragment $\Phi_1 \oplus \Phi_2$ the *monadic fusion* of Φ_1 and Φ_2. For monadic fusions we have the following:

Theorem 5.21 (Decidability transfer for monadic fusions). *If the monadically suitable i.a.f.'s Φ_1, Φ_2 operating on disjoint signatures have decidable constraint satisfiability problems, so does their monadic fusion.*

Proof. Using Definition 5.16, we can say the following about the shared fragment $\Phi_0 = \langle \mathcal{L}_0, T_0, \mathcal{S}_0 \rangle$: (i) $\mathcal{L}_0$ is the empty signature $\mathcal{L}_\emptyset$; (ii) T_0 contains $T_{11}^{\mathcal{L}_\emptyset}$ and hence it includes the terms T_{M_0} of Example 3.10 relative to the empty modal signature O_{M_0}; (iii) for every tuple of free constants $\underline{c}_0$, the closed $\Phi_0(\underline{c}_0)$-terms $t[\underline{c}_0]$, modulo $\beta\eta$-equivalence, are a subset of the terms of the kind $\{x \mid \varphi[x]\}$, where $\varphi[x]$ is a monadic formula of first-order language, possibly with equality (that is, to build $\varphi[x]$, at most equality and the free constants $\underline{c}_0$ of type $W \to \Omega$ can be used); (iv) the structures in $\mathcal{S}_0(\underline{c}_0)$ are closed under disjoint I-copies and are $\Phi_0(\underline{c}_0)$-equivalent to their disjoint I-copies.

To justify (iv), argue as follows: if $\mathcal{A} \in \mathcal{S}_0(\underline{c}_0)$, then by Definition 4.1 it is the $\mathcal{L}_\emptyset(\underline{c}_0)$-reduct of a $\Phi_i(\underline{c}_0)$-structure $\mathcal{B}$ ($i = 1, 2$); since taking disjoint I-copies of $\mathcal{L}_\emptyset(\underline{c}_0)$-structures is $\Phi_i(\underline{c}_0)$-extensible by Definition 5.16(iii), we have that for every index set I, there is a $\Phi_i(\underline{c}_0)$-structure $\mathcal{B}'$ having $\sum_I \mathcal{A}$ as a $\mathcal{L}_\emptyset(\underline{c}_0)$-reduct and such that $\mathcal{B}$ is $\Phi_i(\underline{c}_0)$-equivalent to $\mathcal{B}'$. Taking $\mathcal{L}_\emptyset(\underline{c}_0)$-reducts, it follows that $\mathcal{A} \equiv_{\Phi_0(\underline{c}_0)} \sum_I \mathcal{A}$.

Using (ii) and (iv) above, we can repeat word-by-word the proof of Proposition 5.12 and in order to apply Theorem 5.6 we only have to show that Φ_0 is effectively locally finite. Despite the fact that there are infinitely many non equivalent monadic first-order sentences with equality, by using (iii)-(iv) we show that there are only finitely many closed $\Phi_0(\underline{c}_0)$-terms $t[\underline{c}_0]$ which are differently interpreted in structures from $\mathcal{S}_0(\underline{c}_0)$ (here $\underline{c}_0 := \{P_1, \dots, P_n\}$ are free constants, which must be of type $W \to \Omega$, because this is the only type of Φ_0). Recall that $t[\underline{c}_0] \sim_{\beta\eta} \{x \mid \varphi[x]\}$, where $\varphi[x]$ is as in (iii) above.

By closure under disjoint I-copies and $\Phi_0(\underline{c}_0)$-equivalence to disjoint I-copies (see (iv)), we can limit ourselves to the consideration of at most 2^{2^n}-structures from $\mathcal{S}_0(\underline{c}_0)$: each of these structures is uniquely determined by the fact that the cardinal invariants[25] a_ε are either 0 or m in it (here m is an infinite, big enough,

[25]Here and below, we freely use notation from the proof of Proposition 5.12.

cardinal). Each of these at most 2^{2^n} structures $\mathcal{A}_S$ is characterized by a set S of formulae of the kind $\varepsilon(x)$, in the sense that we have $\mathcal{A}_S \models \psi_S$, where ψ_S is the one-variable monadic sentence $\bigwedge_{\varepsilon\in S} \exists x\, \varepsilon(x) \wedge \bigwedge_{\varepsilon\notin S} \neg\exists x\, \varepsilon(x)$ (notice also that for $S \neq S'$, we have $\mathcal{A}_S \not\models \psi_{S'}$).

