A SHORT OVERVIEW OF HIDDEN LOGIC

ISABEL FERREIRIM AND MANUEL A. MARTINS

Abstract. In this paper we review a hidden (sorted) generalization of k-deductive systems - *hidden k-logics*. They encompass deductive systems as well as hidden equational logics and inequational logics. The special case of hidden equational logics has been used to specify and to verify properties in program development of behavioral systems within the dichotomy visible vs. hidden data. We recall one of the main applications of this work - the study of behavioral equivalence. Related results are obtained through combinatorial properties of the Leibniz congruence relation.

In addition we obtain a few new developments concerning hidden equational logic, namely we present a new characterization of the behavioral consequences of a theory.

1. Introduction

This paper is intended in part as a survey of results which have been developed in a series of papers in a broader context - the hidden k-logics. These logics are a generalization of k-deductive systems and their study originated in a series of lectures on Abstract Algebraic Logic - Application to Computer Science, during Don Pigozzi's visit to CAUL, Lisbon, in 1999 ([45]).

When we refer to a deductive system we usually mean a 1-dimensional deductive system such as the deductive system of the classical, the intuitionistic and modal propositional calculi. The notion of deductive system, as an abstract consequence operator, is due to Tarski (see [51]). In their groundbreaking work [7], Blok and Pigozzi introduced and studied the notion of Leibniz congruence in the context of 1-deductive systems. These systems have been studied for years by several logicians, for example Blok and Pigozzi [8], Czelakowski [12] and [13], the Barcelona group led by Font and Jansana [20], and also Hermann [30], Pigozzi [46] and Wójcicki [55].

The higher dimensional systems, called *k-deductive systems*, constitute a natural generalization including many logical systems, e.g., equational and inequational

Date: March 24, 2018.

Accepted authors' manuscript published as: Ferreirim I., Martins M.A. (2018) A short overview of Hidden Logic. In: Czelakowski J. (eds) Don Pigozzi on Abstract Algebraic Logic, Universal Algebra, and Computer Science. Outstanding Contributions to Logic, vol 16, pp. 167-201, Springer, Cham [DOI:10.1007/978-3-319-74772-9_6]. The final publication is available at Springer via https://link.springer.com/chapter/10.1007/978-3-319-74772-9_6.

Acknowledgments. This work was funded by ERDF - European Regional Development Fund through the COMPETE Programme (operational programme for competitiveness) and by National Funds through the FCT (Portuguese Foundation for Science and Technology) within project UID/MAT/04106/2013 at CIDMA. The second author also acknowledges the financial assistance by the EU FP7 Marie Curie PIRSES-GA-2012-318986 project GeTFun: Generalizing Truth-Functionality.

logics. k-deductive systems were introduced by Blok and Pigozzi in [9]; other important references are [15] and [43]. For these higher dimension deductive systems Blok and Pigozzi developed a theory similar to the theory of 1-deductive systems. The notion of Leibniz congruence still plays the central role in the study of k-deductive systems. There is another generalization of k-deductive systems, called K-deductive systems, that includes Gentzen systems. This theory was developed by Pałasińska and presented in her PhD thesis [43].

We also should call attention to the work by Voutsadakis on the study of deductive systems within category theory. Voutsadakis has done extensive work on categorical abstract algebraic logic of π-institutions ([52]).

Abstract algebraic logic (AAL) is an area of algebraic logic that focuses on the study of the relationship between logical equivalence and logical truth. Moreover, AAL is centered on the process of associating a class of algebras to a logical system. This approach contrasts with the usual treatment given in algebraic logic where the emphasis is on the study of the class of algebras obtained by this process. A logical system, a *deductive system* as it has been called, is taken to be a pair formed by a signature Σ and a substitution-invariant closure relation on the set of terms over Σ in a countably infinite fixed set of variables X, $\mathrm{Te}_\Sigma(X)$ (we will use the word 'formula' as a synonym for 'term'). By a *closure relation* on $\mathrm{Te}_\Sigma(X)$ we mean a binary relation $\vdash$, where $\vdash\ \subseteq \mathcal{P}(\mathrm{Te}_\Sigma(X)) \times \mathrm{Te}_\Sigma(X)$, between subsets of terms and individual terms satisfying the following conditions: (1) $\Gamma \vdash \gamma$ for each $\gamma \in \Gamma$ and (2) $\Gamma \vdash \varphi$ and $\Delta \vdash \gamma$ for each $\gamma \in \Gamma$ imply $\Delta \vdash \varphi$. The relation $\vdash$ is said to be *substitution-invariant* if $\Gamma \vdash \varphi$ implies $\sigma(\Gamma) \vdash \sigma(\varphi)$ for every substitution $\sigma : X \to \mathrm{Te}_\Sigma(X)$. Moreover, $\vdash$ is said to be *finitary* if $\Gamma \vdash \varphi$ implies $\Delta \vdash \varphi$ for some finite subset Δ of Γ.

The main paradigm in AAL is the representation of the classical propositional calculus in the equational theory of Boolean algebras by means of the *Lindenbaum-Tarski process.* In its traditional form, the Lindenbaum-Tarski process relies on the fact that the classical propositional calculus has a biconditional "$\leftrightarrow$" that defines logical equivalence. The set of all formulas is partitioned into logical equivalence classes and then abstracted by the familiar algebraic process of forming the quotient algebra. This algebra is called the *Lindenbaum-Tarski algebra.* There are many deductive systems that do not have a biconditional, and hence the Lindenbaum-Tarski process cannot be applied directly. However, there is an abstract notion of logical equivalence in every deductive system called the *Leibniz congruence.* In this way the Lindenbaum-Tarski process can be generalized so as to apply to many deductive systems.

The Leibniz congruence $\Omega(T)$ on the term algebra over a theory T is characterized in the following way: for any pair of terms α, β, $\alpha \equiv \beta\ (\Omega(T))$ if for every term φ and any variable p occurring in φ, $\varphi(p/\alpha) \in T$ if and only if $\varphi(p/\beta) \in T$. The Leibniz congruence is extended in a natural way to the power set of an arbitrary algebra. Given a Σ-algebra A and a designated subset F of A, the pair $\langle \mathbf{A}, F\rangle$ is called a *matrix.* The relation $\Omega(F)$ identifies any two elements which cannot be distinguished by any property defined by a formula. More precisely, for any pair of elements a, b of $\mathbf{A}$, $a \equiv b\ (\Omega(F))$ if for each formula $\varphi(x, u_0, \cdots, u_{k-1})$, and all parameters $\bar{c} \in A^k$, $\varphi^{\mathbf{A}}(a, \bar{c}) \in F$ if and only if $\varphi^{\mathbf{A}}(b, \bar{c}) \in F$. Moreover, $\Omega(F)$ is a congruence on $\mathbf{A}$. A matrix $\langle \mathbf{A}, F\rangle$ is said to be *reduced* if $\Omega(F)$ is the identity relation. The congruence $\Omega(F)$ is called the Leibniz congruence since it may be seen as the sentential version of the second order definition of equality given by Leibniz. He defined two objects to be equal if they have exactly the same properties. In the model of a world given by the matrix $\langle \mathbf{A}, F\rangle$, a property is determined by a formula $\varphi(x, \bar{u})$ and parameters $\bar{c} \in A^k$. Thus two elements are equal, in the Leibniz sense, if the condition above holds.

Equational logic serves as the underlying logic in many formal approaches to program specification. The algebraic data types specified in this formal way may be viewed as abstract machines on which the programs are to be run. This is one way of giving a precise algebraic semantics for programs, against which the correctness of a program can be tested. Equational logic can be seen as a 2-deductive system and then the tools and results of AAL can be applied to it.

However, object oriented (OO) programs present a special challenge for equational methods. This is due to properties inherent to the OO programs. A more appropriate model for the abstract machine in the case of an OO program is, arguably, a state transition system: as in the case of a state of such a system, a state of an OO program can be viewed as encapsulating all pertinent information about the abstract machine when it reaches the state during execution of the program. As a way of meeting the aforementioned challenge the standard equality predicate can be augmented by *behavioral equivalence*; in this way many of the characteristic properties of state transition systems can be grafted onto equational logic.

Two terms are said to be behaviorally equivalent if and only if they cannot be distinguished by any visible context. This is the primitive notion of behavioral equivalence due to Reichel ([48]). The idea of looking at the satisfaction relation of hidden terms as behavioral equivalence was also introduced by Reichel in the 80's [48] and it seems to be the correct way of interpreting equality between hidden terms. Since then, it has been adopted and generalized by many people. The most significant contributions have been given by Goguen and Malcolm [25], Bidoit and Hennicker [4] and their coworkers.

Generalizations of the notion of behavioral equivalence have been considered in the literature. Goguen et al. consider Γ-*behavioral equivalence*, where Γ is a subset of the set of all operation symbols in the signature (see, e.g., [26]). Γ-behavioral equivalence is defined in a manner analogous to ordinary behavioral equivalence, but making use only of the contexts built from the operation symbols in Γ. It can be proved that the Γ-behavioral equivalence is the largest Γ-congruence with the identity as the visible part. Thus, coinduction methods, based on this fact, may still be formulated for this more general notion. Other interesting questions concerning Γ-behavioral equivalence may arise, such as the study of the compatibility of some operation symbols outside of Γ with respect to Γ-behavioral equivalence. This problem has been studied by Diaconescu and Futatsugui [19] and Bidoit and Hennicker [5].

On the other hand, Bidoit and Hennicker [4] generalize this notion by endowing the hidden algebras with a binary relation that may be partial. As a particular case we can apply their algebraic approach to the behavioral setting by considering algebras endowed with the Γ-behavioral equivalence.