We *claim* that quantifier elimination holds in $\mathcal{A}_S$, i.e. that for every first-order formula $\varphi[\underline{x}]$ (built up from equality and from the unary predicates P_j) one can effectively compute from S a formula $\theta_S[\underline{x}]$ such that $\mathcal{A}_S \models \forall \underline{x}(\varphi[\underline{x}] \leftrightarrow \theta_S[\underline{x}])$ holds and such that $\theta_S[\underline{x}]$ does not contain quantifiers. To show the claim, we use the fact that for every $\varepsilon \in S$, the set $\varepsilon(\mathcal{A}_S)$ is infinite; recall also that in order to eliminate quantifiers, it is sufficient to eliminate them from primitive formulae, i.e. from formulae of the kind $\exists y\, \chi(y, \underline{x})$, where $\chi(y, \underline{x})$ is a conjunction of literals. In our case, these literals can only be $y = x_i, y \neq x_i, P_j(y), \neg P_j(y)$ (of course, literals in which y does not occur are not relevant). Since equations $y = x_i$ causes the quantifier $\exists y$ to be removed by replacement, we can assume that our χ is equivalent to a conjunction of negative literals $y \neq x_i$ and of a Boolean combination of atomic formulae of the kind $P_j(y)$. The set defined by this Boolean combination in $\mathcal{A}_S$ is either infinite or empty, so within $\mathcal{A}_S$, the formula $\exists y\, \chi(y, \underline{x})$ is equivalent either to $\perp$ or to $\top$.

As a consequence of the above claim, in the case in which the tuple $\underline{x}$ reduces to a single variable x, $\theta_S[x]$ is a boolean combination of the atomic formulae $P_j(x)$. Thus, in all the structures that belongs to $\mathcal{S}_0(\underline{c}_0)$, the $\Phi_0(\underline{c}_0)$-atom

$$\{x \mid \varphi[x]\} \;=\; \{x \mid \bigvee_S (\psi_S \wedge \theta_S[x])\}$$

is true, yielding effective local finiteness of Φ_0 (because there are only finitely many possibilities for S). $\square$

Theorem 5.21 offers various combination possibilities, we can for example, combine a guarded and a two-variable fragment, thus getting a rather ‘hybrid’ combined fragment. However notice that: (a) the conditions for a fragment to be monadically suitable are rather strong (for instance, the two variable fragment with identity is not monadically suitable); (b) the notion of monadic fusion is a restricted form of combination, because only unary second order variables are available for replacement when forming formulae of the combined fragment.

The main ingredient of Theorem 5.21 (namely the notion of a monadically suitable fragment) needs the present paper settings to be defined, but it is somewhat implicit in the literature on monodic temporal fragments (see for instance statements like that of Theorem 11.21 in [Gabbay et al. 2003]). As already mentioned in Subsection 1.4, our Main Decidability Transfer Theorem 5.6 can be used to prove a decidability theorem for monodic temporal/modal fragments whose extensional component is a monadically suitable fragment (see [Ghilardi et al. 2005; Nicolini 2006] for details), thus extending relevant results from the literature (see again [Gabbay et al. 2003]).

ACKNOWLEDGMENTS

The authors thank Manfred Schmidt-Schauss, Silvio Ranise, and the two anonymous referees for comments on previous versions of this paper.

REFERENCES

ANDRÉKA, H., VAN BENTHEM, J., AND NEMETI, I. 1998. Modal languages and bounded fragments of predicate logics. *Journal of Philosophical Logic 27*, 217–274.

ANDREWS, P. B. 2001. Classical type theory. In *Handbook of Automated Reasoning*, A. Robinson and A. Voronkov, Eds. Vol. 2. Elsevier and MIT Press, Chapter 15, 966–1007.

ANDREWS, P. B. 2002. *An Introduction to Mathematical Logic and Type Theory: To Truth Through Proof*, second ed. Applied Logic Series, vol. 27. Kluwer Academic Publishers, Dordrecht (Holland).

BAADER, F., CALVANESE, D., MCGUINNESS, D. L., NARDI, D., AND PATEL-SCHNEIDER, P. F., Eds. 2003. *The Description Logic Handbook. Theory, Implementation, and Applications.* Cambridge University Press, Cambridge (UK).

BAADER, F. AND GHILARDI, S. 2005a. Connecting many-sorted structures and theories through adjoint functions. In *Proceedings of the 5th International Workshop on Frontiers of Combining Systems (FroCoS 2005).* Lecture Notes in Computer Science, vol. 3717. Springer, Wien (Austria), 31–47.