One important feature of behavioral equivalence in computer science is that it is the largest congruence that is the identity on the visible part. This is, in some way, similar to the property of the Leibniz congruence being the largest congruence compatible with the filter. To apply AAL to the theory of the specification of abstract data types, we have to view specification logic as a deductive system (i.e., as a substitution-invariant closure relation on an appropriate set of formulas) and behavioral equivalence as a generalized notion of Leibniz congruence. The class of deductive systems has to be expanded so as to include multisorted as well as one-sorted systems. The notion of Leibniz congruence has to be considered in the context of the dichotomy of visible vs. hidden; namely, the formulas used in the characterization of the Leibniz congruence also have to be restricted to an appropriate proper subset of all formulas, namely the visible formulas, which are called *contexts*. Therefore, the notion of k-deductive systems has to be generalized by considering the data to be heterogeneous in the sense that the data elements may be of different kinds. Specifically, there are the basic data, like integers, reals and Boolean, whose properties are well-known and for which well-defined and easily manipulated representations are available; and there are the auxiliary data such as arrays, lists, stacks, whose properties are specified by their behavior under the programs with visible output, and hence ultimately in terms of the basic data. Thus, we use distinct representations for each kind of data elements.

This leads to the notion of hidden k-logics. They are a natural generalization of k-deductive systems. They encompass deductive systems as well as equational logics and inequational logics and their respective hidden versions. Hidden k-logics are used to specify systems whose data may be heterogeneous, i.e., split in different kinds, usually called sorts. Moreover, in hidden k-logics we are also able to distinguish internal data (*hidden data*) and the real data (*visible data*). This advantage is central in the specification of OO systems.

Hidden k-logics, as a natural generalization of k-deductive systems, were introduced by Martins and Pigozzi in [39]. Preliminary work on applications of AAL to the specification of abstract data types had been discussed in Lisbon, CAUL, in a series of lectures given by Don Pigozzi in 1999. The theory was then developed in [35], where improvements concerning specification and verification of programs were established using tools from *abstract algebraic logic* (AAL). A generalization of the AAL theory to the hidden setting has been successfully explored and several applications to the OO paradigm have been developed using AAL methods (cf. [35, 36, 37, 38, 39]). This generalization is not straightforward. The multisort aspect is present for example in the following: in the one-sorted case one can show that a hidden k-logic is *protoalgebraic* (an important semantic property) if and only if it admits a protoequivalence system without parameters; however, in the broader context of multi-sorted logics, a generic protoequivalence system contains parameters (cf. [37]). This new bridge between AAL and the specification and verification theory of software systems has yet to be further developed. On the other hand, it must be mentioned that behavioral specification theory has also influenced the development of AAL, namely the recent theory of behavioral algebraization of logics (cf. [11] and [53]).

For the purposes of this paper, it is useful to define a hidden k-logic as an abstract closure relation on the set of k-formulas. That is, a *hidden k-logic* is a pair $\mathcal{L} = \langle \Sigma, \vdash_{\mathcal{L}} \rangle$, where Σ is a hidden signature and $\vdash_{\mathcal{L}}$ is a substitution-invariant closure relation on the set of visible k-formulas, called *the consequence relation of* $\mathcal{L}$. This consequence relation may be finitary or not. It is finitary just in case it admits a presentation by axioms and inference rules, in the usual Hilbert style. In this case, $\vdash_{\mathcal{L}}$ is said to be *specifiable*. An *$\mathcal{L}$-theory* of a hidden k-logic $\mathcal{L}$ is a set of visible k-formulas that are closed under the consequence relation $\vdash_{\mathcal{L}}$. The set of all $\mathcal{L}$-theories is denoted by $\mathrm{Th}(\mathcal{L})$.

Hidden k-logics are useful mainly because they encompass not only the 2-dimensional hidden and standard equational logics, but also *Boolean logics*; these are 1-dimensional multisorted logics with Boolean as the only visible sort, and with equality-test operations for some of the hidden sorts in place of equality predicates. They also include all assertional logics, the purview of AAL. In this way we obtain a unified theory for a variety of logical systems. We give special attention to a special hidden 2-logic, the *hidden equational logic*. In the hidden equational case we only consider a primitive notion of equality between visible data. It is defined as a sorted equational logic, using reflexivity, symmetry, transitivity and congruence rules, but only on the visible part. The expression "hidden equational logic" comes from the fact that the equality predicate is restricted so as to apply only to visible data elements. There is no primitive notion of equality for hidden data elements in the logic.

There is an important assumption about the syntax of hidden k-logics, as we define them, that arises from the fact that they are intended to serve as the underlying logic in the specification of object oriented systems. The assumption is that the specification can use only visible axioms since we only have access to the internal behavior by programs with visible output, i.e., the equality between two hidden data elements of the same sort is not specifiable by abstract equality axioms as in the standard equational logic. This assumption follows the work of Leavens and Pigozzi [32, 33]. The restriction to axioms of visible type is natural from the perspective of operational semantics. That is, in operational terms, one views the axioms as specifying the output of programs, which indirectly determine the behavior of the hidden data objects the programs manipulate. Hence, only the visible part of the system is specified. This does not follow some approaches in the area (see, e.g., [26] and [50]) but it does not restrict the power of specifications in practice. On the contrary, it endows the underlying theory with even richer modes of specification. Indeed, we may also specify internal properties of the system after checking that they do not produce unexpected behavioral changes on the system, i.e., that by adding those properties to the specification we do not obtain a different set of behavioral consequences. In the hidden equational case, we show that if a conditional equation is behaviorally valid, then it may be added as a new axiom without any undesired consequences ([39]).

The semantics of hidden k-logics reflects all the special features of these logics that have been discussed. A *k-data structure* is a collection of data items of different sorts, such as lists, Booleans, numbers, and operations involving them, together with a set of k-tuples of elements, called a *filter*, that serves as a set of generalized truth values (the term filter comes from the one-sorted case in AAL). Moreover, the universe, which is a sorted algebra, is split into two disjoint sets, namely the *hidden* part, which corresponds to the states of a state transition system, and the *visible* part. There are also two different interpretations of the operation symbols: the *attributes* return visible data and are used to observe the state of the system while the *methods* may change the state. As in AAL, the main object is to understand and clarify the relationship between logical truth and logical equivalence, which for hidden k-logics correspond, respectively, to the visible properties of states, as specified by the axioms of the logic, and their behavioral equivalence.

Since the consequence relation $\vdash$ is a closure relation on the set of visible k-formulas, the filter consists exclusively of k-tuples of visible elements of the k-data structure. The designated filter F of a k-data structure $\mathcal{A} = \langle \mathbf{A}, F\rangle$ is considered as the set of "truth values" in $\mathcal{A}$. Thus, we say that $\mathcal{A} = \langle \mathbf{A}, F\rangle$ is a *model of* a hidden k-logic $\mathcal{L}$ if every consequence $\Gamma \vdash \varphi$ of $\mathcal{L}$ is a semantic consequence of $\mathcal{A}$, in the sense that for every assignment $h : X \to A$, $h(\varphi) \in F$ whenever $h(\Gamma) \subseteq F$. In this case, we say that F is an *$\mathcal{L}$-filter.* The $\mathcal{L}$-filters on the term algebra are the theories of $\mathcal{L}$ and consequently $\langle \mathrm{Te}_\Sigma(X), T\rangle$, with T a theory of $\mathcal{L}$, is always a model of $\mathcal{L}$.

1.1. **Related work.** Many computer scientists have studied behavioral equivalence for the last 20 years. Here we present some approaches which are important to contextualize our work.

1.1.1. *Hidden algebras.* Hidden algebras were introduced by Goguen in [22] and further developed in [25, 26, 50], in order to generalize many-sorted algebras and give an algebraic semantics for the object oriented paradigm.

When they first appeared, hidden algebras were considered over restricted signatures. These were assumed to have the visible part fixed, in the sense that all sorted algebras over it have the same visible part. Usually, this visible part was a standard algebra such as the natural numbers or the two-element Boolean algebra. This is called *fixed-data semantics.* Another restriction which is sometimes assumed in order to apply coalgebraic methods and results to the study of behavioral equivalence is the requirement that the methods and the attributes must have exactly one hidden argument. In this case it is called *monadic semantics.*

The behavioral aspects of modern software make hidden algebras more suitable than standard algebras for abstract machine implementations. Consequently, there has been an increasing development in this field. In the last fifteen years the theory on hidden algebras has been further developed and applied to more general settings, first by Goguen and Malcolm [25] and more recently by his former collaborators [27, 41]. Currently, almost all of the results may be established for *polyadic loose-data semantics.* Polyadic loose-data semantics allows any kind of operation symbols. Furthermore, in order to have more freedom to choose an adequate implementation, the visible part of the algebras is no longer fixed: it may be any sorted algebra in which the requirements (axioms) of the given specification are valid. Moreover, some authors are interested in applying coalgebraic methods, and then they have to restrict their signatures to the monadic fixed-data semantics. Malcolm [34] has shown that behavioral equivalence may be formulated in the context of coalgebra (see also [49]).

1.1.2. *Behavioral equivalence and behavioral validity.* Two terms are said to be behaviorally equivalent if and only if they cannot be distinguished by any visible context. This is the primitive notion of behavioral equivalence due to Reichel ([48]).

The idea of looking at the satisfaction relation of hidden terms as behavioral equivalence was also introduced by Reichel in the 80's [48] and it seems to be the correct way of interpreting equality between hidden terms. Since then, it has been adopted and generalized by many authors. The most significant contributions have been given by Goguen, Bidoit, Bouhoula and their co-authors (e.g., [24, 4, 10]).

Generalizations of the notion of behavioral equivalence have also been considered in the literature. Goguen and Roşu [26] introduced and studied Γ-behavioral equivalence, where Γ is a subset of the set of all operation symbols in the signature. Γ-behavioral equivalence is defined analogously to ordinary behavioral equivalence, but making use only of the contexts built from the operation symbols in Γ. It can be proved that the Γ-behavioral equivalence is the largest Γ-congruence with the identity as the visible part. Thus, coinduction methods, based on this fact, may still be formulated for this more general notion. Other interesting questions concerning Γ-behavioral equivalence may arise, such as the study of the compatibility of some operation symbols outside of Γ with respect to Γ-behavioral equivalence. This problem has been studied by Diaconescu and Futatsugi [19] and Bidoit and Hennicker [5].

On the other hand, Bidoit et al. [6] generalize this notion by endowing the hidden algebras with a binary relation. As a particular case we can apply their algebraic approach to the behavioral setting by considering the algebras together with the Γ-behavioral equivalence.

1.1.3. *Hidden logics.* Many behavioral logics have been defined and studied in the literature. The most relevant versions are *hidden logic* [24, 25] and *observational logic* [29, 4]. There is also another observational logic due to Padawitz ([42]), called *swinging types logic*, but it is similar to the observational logic of Bidoit and his coworkers (see `http://ls5-www.cs.uni-dort mund.de/~peter/ Swinging.html` for more details).

Hidden logic is a variant of the equational logic in which some part of the specification is visible and another is hidden. The formulas are just equations and the satisfaction relation is taken behaviorally.

Observational logic is different from hidden logic but both are based on behavioral equivalence, which means indistinguishability under contexts. Observational logic was introduced by Bidoit and Hennicker (see [4], [29] and [28]) to formalize behavioral validity (correctness). Tarski's satisfaction relation of first-order formulas (with equality) is considered as a "behavioral satisfaction relation" which is determined, in a natural way, by the family of congruence relations (possibly partial) with which each algebra is endowed. This relation is called *behavioral equality.* The behavioral satisfaction relation is just defined by considering the equality symbol interpreted as the behavioral equality. First-order theories are generalized to the so-called *behavioral theories* where the equality symbol is interpreted as the behavioral equality. In [4] Bidoit and Hennicker develop a method for proving behavioral theorems whenever an axiomatization of the behavioral equality is provided. This is based on reducing behavioral satisfaction to ordinary satisfaction. Consequently any proof system for first-order logic can be used to prove the behavioral validity, with respect to a given behavioral equality, of first-order formulas.

1.1.4. *Automatic methods for behavioral reasoning in hidden logics.* As far as we know, the languages that support automated behavioral reasoning are Spike [3], CafeOBJ [18] and BOBJ [31].

In [10], Bouhoula and Rusinowitch set forth an automatic method for proving behavioral validity of conditional equations in conditional specifications. They use the fact that there are specifications for which a smaller set of contexts is enough to know what the outputs of the remaining ones are. They call them *critical contexts.* The work of Bouhoula and Rusinowitch was the genesis of the SPIKE language which is based on context induction.

The CafeOBJ language was developed by Diaconescu and Futatsugi [18]. It implements behavioral rewriting to make behaviorally sound reductions of terms. It is based on a behavioral version of the well known efficient method of rewriting for automated theorem proving (see `http://www.ldl.jaist.ac.jp/Research/CafeOBJ/`).

Goguen et al. have been developing algorithms for automating behavioral reasoning based on their techniques of coinduction and have been making use of cobases. Coinduction in its pure form requires human intervention in the choice of the cobasis. A cobasis is just a set of operation symbols that generates a relation on the set of terms which is a subset of the behavioral equivalence. A good choice of a cobasis can simplify the proof enormously. Those algorithms have been improved in order to be applied to more general situations and have been implemented in the BOBJ language. In [31] Goguen et al. presented a new technique which combines behavioral rewriting and coinduction. The most recent version is CCCRW, called *conditional circular coinductive rewriting with case analysis*. The authors claim that it is in fact the most powerful automated proof technique available at present [23] (see also [27, 41]). Besides the fact that this new algorithm uses conditional circular coinductive rewriting to prove behavioral validity, it also allows for case analysis (see `http://www-cse.ucsd.edu/groups/tatami/bobj`). This theoretical result supports the automated behavioral prover **Circ** based on the circularity principle (`http://fsl.cs.illinois.edu/index.php/Circ`), which generalizes both circular coinduction and structural induction.

1.1.5. *AAL approach to behavioral equivalence.* As mentioned earlier, Pigozzi gave a series of lectures on the application of AAL to Computer Science. These lectures marked the starting point of Martins' investigations on the algebraic theory of hidden k-logics, which led to his PhD thesis [35]. In an introductory paper with Pigozzi [39] the instantiation to hidden equational logic was studied in depth. Closure properties of the class of behavioral models (reduced models) are studied in [37]. Refinement and institutions for behavioral logics in the context of our approach were discussed in [36]. A natural generalization of the Nerode equivalence of finite automata to k-data structures concerning this general notion of behavioral equivalence of a k-data structure can be found in [38]. Recently, a deduction-detachment theorem for hidden k-logics was presented in [1] and the behavioral equivalence between hidden k-logics is investigated in [2] by Babenyshev and Martins.

Outline of the paper. This paper is organized in two main sections. In section 2 we give an overview of basic concepts and recent results pertaining to hidden k-logic. The Leibniz congruence is one of the tools developed in this context. As in standard AAL, it plays a crucial role in the theory. Section 3 is devoted to behavioral equivalence. Theorem 3.3 shows the adequacy of this approach: the behavioral equivalence is, in fact, the (generalized) Leibniz congruence. In Sections 3.1 and 3.2 we present new characterizations for an equation to be a behavioral consequence of a theory of a HEL. In Section 3.2 we get a simpler characterization for the case of strict equational HEL's.

2. Hidden k-logic

A *hidden* (*sorted*) *signature* is a triple $\Sigma = \big\langle \mathrm{SORT}, \mathrm{VIS}, \langle \mathrm{OP}_\tau \mid \tau \in \mathrm{TYPE}\rangle \big\rangle$, where: SORT is a nonempty, countable set whose elements are called *sorts*; VIS is a subset of SORT, called the set of *visible sorts*; TYPE is a set of nonempty sequences $S_0, \dots, S_n$ of sorts, called *types* and usually written as $S_0, \dots, S_{n-1} \to S_n$; and, for each $\tau \in \mathrm{TYPE}$, OP_τ is a countable set; the elements of OP_τ are called *operation symbols* of type τ. Operation symbols of type $\to S$ are said to be *constants*. We will denote $\langle \mathrm{OP}_\tau \mid \tau \in \mathrm{TYPE} \rangle$ by OP.

The sorts in SORT \ VIS, that are not visible, are called *hidden sorts*. The set of hidden sorts is denoted by HID. For simplicity, we require the sets of operation symbols to be pairwise disjoint in order to avoid overloading of names (i.e., for any distinct $\tau, \tau' \in \mathrm{TYPE}$, $\mathrm{OP}_\tau \cap \mathrm{OP}_{\tau'} = \varnothing$).

From each hidden signature Σ we obtain the associated *un-hidden* signature Σ^{UH} by making all sorts of Σ visible.

A Σ*-algebra* is a pair $\big\langle A, \langle O^A \mid \tau \in \mathrm{TYPE}, O \in \mathrm{OP}_\tau \rangle \big\rangle$, where A is a SORT-sorted set, such that $A_S \neq \varnothing$, for all $S \in \mathrm{SORT}$, and for any $\tau \in \mathrm{TYPE}$ and $O \in \mathrm{OP}_\tau$, O^A is an operation on A of type τ (i.e., if $\tau = S_0, \dots, S_{n-1} \to S_n$ then $O^A : A_{S_0} \times \dots \times A_{S_{n-1}} \to A_{S_n}$). As usual, we use the same symbol to denote an algebra and the carrier of the algebra.

We assume for carrier sets A of data structures that $A_S \neq \varnothing$ for all $S \in \mathrm{SORT}$, a condition similar to one used to define regular universal algebras. With this assumption we exclude some data structures of practical interest. However, the mathematics is simpler in this case and most results of universal algebra hold in their usual form.

A (*sorted*) *congruence on* a Σ-algebra A is a sorted binary relation $\Theta \subseteq A^2$ such that: (i) for each $S \in \mathrm{SORT}$, Θ_S is an equivalence relation on A_S and (ii) Θ satisfies the congruence condition: for every operation symbol $O \in \mathrm{OP}_\tau$ with $\tau = S_0, \dots, S_{n-1} \to S_n$, and all $a_0, a'_0 \in A_{S_0}, \dots, a_{n-1}, a'_{n-1} \in A_{S_{n-1}}$ such that $a_i \Theta_{S_i} a'_i$, $O^A(a_0, \dots, a_{n-1}) \Theta_{S_n} O^A(a'_0, \dots, a'_{n-1})$ holds. The set of all congruences over A is denoted by $\mathrm{Con}(A)$.

The sorted notions of subalgebra, homomorphism, isomorphism, etc. are defined in a natural way (see [40] for the formal definitions).

For each set of sorts SORT we fix a locally countably infinite sorted set $X = \langle X_S : S \in \mathrm{SORT} \rangle$ of *(propositional sorted) variables*. We assume the components of the sorted set of variables are pairwise disjoint. The elements in X_S are called S*-variables*. To denote that a variable x is of sort S (i.e., that $x \in X_S$) we write $x{:}S$.

We say that a term $O(t_0, \dots, t_{n-1})$, where $O \in \mathrm{OP}_\tau$ with $\tau = S_0, \dots, S_{n-1} \to S_n$, has type S_n. Given a signature Σ we define the SORT-sorted set $\mathrm{Te}_\Sigma(X)$ of terms over the signature Σ with variables in X as usual. Note that, since the components of the family $\mathrm{Te}_\Sigma(X)$ are pairwise disjoint, a SORT-sorted subset Γ of $\mathrm{Te}_\Sigma(X)$ can be identified with the unsorted set $\bigcup_{S \in \mathrm{SORT}} \Gamma_S$. A hidden signature Σ is said to be *standard* if there is a ground term (i.e., a term without variables) of every sort. We use the lower case Greek letters $\varphi, \psi, \vartheta, \dots$ to represent terms, possibly with annotations to indicate sorts of terms and variables. Specifically, writing φ in the form

$$\varphi(x_0{:}S_0, \dots, x_{n-1}{:}S_{n-1}){:}S \tag{1}$$

indicates that φ is of sort S and that the variables that actually occur in φ are included in the list $x_0, \dots, x_{n-1}$ of variables of sorts $S_0, \dots, S_{n-1}$, respectively.