BAADER, F. AND GHILARDI, S. 2005b. Connecting many-sorted theories. In *Proceedings of the 20th International Conference on Automated Deduction (CADE 2005).* Lecture Notes in Computer Science, vol. 3632. Springer, Tallinn (Estonia), 278–294.

BAADER, F., GHILARDI, S., AND TINELLI, C. 2006. A new combination procedure for the word problem that generalizes fusion decidability results in modal logics. *Information and Computation.* (To appear).

BAADER, F. AND HANSCHKE, P. 1991. A scheme for integrating concrete domains into concept languages. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence (IJCAI 1991).* Vol. 1. Morgan Kaufman, Sydney (Australia), 452–457.

BAADER, F., LUTZ, C., STURM, H., AND WOLTER, F. 2002. Fusions of description logics and abstract description systems. *Journal of Artificial Intelligence Research 16*, 1–58.

BAADER, F. AND NIPKOW, T. 1998. *Term Rewriting and All That.* Cambridge University Press, Cambridge (UK).

BAADER, F. AND TINELLI, C. 2002. Deciding the word problem in the union of equational theories. *Information and Computation 178,* 2, 346–390.

BAUMGARTNER, P., FURBACH, U., AND PETERMANN, U. 1992. A unified approach to theory reasoning. Research Report 15–92, Universität Koblenz-Landau, Koblenz (Germany). Fachberichte Informatik.

BONACINA, M. P., GHILARDI, S., NICOLINI, E., RANISE, S., AND ZUCCHELLI, D. 2006. Decidability and undecidability results for Nelson-Oppen and rewrite-based decision procedures. In *Proceedings of the 3rd International Joint Conference on Automated Reasoning (IJCAR 2006).* Lecture Notes in Computer Science, vol. 4130. Springer, Seattle (WA, USA), 513–527.

BÖRGER, E., GRÄDEL, E., AND GUREVICH, Y. 2001. *The Classical Decision Problem.* Universitext. Springer, Berlin (Germany). Reprint of the 1997 original.

BOZZANO, M., BRUTTOMESSO, R., CIMATTI, A., JUNTTILA, T., VAN ROSSUM, P., RANISE, S., AND SEBASTIANI, R. 2005. Efficient theory combination via boolean search. *Information and Computation.* (To appear).

CHANG, C.-C. AND KEISLER, H. J. 1990. *Model Theory*, third ed. North-Holland, Amsterdam (Holland).

CHENADEC, P. L. 1986. *Canonical Forms in Finitely Presented Algebras.* Research Notes in Theoretical Computer Science. Pitman Publishing Ltd., London (UK).

COX, D., LITTLE, J., AND O'SHEA, D. 1997. *Ideals, Varieties, and Algorithms*, second ed. Undergraduate Texts in Mathematics. Springer-Verlag, New York (NY, USA).

DOWEK, G. 2001. Higher order unification and matching. In *Handbook of Automated Reasoning*, A. Robinson and A. Voronkov, Eds. Vol. 2. Elsevier and MIT Press, Chapter 16, 1009–1062.

FERMÜLLER, C., LEITSCH, A., TAMMET, T., AND ZAMOV, N. 1993. *Resolution Methods for the Decision Problem.* Lecture Notes in Computer Science, vol. 679. Springer-Verlag, Berlin (Germany).

Fiorentini, C. and Ghilardi, S. 2003. Combining word problems through rewriting in categories with products. *Theoretical Computer Science 294*, 103–149.

Friedman, H. 1975. Equality between functionals. In *Logic Colloquium*. Lecture Notes in Mathemathics, vol. 453. Springer-Verlag, Boston (MA, USA), 22–37.

Gabbay, D. M., Kurucz, A., Wolter, F., and Zakharyaschev, M. 2003. *Many-Dimensional Modal Logics: Theory and Applications*. Studies in Logic and the Foundations of Mathematics, vol. 148. North-Holland Publishing Co., Amsterdam (Holland).

Ganzinger, H. and de Nivelle, H. 1999. A superposition decision procedure for the guarded fragment with equality. In *14th Symposium on Logic in Computer Science (LICS 1999)*. IEEE Computer Society Press, Trento (Italy), 295–303.

Ganzinger, H., Ruess, H., and Shankar, N. 2005. Modularity and refinement in inference systems. In *Notes of the 3rd Workshop on Pragmatics of Decision Procedures in Automated Reasoning (PDPAR 2005)*. Edinburgh (Scotland).

Ghilardi, S. 2004. Model theoretic methods in combined constraint satisfiability. *Journal of Automated Reasoning 33,* 3-4, 221–249.