We define, in the usual way, operations over $\mathrm{Te}_\Sigma(X)$ to obtain the *term algebra* over the signature Σ. It is well known that $\mathrm{Te}_\Sigma(X)$ has the *universal mapping* *property over* X in the sense that, for every Σ-algebra A and every sorted map $h : X \to A$, called an *assignment*, there is a unique sorted homomorphism $h^* : \mathrm{Te}_\Sigma(X) \to A$ that extends h. In the sequel, we will not distinguish between these two maps. If φ is the term (1), and $a_i \in A_{S_i}$, $i < n$, we write $\varphi^A(a_0, \dots, a_{n-1})$ for the image $h(\varphi)$ under any homomorphism $h : \mathrm{Te}_\Sigma(X) \to A$ such that $h(x_i) = a_i$ for all $i < n$. A map from X to the set of terms, and its unique extension to an endomorphism of $\mathrm{Te}_\Sigma(X)$, is called a *substitution*.

To provide a context that allows us to deal simultaneously with specification logics that are assertional (for example the ones with a Boolean sort but no equality) and equational, we introduce the notion of a k-term for any nonzero natural number k. In the sequel k denotes a fixed nonzero natural number. A *k-variable of sort S* is a sequence of k variables all of the same sort S. A *k-term (k-*formula *in logical context) of sort S over* Σ is a sequence of k Σ-terms all of the same sort S. We indicate k-terms by overlining, so $\bar{\varphi}(x_1, \cdots, x_n){:}S = \langle \varphi_0(x_1, \cdots, x_n){:}S, \dots, \varphi_{k-1}(x_1, \cdots, x_n){:}S\rangle$. When we do not need to make the common sort S of each term of $\bar{\varphi}{:}S$ explicit, we simply write it as $\bar{\varphi}$. $\mathrm{Te}^k_\Sigma(X)$ is the sorted set of all k-terms over Σ. Thus $\mathrm{Te}^k_\Sigma(X) = \langle (\mathrm{Te}_\Sigma(X))^k_S : S \in \mathrm{SORT}\rangle$. The set of all *visible k-terms* $(\mathrm{Te}^k_\Sigma(X))_{\mathrm{VIS}}$ is the VIS-sorted set $\langle (\mathrm{Te}_\Sigma(X))^k_V : V \in \mathrm{VIS}\rangle$.

2.1. Data structures and Leibniz congruence.

Let Σ be a hidden signature. A *visible k-data structure* (a *k-data structure* for short) *over* Σ is a pair $\mathcal{A} = \langle A, F\rangle$, where A is a Σ-algebra and $F \subseteq A^k_{\mathrm{VIS}}$; A is called the *underlying algebra* and F the *designated filter of* $\mathcal{A}$ (see [39] for examples in the hidden equational case).

Let $\mathcal{A} = \langle A, F\rangle$ be a k-data structure. A congruence relation Θ on A is VIS-*compatible* (or simply *compatible*) with F if for all $V \in \mathrm{VIS}$ and for all $\bar{a}, \bar{a}' \in A^k_V$ the following condition holds.

$$\text{if } a_i \equiv a'_i(\Theta_V) \text{ for all } i \le k \text{ then, } \bar{a} \in F_V \text{ iff } \bar{a}' \in F_V;$$

that is, each F_V is the union of a cartesian product of Θ_V-classes i.e.,

$$F_V = \bigcup_{\bar{a} \in F_V} (a_1/\Theta_V) \times (a_2/\Theta_V) \times \cdots \times (a_k/\Theta_V).$$

Lemma 2.1. *Let $\mathcal{A} = \langle A, F\rangle$ be a k-data structure. There is a largest congruence relation on A compatible with F.*

Proof. Let Φ and Ψ be two congruences on A compatible with F. The relational product $\Phi \circ \Psi$, defined for each $S \in \mathrm{SORT}$ by

$$(\Phi \circ \Psi)_S := \big\{ \langle a, b\rangle \in A^2_S : \exists c \in A_S \; \big(\langle a, c\rangle \in \Phi_S \text{ and } \langle c, b\rangle \in \Psi_S \big) \big\},$$

is also compatible with F. Since the join $\Phi \vee \Psi$, in the lattice of congruences, is given by $\bigcup_{i<\omega} \Phi \circ^i \Psi$, where $\Phi \circ^0 \Psi = \Delta_A$ and $\Phi \circ^{i+1} \Psi = (\Phi \circ^i \Psi) \circ (\Phi \circ \Psi)$, we have that $\Phi \vee \Psi$ is also compatible with F. Hence, the set of all congruence relations on A compatible with F is directed in the sense that, for any pair of congruences compatible with F, there is a third congruence with the same property that includes both of them. We can conclude from this that the union of all compatible congruences is again a compatible congruence. Therefore, the largest congruence compatible with F always exists. □

Definition 2.2. Let $\mathcal{A} = \langle A, F\rangle$ be a k-data structure. The largest congruence relation on A compatible with F is called the *Leibniz congruence of F on A* and is denoted by $\Omega_A(F)$.

The Leibniz congruence plays a central role in abstract algebraic logic when restricted to single-sorted, k-data structures; see for example [46] and [21]. The term was introduced in [8], but the concept appeared much earlier. The motivation behind the choice of the term *Leibniz* will become clear after Theorem 3.3. A systematic study of the Leibniz congruence in hidden k-logics can be found in [35], in particular a proof of its characterization is given in Theorem 3.3. In the case of single-sorted 1-data structures, this result was well known in the literature of sentential logic; see for example [8].

An interesting property of the Leibniz congruence is its preservation under inverse images of surjective homomorphisms, i.e., given a k-data structure $\mathcal{A} = \langle A, F\rangle$ over Σ, a Σ-algebra B and a surjective homomorphism $h : B \to A$, we have that $h^{-1}(\Omega_A(F)) = \Omega_B(h^{-1}(F))$.

2.2. **Hidden k-logic.** For each nonzero natural number k, a hidden k-logic is considered to be a consequence relation on the set of visible k-terms of some hidden signature, independently of any specific choice of axioms and rules of inference. More precisely, it is defined as a substitution invariant consequence relation on the set of visible k-terms.

Definition 2.3. A *hidden k-logical system* (*hidden k-logic* for short) is a pair $\mathcal{L} = \langle \Sigma, \vdash_{\mathcal{L}}\rangle$, where Σ is a hidden signature with VIS as its set of visible sorts, and $\vdash_{\mathcal{L}} \subseteq \mathcal{P}((\mathrm{Te}^k_\Sigma(X))_{\mathrm{VIS}}) \times (\mathrm{Te}^k_\Sigma(X))_{\mathrm{VIS}}$ is an (unsorted) relation that satisfies for all $\Gamma \cup \Delta \cup \{\bar\gamma, \bar\varphi\} \subseteq (\mathrm{Te}^k_\Sigma(X))_{\mathrm{VIS}}$ the following conditions:

(i) $\Gamma \vdash_{\mathcal{L}} \bar\gamma$ for each $\bar\gamma \in \Gamma$;
(ii) if $\Gamma \vdash_{\mathcal{L}} \bar\varphi$, and $\Delta \vdash_{\mathcal{L}} \bar\gamma$ for each $\bar\gamma \in \Gamma$, then $\Delta \vdash_{\mathcal{L}} \bar\varphi$;
(iii) if $\Gamma \vdash_{\mathcal{L}} \bar\varphi$, then $\sigma(\Gamma) \vdash_{\mathcal{L}} \sigma(\bar\varphi)$ for every substitution σ.

Note, that being unsorted, $\vdash_{\mathcal{L}}$ can relate premises and consequences of different visible sorts.

A hidden k-logic is *specifiable* if $\vdash_{\mathcal{L}}$ is finitary (or compact), i.e., if $\Gamma \vdash_{\mathcal{L}} \bar\varphi$ implies $\Delta \vdash_{\mathcal{L}} \bar\varphi$ for some globally finite subset Δ of Γ (recall that a set Γ is said to be globally finite if for every $S \in \mathrm{SORT}$ A_S is a finite set and A_S is empty except for a finite number of sorts). The relation $\vdash_{\mathcal{L}}$ is called *the consequence relation of $\mathcal{L}$*; when $\mathcal{L}$ is clear from the context we simply write $\vdash$. A hidden k-logic with VIS = SORT will be called a *visible k-logic*, or simply a *k-logic*.

As it is usual in a sentential logic framework, we treat formulas (k-formulas) as synonymous to terms (k-terms, respectively). Moreover, for a given hidden k-logic $\mathcal{L} = \langle \Sigma, \vdash_{\mathcal{L}}\rangle$ we denote $\mathrm{Te}^k_\Sigma(X)$ and $(\mathrm{Te}^k_\Sigma(X))_{\mathrm{VIS}}$ by $\mathrm{Fm}(\mathcal{L})$ and $\mathrm{Fm}_{\mathrm{VIS}}(\mathcal{L})$, respectively.

Hidden k-logics were introduced by Martins and Pigozzi (cf. [39]) in the context of the algebraic specification and verification of software systems. The basic theory of hidden k-logics was presented in [35]. The class of hidden k-logics includes such well-known logical systems as the 2-dimensional hidden and standard equational logics, as well as the Boolean logic (for more examples see [35]).

Every consequence relation $\vdash$ has a natural extension to a relation, also denoted by $\vdash$, between sets of visible k-terms; it is defined by $\Gamma \vdash \Delta$ if $\Gamma \vdash \bar\varphi$ for each $\bar\varphi \in \Delta$. We define the relation of *interderivability* between sorted sets in the following way: $\Gamma \dashv\vdash \Delta$ if, $\Gamma \vdash \Delta$ and $\Delta \vdash \Gamma$. We will abbreviate $\{\bar\psi\} \vdash \bar\varphi$, $\Gamma \cup \{\bar\varphi_0, \dots, \bar\varphi_{n-1}\} \vdash \bar\varphi$ and $\Gamma_0 \cup \cdots \cup \Gamma_{n-1} \vdash \bar\varphi$ by $\bar\psi \vdash \bar\varphi$, $\Gamma, \bar\varphi_0, \dots, \bar\varphi_{n-1} \vdash \bar\varphi$ and $\Gamma_0, \dots, \Gamma_{n-1} \vdash \bar\varphi$, respectively.

Let $\mathcal{L}$ be a (not necessarily specifiable) hidden k-logic. By a *theorem* of $\mathcal{L}$ we mean a visible k-term $\bar{\varphi}$ such that $\vdash_{\mathcal{L}} \bar{\varphi}$, i.e., $\varnothing \vdash_{\mathcal{L}} \bar{\varphi}$. The set of all theorems is denoted by $\mathrm{Thm}(\mathcal{L})$. A rule such as

$$\frac{\bar{\varphi}_0 : V_0, \dots, \bar{\varphi}_{n-1} : V_{n-1}}{\bar{\varphi}_n : V_n}, \tag{2}$$

where $\bar{\varphi}_0, \dots, \bar{\varphi}_n$ are all visible k-terms, is said to be a *derivable rule* of $\mathcal{L}$ if $\{\bar{\varphi}_0, \dots, \bar{\varphi}_{n-1}\} \vdash_{\mathcal{L}} \bar{\varphi}_n$. A set of visible k-terms T closed under the consequence relation, i.e., $T \vdash_{\mathcal{L}} \bar{\varphi}$ implies $\bar{\varphi} \in T$, is called a *theory* of $\mathcal{L}$ or $\mathcal{L}$*-theory.* The set of all theories is denoted by $\mathrm{Th}(\mathcal{L})$; it forms a complete lattice under set-theoretic inclusion, which is algebraic if $\mathcal{L}$ is specifiable. Let $T_i \in \mathrm{Th}(\mathcal{L})$, for $i \in I$. Their meet is $\bigcap_{i \in I} T_i$ and their joint is the intersection of all theories that contain each T_i, i.e., $\bigvee_{i \in I}^{\mathcal{L}} T_i = \bigcap\{T \in \mathrm{Th}(\mathcal{L}) : T_i \subseteq T \text{ for all } i \in I\}$. Given any set Γ of visible k-terms, the set $\mathrm{Con}_{\mathcal{L}}(\Gamma)$ is the smallest $\mathcal{L}$-theory containing Γ. It is easy to see that $\mathrm{Con}_{\mathcal{L}}(\Gamma) = \{\, \bar{\varphi} \in (\mathrm{Te}^k_{\Sigma}(X))_{\mathrm{VIS}} : \Gamma \vdash_{\mathcal{L}} \bar{\varphi}\}$, i.e., the set of all consequences of Γ.

Very often, a specifiable hidden k-logic has a Hilbert style presentation, i.e., it is given by a set of axioms (visible k-terms) and inference rules of the general form (2). We say that a visible k-term $\bar{\psi}$ is *directly derivable* from a set Γ of visible k-terms by a rule such as (2) if there is a substitution $h : X \to \mathrm{Te}_{\Sigma}(X)$ such that $h(\bar{\varphi}_n) = \bar{\psi}$ and $h(\bar{\varphi}_0), \dots, h(\bar{\varphi}_{n-1}) \in \Gamma$.

Given a set AX of visible k-terms and a set IR of inference rules, we say that $\bar{\psi}$ is *derivable* from Γ by the set AX and the set IR, in symbols $\Gamma \vdash_{\mathrm{AX,IR}} \bar{\psi}$, if there is a finite sequence of k-terms, $\bar{\psi}_0, \dots, \bar{\psi}_{n-1}$ such that $\bar{\psi}_{n-1} = \bar{\psi}$, and for each $i < n$ one of the following conditions hold:

(a) $\bar{\psi}_i \in \Gamma$,
(b) $\bar{\psi}_i$ is a substitution instance of a k-term in AX
(c) $\bar{\psi}_i$ is directly derivable from $\{\bar{\psi}_j\}_{j<i}$ by one of the inference rules in IR.

It is clear that $\langle \Sigma, \vdash_{\mathrm{AX,IR}} \rangle$ is a specifiable hidden k-logic. Moreover, a hidden k-logic $\mathcal{L}$ is specifiable iff there exist (possibly infinite) sets AX and IR, of axioms and inference rules, respectively, such that, for any visible k-terms $\bar{\psi}$ and any set Γ of visible k-terms, $\Gamma \vdash_{\mathcal{L}} \bar{\psi}$ iff $\Gamma \vdash_{\mathrm{AX,IR}} \bar{\psi}$. The pair $\langle \mathrm{AX}, \mathrm{IR} \rangle$ is called a *presentation* of $\mathcal{L}$ by axioms and inference rules. Hence we can present our examples of specifiable logics by exhibiting their set of axioms and of inference rules. If $\mathcal{L} = \langle \Sigma, \vdash_{\mathrm{AX,IR}} \rangle$, for some AX and IR with $|\mathrm{AX} \cup \mathrm{IR}| < \omega$, we say that $\mathcal{L}$ is *finitely axiomatizable.*

2.2.1. *Semantics.* Let $\mathcal{A} = \langle A, F \rangle$ be a k-data structure. A visible k-term $\bar{\varphi} : V$ is said to be a *semantic consequence* of a set of visible k-terms Γ in $\mathcal{A}$, in symbols $\Gamma \models_{\mathcal{A}} \bar{\varphi}$, if, for every assignment $h : X \to A$, $h(\bar{\varphi}) \in F_V$ whenever $h(\bar{\psi}) \in F_W$ for every $\bar{\psi} : W \in \Gamma$. A visible k-term $\bar{\varphi}$ is a *validity* of $\mathcal{A}$, and conversely $\mathcal{A}$ is a *model* of $\bar{\varphi}$, if $\varnothing \models_{\mathcal{A}} \bar{\varphi}$. A rule such as (2) is a *valid rule* of $\mathcal{A}$, and conversely $\mathcal{A}$ is a *model* of the rule, if $\{\bar{\varphi}_0, \dots, \bar{\varphi}_{n-1}\} \models_{\mathcal{A}} \bar{\varphi}_n$. A visible formula $\bar{\varphi}$ is a *semantic consequence* of a set of visible k-terms Γ for an arbitrary class $\mathcal{M}$ of k-data structures over Σ, in symbols $\Gamma \models_{\mathcal{M}} \bar{\varphi}$, if $\Gamma \models_{\mathcal{A}} \bar{\varphi}$ for each $\mathcal{A} \in \mathcal{M}$. It can be proved that $\models_{\mathcal{M}}$ is always a k-logic, however it may not be specifiable. A visible k-term or rule such as (2) is a *valid* rule of $\mathcal{M}$ if it is a validity of each member of $\mathcal{M}$.

A k-data structure $\mathcal{A}$ is a *model* of a hidden k-logic $\mathcal{L}$ if $\Gamma \vdash_{\mathcal{L}} \bar{\varphi}$ implies $\Gamma \models_{\mathcal{A}} \bar{\varphi}$, for every $\Gamma \cup \{\bar{\varphi}\} \subseteq (\mathrm{Te}^k_{\Sigma}(X))_{\mathrm{VIS}}$. When $\mathcal{A}$ is a model of $\mathcal{L}$ the designated filter of $\mathcal{A}$ is called an $\mathcal{L}$*-filter over* A. The set of all $\mathcal{L}$-filters over an algebra A is denoted by

$\mathrm{Fi}_{\mathcal{L}}(A)$. The special models whose underlying algebra is the formula algebra, i.e., of the form $\langle \mathrm{Te}_\Sigma(X), T\rangle$, with $T \in \mathrm{Th}(\mathcal{L})$ are called *Lindenbaum-Tarski models*. The class of all models of $\mathcal{L}$ is denoted by $\mathrm{Mod}(\mathcal{L})$. If $\mathcal{L}$ is a specifiable hidden k-logic, then $\mathcal{A}$ is a model of $\mathcal{L}$ iff every axiom and rule of inference is a validity of $\mathcal{A}$. The class of all *reduced models of* $\mathcal{L}$, i.e., all models $\langle A, F\rangle$ such that $\Omega_A(F) = \mathrm{id}_A$, is denoted by $\mathrm{Mod}^*(\mathcal{L})$. A class of k-data structures $\mathcal{M}$ is a *data structure semantics* for $\mathcal{L}$ if $\vdash_{\mathcal{L}} = \models_{\mathcal{M}}$. The Completeness Theorem holds for hidden k-logics (cf. [39]), i.e., for every $\Gamma \cup \{\bar\varphi\} \subseteq (\mathrm{Te}^k_\Sigma(X))_{\mathrm{VIS}}$,

$$\Gamma \vdash_{\mathcal{L}} \bar\varphi \quad \text{iff} \quad \Gamma \models_{\mathrm{Mod}(\mathcal{L})} \bar\varphi \quad \text{iff} \quad \Gamma \models_{\mathrm{Mod}^*(\mathcal{L})} \bar\varphi.$$

An important class of hidden 2-logics is the class of *hidden equational logics*, where the notion of equality is only considered for visible data. It is defined (using the reflexivity, symmetry, transitivity and congruence rules) as a sorted equational logic, restricted to the visible part (cf. [39]).

In an equational logic framework, a pair of terms of the same sort $\langle s, t\rangle$ is called *an equation* and it is denoted by $s \approx t$.

Definition 2.4 (Free hidden equational logic, cf. [35])**.** Let Σ be a hidden signature and VIS its set of visible sorts.