Ghilardi, S., Nicolini, E., and Zucchelli, D. 2005. A comprehensive framework for combined decision procedures. Rapporto Interno DSI 304-05, Università degli Studi di Milano, Milano (Italy). Available at `http://homes.dsi.unimi.it/~zucchell/publications/techreport/GhiNiZu-RI304-05.pdf`.

Ghilardi, S. and Santocanale, L. 2003. Algebraic and model theoretic techniques for fusion decidability in modal logics. In *Proceedings of the 10th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR 2003)*. Lecture Notes in Computer Science, vol. 2850. Springer, Almaty (Kazakhstan), 152–166.

Grädel, E. 1999. Decision procedures for guarded logics. In *Proceedings of 16th International Conference on Automated Deduction (CADE 1999)*. Lecture Notes in Computer Science, vol. 1632. Springer-Verlag, Trento (Italy), 31–51.

Hirsch, R. and Hodkinson, I. 2002. *Relation Algebras by Games*. Studies in Logic and the Foundations of Mathematics, vol. 147. North-Holland Publishing Co., Amsterdam (Holland).

Kozen, D. 1983. Results on the propositional μ-calculus. *Theoretical Computer Science 27,* 3, 333–354.

Kutz, O., Lutz, C., Wolter, F., and Zakharyaschev, M. 2004. $\mathcal{E}$-connections of abstract description systems. *Artificial Intelligence 156*, 1–73.

Lambek, J. and Scott, P. J. 1988. *Introduction to higher order categorical logic*. Cambridge Studies in Advanced Mathematics, vol. 7. Cambridge University Press, Cambridge (UK). Reprint of the 1986 original.

MacLane, S. and Birckhoff, G. 1988. *Algebra*, third ed. Chelsea Publishing Co., New York (NY, USA).

Marx, M. 2001. Tolerance logic. *Journal of Logic, Language and Information 10*, 353–374.

Nelson, G. and Oppen, D. C. 1979. Simplification by cooperating decision procedures. *ACM Transaction on Programming Languages and Systems 1,* 2, 245–257.

Nicolini, E. 2006. Combined decision procedures for constraint satisfiability. Ph.D. thesis, Università degli Studi di Milano.

Nieuwenhuis, R., Oliveras, A., and Tinelli, C. 2005. Abstract DPLL and abstract DPLL modulo theories. In *Proceedings of the 11th International Conference on Logic for Programming, Artificial Intelligence and Reasoning (LPAR 2004)*. Lecture Notes in Computer Science, vol. 3452. Springer, Montevideo (Uruguay), 36–50.

Odifreddi, P. 1989. *Classical Recursion Theory*. Studies in Logic and the Foundations of Mathematics, vol. 125. North-Holland Publishing Co., Amsterdam (Holland).

Oppen, D. C. 1980. Complexity, convexity and combinations of theories. *Theoretical Computer Science 12*, 291–302.

Rabin, M. O. 1969. Decidability of second-order theories and automata on infinite trees. *Transactions of the American Mathematical Society 141*, 1–35.

Tinelli, C. 2002. A DPLL-based calculus for ground satisfiability modulo theories. In *Proceedings of the 8th European Conference on Logics in Artificial Intelligence (JELIA 2002)*. Lecture Notes in Computer Science, vol. 2424. Springer, Cosenza (Italy), 308–319.

Tinelli, C. 2003. Cooperation of background reasoners in theoryreasoning by residue sharing. *Journal of Automated Reasoning 3,* 1, 1–31.

Tinelli, C. and Harandi, M. T. 1996. A new correctness proof of the Nelson-Oppen combination procedure. In *Proceedings of the 1st International Workshop on Frontiers of Combining Systems (FroCoS 1996)*. Applied Logic, vol. 3. Kluwer Academic Publishers, Munich (Germany), 103–119.

van Benthem, J. 1985. *Modal Logic and Classical Logic.* Monographs in Philosophical Logic and Formal Linguistics. Bibliopolis, Naples (Italy).

Wolter, F. 1998. Fusions of modal logics revisited. In *Advances in Modal Logic*, M. Kracht, M. de Rijke, H. Wansing, and M. Zakharyaschev, Eds. CSLI Lecture Notes, vol. 87. CSLI Publ., Stanford (CA, USA), 361–379.

Wolter, F. and Zakharyaschev, M. 2001. Decidable fragments of first-order modal logics. *Journal of Symbolic Logic 66*, 1415–1438.

Received January 2006; revised June 2006; accepted July 2006