(1) The *free hidden equational logic over* Σ (*free* HEL_Σ for short) is the specifiable hidden 2-logic presented as follows:

Axioms: for all $V \in \mathrm{VIS}$

$$x{:}V \approx x{:}V$$

Inference rules: for each $V, W \in \mathrm{VIS}$,

$$(\mathrm{IR}_1) \qquad \frac{x{:}V \approx y{:}V}{y{:}V \approx x{:}V};$$

$$(\mathrm{IR}_2) \qquad \frac{x{:}V \approx y{:}V,\, y{:}V \approx z{:}V}{x{:}V \approx z{:}V};$$

$$(\mathrm{IR}_3) \qquad \frac{\varphi{:}V \approx \psi{:}V}{\vartheta(x/\varphi){:}W \approx \vartheta(x/\psi){:}W} \text{ for each } \vartheta \in \mathrm{Te}_W \text{ and each } x \in X_V.$$

(2) The *free un-hidden equational logic over* Σ (*free* UHEL_Σ, for short) contains an equality predicate for each sort, visible and hidden. The axioms and inference rules are the same as those of the free HEL_Σ, except that V and W are now allowed to range over all sorts. Thus $\mathrm{UHEL}_\Sigma = \mathrm{HEL}_{\Sigma^{\mathrm{UH}}}$.

An *applied hidden equational logic over* Σ (or simply a HEL_Σ) is any hidden 2-logic $\mathcal{L}$ over Σ that satisfies all axioms and rules of inference of the free HEL_Σ. An *applied un-hidden equational logic over* Σ (UHEL_Σ) is defined similarly; the subscript Σ may be omitted if it is clear from the context. We say that a specified applied hidden or unhidden equational logic is *strict equational* if it does not have extra-logical inference rules.

Definition 2.5. Let $\mathcal{L}$ be a HEL_Σ and E a set of equations of arbitrary, possibly un-hidden, sort. We define $\mathcal{L}^{\mathrm{UH}}[E]$ as the natural extension of $\mathcal{L}$ by E to a UHEL over the same signature (when E is empty we just write $\mathcal{L}^{\mathrm{UH}}$).

The standard model of the free HEL_Σ is of the form $\langle A, \mathrm{id}_{A_{\mathrm{VIS}}}\rangle$, where A is a Σ-algebra and $\mathrm{id}_{A_{\mathrm{VIS}}}$ is the identity relation on the visible part of A, but one gets more general 2-data structures as models by taking any congruence relation on the visible part of A in place of $\mathrm{id}_{A_{\mathrm{VIS}}}$. By a *congruence relation on the visible part of* A, or simply a VIS-*congruence*, we mean a VIS-sorted set $\langle\, F_V : V \in \mathrm{VIS}\,\rangle$ such that, for every $V \in \mathrm{VIS}$, F_V is an equivalence relation on A_V, and for every term $\varphi(x_0{:}V_0,\dots,x_{n-1}{:}V_{n-1},y_0{:}H_0,\dots,y_{m-1}{:}H_{m-1}){:}V$ with $V_0,\dots,V_{n-1},V \in$ VIS and $H_0,\dots,H_{m-1},\in$ HID, if $\langle a_i,b_i\rangle \in F_{V_i}$ for all $i<n$, then for all $c_j \in A_{H_j}$ $j<m$,

$$\langle \varphi^A(a_0,\dots,a_{n-1},c_0,\dots,c_{m-1}),\varphi^A(b_0,\dots,b_{n-1},c_0,\dots,c_{m-1})\rangle \in F_V.$$

The basic notions and results about hidden k-logics, as well as many examples of HELs may be found in [35] and [39]. An interesting fact about HELs is that the visible consequences of any set of visible equations are the same either for $\mathcal{L}^{\mathrm{UH}}$ or for $\mathcal{L}$, i.e., for any $\Gamma\cup\{t\approx t'\}\subseteq \mathrm{Te}^2_\Sigma(X)_{\mathrm{VIS}}$, $\Gamma\vdash_{\mathcal{L}^{\mathrm{UH}}} t\approx t'$ iff $\Gamma\vdash_{\mathcal{L}} t\approx t'$.

2.3. **Concrete examples.** We give several examples of specifiable hidden logics. We have purposely chosen simple, well-known ones that allow us to illustrate the basic ideas without burdening the reader with irrelevant detail. The first two illustrate how the logic of a particular data structure can be alternatively formalized as a Boolean 1-logic and as an equational 2-logic, a HEL. The flag logics provide two different ways of specifying semaphores, which are commonly used in scheduling resources [24].

Example 2.6. (**Flags** as a Boolean 1-logic)
Consider the hidden signature Σ_{flag}:

SORT $=\{\mathrm{flag},\mathrm{bool}\}$, with bool the unique visible sort, and the following operation symbols:

$$\begin{array}{ll} \mathrm{up}:\mathrm{flag}\to\mathrm{flag}; & \mathrm{rev}:\mathrm{flag}\to\mathrm{flag};\\ \mathrm{dn}:\mathrm{flag}\to\mathrm{flag}; & \mathrm{up?}:\mathrm{flag}\to\mathrm{bool}, \end{array}$$

and the operation symbols for the Boolean part: $\neg,\wedge,\vee$, true and false. The Boolean biconditional $\varphi\leftrightarrow\psi$ is an abbreviation for the compound operation $(\neg\varphi\vee\psi)\wedge(\neg\psi\vee\varphi)$.

The Boolean logic of flags, $\mathcal{L}_{\mathrm{bflag}}$, is the 1-logic with the following extra-logical axioms:

$$\begin{array}{ll} \mathrm{up?}(\mathrm{up}(F)) & \mathrm{up?}(\mathrm{rev}(F))\leftrightarrow\neg(\mathrm{up?}(F))\\ \neg\mathrm{up?}(\mathrm{dn}(F)) & \end{array}$$

and including usual logical axioms for the classical propositional logic. There are no extra-logical rules of inference. ♢

Example 2.7. (**Flags** as a HEL) The signature is the same as above.
The equational logic of flags, $\mathcal{L}_{\mathrm{eflag}}$, is the $\mathrm{HEL}_{\Sigma_{\mathrm{flag}}}$ with the following extra-logical axioms:

$$\begin{array}{ll} \mathrm{up?}(\mathrm{up}(F))\approx\mathrm{true} & \mathrm{up?}(\mathrm{rev}(F))\approx\neg(\mathrm{up?}(F))\\ \mathrm{up?}(\mathrm{dn}(F))\approx\mathrm{false} & \end{array}$$

and including the usual logical axioms for Boolean algebra. There are no extra-logical rules of inference. ♢

As expected, $\mathcal{L}_{\mathrm{bflag}}$ and $\mathcal{L}_{\mathrm{eflag}}$ are equivalent. Precisely, $\dfrac{\varphi_1\leftrightarrow\varphi_1',\dots,\varphi_n\leftrightarrow\varphi_n'}{\psi\leftrightarrow\psi'}$ is a derivable rule of $\mathcal{L}_{\mathrm{bflag}}$ iff $\dfrac{\varphi_1\approx\varphi_1',\dots,\varphi_n\approx\varphi_n'}{\psi\approx\psi'}$ is a derivable rule of $\mathcal{L}_{\mathrm{eflag}}$.

Example 2.8. (**Stacks of Natural Numbers** as a HEL) As in the standard specification of the logic of stacks, only the natural numbers are visible. Consequently, the axioms and rules of inference can only reference "numerical behavior" of stacks rather than the stacks themselves. In particular there can be no axiom or rule involving equality between stacks. Because of this we get an infinite number of axioms, while in the standard formalizations, where assertions about the equality of stacks are allowed, the axiomatization is finite and conceptually simpler.

The specification differs from the usual one in another regard. The top of the empty stack is zero and pushing zero on the empty stack gives the empty stack. This is done to simplify the specification logic and agrees with what is done in [25].

Consider the hidden signature Σ_{stacks}:

SORT $= \{\text{nat}, \text{stack}\}$, with nat the unique visible sort and the following operation symbols:

$\text{empty} : \quad \to \text{stack}$
$\text{zero} : \quad \to \text{nat}$
$\text{push} : \text{nat}, \text{stack} \to \text{stack}$
$\text{top} : \text{stack} \to \text{nat}$
$\text{pop} : \text{stack} \to \text{stack}$
$s : \text{nat} \to \text{nat}$

The specification logic of stacks, $\mathcal{L}_{\text{stacks}}$, is the logic with hidden signature Σ_{stacks} and the following axioms and inference rules:

Extra-logical axioms:

$\text{top}(\text{pop}^n(\text{empty})) \approx \text{zero}, \text{ for all } n;$
$\text{top}(\text{push}(x, y)) \approx x;$
$\text{top}(\text{pop}^{n+1}(\text{push}(x, y))) \approx \text{top}(\text{pop}^n(y)), \text{ for all } n.$

Extra-logical inference rule:
$s(x) \approx s(y) \to x \approx y.$ ♢

2.3.1. *Other hidden k-logics.*

Example 2.9 (Free inequational logic)**.** Let Σ be any one-sorted signature. The *free inequational logic* is the one-sorted 2-logic over Σ defined by the axioms and inference rules in Fig. 1. As in the equational case, we use a special symbol to denote the 2-formula $\langle \varphi, \psi \rangle$; namely we write $\varphi \preceq \psi$ for $\langle \varphi, \psi \rangle$. This logic is relevant in the context of ordered (universal) algebra (see [54]) and abstract algebra. We can generalize the inequational logic to the sorted case and, more generally, to the hidden sorted case in the same way we generalized the equational logic to the

Axioms:
$x \preceq x;$
Inference rules:

$$\frac{x \preceq y \, , y \preceq z}{x \preceq z};$$

$$\frac{x_0 \preceq y_0, \dots, x_{n-1} \preceq y_{n-1}}{O(x_0, \dots, x_{n-1}) \preceq O(y_0, \dots, y_{n-1})} \, ,$$

for every operation symbol O.

Figure 1. Free inequational logic.

hidden equational logic. A more general notion of inequational logic can be found in [2]. ♢

Example 2.10 (Stacks of natural numbers with Booleans)**.** The signature is obtained from the signature of stacks of natural numbers by adjoining a new sort bool, for the Boolean operation symbols, and one new attribute $eq : \text{nat}, \text{nat} \to \text{bool}$, the equality test for natural numbers. The sort bool is the only visible sort. Informally, we can say that the axioms and inference rules are obtained by applying eq to each of the axioms and inference rules of the specification of stacks (see Fig. 2). The operation symbol eq is called an *equational test function* and the models are called **generalized equality test models**. These models have been studied in [44].

Axioms:

$eq(x, x)$

$eq(\text{top}(\text{pop}^n(empty)), zero)$, for all n;

$eq(\text{top}(\text{push}(x, y)), x)$;

$eq(\text{top}(\text{pop}^{n+1}(\text{push}(x, y))), \text{top}(\text{pop}^n(y)))$, for all n;

Inference rules:

$$\frac{eq(x,y)}{eq(y,x)} \qquad\qquad \frac{eq(x,y)}{eq(s(x),s(y))}$$

$$\frac{eq(x,y), eq(y,z)}{eq(x,z)} \qquad\qquad \frac{eq(s(x),s(y))}{eq(x,y)}$$

Figure 2. Stacks of natural numbers with Booleans.

♢

3. Behavioral equivalence

In hidden equational logic, we may say that two hidden data elements of the same sort are *behaviorally equivalent* if any visible procedure returns the same value when executed with either of the two objects as input. The notion arises from the alternative view of a data structure as a transition system in which the hidden data elements represent states of the system and the operations (i.e., the *methods*) that return hidden, as opposed to visible, elements induce transitions between states.

In the formalism of HEL, the concept of procedure takes the form of a context. Formally, a *S-context* over a hidden signature Σ is a term

$$\varphi(z{:}S, u_1{:}T_1, \dots, u_m{:}T_m){:}U \tag{3}$$

with a distinguished variable z of sort S and parametric variables $u_1, \dots, u_m$ of arbitrary (visible or hidden) sort. It is a *visible context* if the sort U of φ is visible.

Definition 3.1. Let A be a Σ-algebra and let S be a arbitrary sort. Then, $a, a' \in A_S$ are *behaviorally equivalent in* A, in symbols $a \equiv^{\text{beh}}_A a'$, if for every visible S-context $\varphi(z{:}S, u_1{:}T_1, \dots, u_m{:}T_m)$ and for all $b_1 \in A_{T_1}, \dots, b_m \in A_{T_m}$,

$$\varphi^A(a, b_1, \dots, b_m) = \varphi^A(a', b_1, \dots, b_m).$$

Variants of this notion of behavioral equivalence have occurred in the literature. For example, Goguen and Malcolm [25] restrict the set of contexts to the ones built from a predefined set of observational operational symbols.

In order to generalize the notion of behavioral equivalence to hidden k-logics we first generalize the notion of context. A (k,S)-*context* over a hidden signature Σ is a k-term

$$\begin{aligned}(4)\quad &\bar\varphi(z{:}S, u_1{:}T_1, \dots, u_m{:}T_m){:}U\\ &\qquad = \big\langle \varphi_1(z{:}S, u_1{:}T_1, \dots, u_m{:}T_m), \dots, \varphi_k(z{:}S, u_1{:}T_1, \dots, u_m{:}T_m)\big\rangle{:}U\end{aligned}$$

with a distinguished variable z of sort S and parametric variables $u_1, \dots, u_m$. It is a *visible context* if the sort U of $\bar\varphi$ is visible.

Definition 3.2. Let $\mathcal{A} = \langle A, F\rangle$ be a k-data structure over a hidden signature Σ. Two elements a, a' of A of arbitrary sort S are said to be *behaviorally equivalent* in $\mathcal{A}$, in symbols $a \equiv^{\text{beh}}_{\mathcal{A}} a'$, if for every visible (k,S)-context $\bar\varphi(z{:}S, u_1{:}T_1, \dots, u_m{:}T_m){:}V$ and for all $b_1 \in A_{T_1}, \dots, b_m \in A_{T_m}$,

$$(5)\qquad \bar\varphi^A(a, b_1, \dots, b_m) \in F_V \quad \text{iff} \quad \bar\varphi^A(a', b_1, \dots, b_m) \in F_V.$$

This notion does indeed generalize behavioral equivalence in equational logic, since, as a consequence of Theorem 3.4 below, we have that a and a' are behaviorally equivalent in a Σ-algebra A iff they are behaviorally equivalent in the 2-dimensional equality data structure $\langle A, \text{id}_{A_{\text{VIS}}}\rangle$ in the sense of Definition 3.2.

Behavioral equivalence over a k-data structure turns out to be a congruence relation on the underlying algebra of the data structure with special properties. In the 1-sorted, 1-data structures (called *matrices*) which constitute the natural models of sentential logic, the detailed combinatorial analysis of this congruence constitutes the basis of a branch of mathematical logic called *abstract algebraic logic* (cf. [46]). Our intention here is to extend this analysis to the behavioral congruences of arbitrary multi-sorted k-data structures and in particular to the models of hidden equational logic. The following two theorems constitute the basis of this approach. They are due to Manuel Martins and Don Pigozzi in [39]. We include their proofs here since this paper is intended also as a survey.

Theorem 3.3. *Let Σ be a hidden signature and let $\mathcal{A} = \langle A, F\rangle$ be a k-data structure over Σ. Then, $\equiv^{\text{beh}}_{\mathcal{A}} = \Omega_A(F)$, i.e., for every $S \in \text{SORT}$ and for all $a, a' \in A_S$, $a \equiv^{\text{beh}}_{\mathcal{A}} a'$ iff $a \equiv a'\ (\Omega_A(F)_S)$.*

Proof. It is easy to see that $\equiv^{\text{beh}}_{\mathcal{A}}$ is an equivalence relation on A. To see that it is a congruence relation, let O be an operation symbol of type $T_1, \dots, T_n \to S$ and suppose $a_i \equiv^{\text{beh}}_{\mathcal{A}} a'_i$, $1 \le i \le n$. We must show that, for any visible (k,S)-context $\bar\varphi(z{:}S, \bar u{:}\bar Q){:}V$, with the designated variable $z{:}S$, and for all parameters $\bar b \in A_{\bar Q}$, we have

$$(6)\qquad \bar\varphi^A\big(O^A(\bar a), \bar b\big) \in F_V \quad \text{iff} \quad \bar\varphi^A\big(O^A(\bar a'), \bar b\big) \in F_V.$$

Consider any $i \le n$. Using the assumption $a_i \equiv^{\text{beh}}_{\mathcal{A}} a'_i$, and taking x_i as the designated variable, $x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n, u_1, \dots, u_n$ as parametric variables, and $a_1, \dots, a_{i-1}$, $a'_{i+1}, \dots, a'_n, b_1, \dots, b_m$ as parameters we have

$$\begin{aligned}&\bar\varphi^A\big(O^A(a_1, \dots, a_{i-1}, a_i, a'_{i+1}, \dots, a'_n), \bar b\big) \in F_V\\ &\qquad\qquad\qquad \text{iff} \quad \bar\varphi^A\big(O^A(a_1, \dots, a_{i-1}, a'_i, a'_{i+1}, \dots, a'_n), \bar b\big) \in F_V.\end{aligned}$$

Since this equivalence holds for all $i \le n$, (6) holds, and hence $\equiv^{\text{beh}}_{\mathcal{A}}$ is a congruence on A.

To see that $\equiv^{\mathrm{beh}}_{\mathcal{A}}$ is compatible with F, consider $\bar{a}, \bar{a}' \in A^k_V$ such that $\bar{a} \left(\equiv^{\mathrm{beh}}_{\mathcal{A}}\right)^k_V \bar{a}'$. Consider the k-sequence of pairwise distinct variables $\bar{x} = \langle x_1 \!:\! V, \dots, x_k \!:\! V\rangle$ (called a *k-variable*, a special k-formula). For each i, $1 \le i \le k$, view $x_1, \cdots, x_n$ as a (k, V)-context with designated variable x_i and treat $a_1, \dots, a_{i-1}, a'_{i+1}, \dots, a'_k$ as parameters. Then from the assumption $a_i \left(\equiv^{\mathrm{beh}}_{\mathcal{A}}\right)_V a'_i$ we conclude that

$$\langle a_1, \dots, a_{i-1}, a_i, a'_{i+1}, \dots, a'_n\rangle \in F_V \quad \text{iff} \quad \langle a_1, \dots, a_{i-1}, a'_i, a'_{i+1}, \dots, a'_n\rangle \in F_V.$$

So $\bar{a} \in F_V$ iff $\bar{a}' \in F_V$. Thus $\equiv^{\mathrm{beh}}_{\mathcal{A}}$ is compatible with F.

Finally, we must show that $\equiv^{\mathrm{beh}}_{\mathcal{A}}$ is the largest congruence on A compatible with F. Let Θ be any congruence on A that is compatible with F. Assume $a \equiv a' \ (\Theta_S)$. Let $\bar{\varphi}(z \!:\! S,\ \bar{u} \!:\! \bar{Q}) \!:\! V$ be a visible (k, S)-formula with designated variable $z \!:\! S$, and let $\bar{b} \in A_{\bar{Q}}$ be a system of parameters. By the congruence property, $\bar{\varphi}^A(a, \bar{b}) \equiv \bar{\varphi}^A(a', \bar{b}) \left(\Theta^k_V\right)$. So by the compatibility of Θ with F we have $\bar{\varphi}^A(a, \bar{b}) \in F_V$ iff $\bar{\varphi}^A(a', \bar{b}) \in F_V$. Thus $\Theta \subseteq \equiv^{\mathrm{beh}}_{\mathcal{A}}$. □

This theorem shows the adequacy of using the Leibniz congruence to study behavioral equivalence. Moreover, when applied to hidden equational logics, Theorem 3.3 takes a more natural form in terms of 1-dimensional contexts as we now see.

Theorem 3.4. *Let Σ be a hidden signature and let $\mathcal{A} = \langle A, F\rangle$ be a model of the free* HEL_Σ*, i.e., F is a* VIS*-congruence on A. Then, for every $S \in$* SORT *and all $a, a' \in A_S$, $a \equiv_{\Omega(F)_S} a'$ iff, for every visible S-context $\varphi(z \!:\! S, u_1 \!:\! Q_1, \dots, u_m \!:\! Q_m) \!:\! V$ and for all $b_1 \in A_{Q_1}, \dots, b_m \in A_{Q_m}$,*

$$\varphi^A(a, b_1, \dots, b_m) \equiv \varphi^A(a', b_1, \dots, b_m)\, (F_V). \tag{7}$$

Proof. By Theorem 3.3, $a \equiv_{\Omega(F)_S} a'$ iff, for every $(2, S)$-context $\langle \varphi(z \!:\! S,\ \bar{u} \!:\! \bar{Q}), \psi(z \!:\! S,\ \bar{u} \!:\! \bar{Q})\rangle$ of sort V, and every $\bar{b} \in A_{\bar{Q}}$,

$$\varphi^A(a,\, \bar{b}) \equiv \psi^A(a,\, \bar{b})\, (F_V) \quad \text{iff} \quad \varphi^A(a',\, \bar{b}) \equiv \psi^A(a',\, \bar{b})\, (F_V). \tag{8}$$

Suppose (7) holds for every S-context $\varphi(z, \bar{u})$ and every $\bar{b} \in A_{\bar{Q}}$. If $\varphi^A(a,\, \bar{b}) \equiv_{F_V} \psi^A(a,\, \bar{b})$, then

$$\varphi^A(a',\, \bar{b}) \equiv \varphi^A(a,\, \bar{b}) \equiv \psi^A(a,\, \bar{b}) \equiv \psi^A(a', \bar{b})\, (F_V)$$

(the first and third equivalences hold because F is a VIS-congruence). Thus (8) holds for every pair of S-contexts and every sequence of parameters $\bar{b}$, hence, $a \equiv_{\Omega(F)_V} a'$.

Conversely, assume $a \equiv_{\Omega(F)_V} a'$. Let $\varphi(z \!:\! S,\ \bar{u} \!:\! \bar{Q}) \!:\! V$ be an arbitrary visible S-context, where $\bar{u} \!:\! \bar{Q} = \langle u_1 \!:\! Q_1, \dots, u_n \!:\! Q_n\rangle$. Let u_{n+1} be a new parametric variable of sort V; the single term u_{n+1} can be viewed as a visible S-context with designated variable z (which does not actually occur) and parametric variables $\bar{u}^+ := \langle u_1, \dots, u_n, u_{n+1}\rangle$. φ can also be viewed as an S-context with the same parametric variables. Let $\langle b_1, \dots, b_n\rangle$ be any system of parameters of sort $\bar{Q}$, and extend it to a system $\bar{b}^+ := \langle b_1, \dots, b_n, b_{n+1}\rangle$, where $b_{n+1} = \varphi^A(a, \bar{b})$. Thus $\varphi^A(a, \bar{b}^+) = b_{n+1} = u^A_{n+1}(a, \bar{b}^+)$. So by (8), $\varphi^A(a', \bar{b}^+) \equiv_{F_V} u^A_{n+1}(a', \bar{b}^+)$. But $u^A_{n+1}(a', \bar{b}^+)$ also equals b_{n+1}. So $\varphi^A(a, \bar{b}) \equiv_{F_V} \varphi^A(a', \bar{b})$. Thus (7) holds for every S context $\varphi(z, \bar{u})$ and every $\bar{b} \in A_{\bar{Q}}$. □

Applying this result to equality models (i.e., models whose filter is the identity), we get that a and a' are behaviorally equivalent, in the sense of Definition 3.1, iff $a \equiv a' \left(\Omega_A(\mathrm{id}_{A_{\mathrm{VIS}}})\right)$; hence behavioral equivalence over k-data structures does indeed generalize the familiar notion of behavioral equivalence over a sorted algebra. This result was obtained independently by Goguen and Malcolm [25].

For hidden equational logics the Leibniz relation has the following useful property; this can also be found in [24, 25] for the case of equality models.

Corollary 3.5. *Let $\mathcal{A} = \langle A, F\rangle$ be a model of the free* HEL_Σ. *Then $\Omega_A(F)$ is the largest congruence on A whose visible part is F.*

Proof. Suppose $a \equiv a' \left(\Omega_A(F)_V\right)$ with $V \in \mathrm{VIS}$. Let z be a variable of sort V. Then z is a visible V-context and hence $a = z^A(a) \equiv z^A(a') = a' \mod F_V$. Thus $\Omega_A(F)_{\mathrm{VIS}} \subseteq F$. Conversely, assume $a \equiv a' \mod F_V$. Then for every V-context $\varphi(z, \bar{u})$ and every choice of parameters $\bar{b} \in A_{\bar{Q}}$, we have $\varphi^A(a, \bar{b}) \equiv \varphi^A(a', \bar{b}) \mod F_V$. Thus $a \equiv a' \;(\Omega_A(F)_V)$ and hence $\Omega_A(F)_{\mathrm{VIS}} = F$. If Θ is any congruence on A such that $\Theta_{\mathrm{VIS}} = F$, then Θ is compatible with F, and hence $\Theta \subseteq \Omega_A(F)$. □

As a special case (for equality models) we have that $\Omega_A(\mathrm{id}_{A_{\mathrm{VIS}}})_{\mathrm{VIS}} = \mathrm{id}_{A_{\mathrm{VIS}}}$, i.e., two visible elements of a Σ-algebra are behaviorally equivalent iff they are equal. However, in computer science it is important to establish procedures to check if two elements are behaviorally equivalent. The last result allows the following method of coinduction.

Given a data structure $\langle A, F\rangle$ we want to know if a pair $\langle a, a'\rangle \in A_S^2$ is in $\Omega(F)_S$. Our method consists of the following three steps:

1 - Define a suitable relation R on A, such that the visible part is F;
2 - Show that this relation is a congruence on A;
3 - Finally show that a and a' are equivalent modulo R.

At first glance, step 1 of this method seems to be a very hard task, however it works very well in many concrete examples. In Example 2.7, we can use this method to prove that $\mathrm{rev}(\mathrm{rev}(F)) \approx \mathrm{rev}(F)$ is behaviorally equivalent in any equality model A of $\mathcal{L}_{\mathrm{eflag}}$. It is enough to consider $R := \left\{(a, a') \in A^2_{\mathrm{flag}} : \mathrm{up?}^A(a) = \mathrm{up?}^A(a')\right\}$.

When applied to Lindenbaum models, Corollary 3.5 gives rise to the following results.

Corollary 3.6. *Let $\mathcal{L}$ be a* HEL *and G an $\mathcal{L}^{\mathrm{UH}}$-theory. Then $G \subseteq \Omega(G_{\mathrm{VIS}})$ and $G_{\mathrm{VIS}} = \Omega(G_{\mathrm{VIS}})_{\mathrm{VIS}}$.*

We can also conclude from Theorem 3.4 that for every sorted algebra A, the operator $\Omega_A : \mathrm{Fi}_{\mathcal{L}}(A) \to \mathrm{Con}(A)$ defined by mapping each $F \in \mathrm{Fi}_{\mathcal{L}}(A)$ into $\Omega_A(F)$ is injective and monotonic.

Corollary 3.7. *Let $\mathcal{L}$ be a* HEL. *Then for every sorted algebra A, Ω_A is injective and monotonic.*

Proof. The proof of injectivity is obvious.

Let $F, G \in \mathrm{Fi}_{\mathcal{L}}(A)$ such that $F \subseteq G$. Suppose that $a \equiv a'(\Omega(F)_S)$. Then for every S-context $\varphi(z{:}S, \overline{x}){:}V$ and for all $\overline{b} \in A^k$, $\varphi^A(a, \overline{b}) \equiv \varphi^A(a', \overline{b})\,(F_V)$. Hence, for every S-context $\varphi(z{:}S, \overline{x}){:}V$ and for all $\overline{b} \in A^k$, $\varphi^A(a, \overline{b}) \equiv \varphi^A(a', \overline{b})\,(G_V)$. Therefore, $a \equiv a'(\Omega(G)_S)$. □

For arbitrary hidden logics this result is false, even in the one-sorted case. The class of logics for which Ω is injective and monotonic is called *Weakly Algebraizable*

Logics. (This class was investigated by Czelakowski and Jansana in [14]). Injectivity and monotonicity are independent properties in sense that neither one of them implies the other (see [16] for an example of a non monotonic injective logic).

For some specifications, certain intuitive properties are not satisfied in the usual sense. This is the case, for instance, of the usual specification of Flags (see [24]), where equation $\mathrm{rev}(\mathrm{rev}(F)) \approx F$ should be a property of the specification of Flags but is not a theorem of the $\mathcal{L}^{\mathrm{UH}}_{\mathrm{eflag}}$. Therefore, we consider, next, a weaker notion of satisfaction, called *behavioral satisfaction.* The perspective according to which satisfaction is to be considered behaviorally is called *behavioral approach* (see, e.g., [50]).

Definition 3.8. Let $t \approx t'$ be an equation of arbitrary sort, and $\mathcal{A} = \langle A, F\rangle$ a k-data structure. We say that the equation $t \approx t'$ is *behaviorally satisfied in* $\mathcal{A}$, in symbols $\models^{beh}_{\mathcal{A}} t \approx t'$, if for all $h : X \to A$, $h(t) \equiv^{beh}_{\langle A,F\rangle} h(t')$. Let $\mathcal{L}$ be a HEL and let $\mathrm{Mod}(\mathcal{L})^{=}$ denote $\{A : \langle A, id_{A_{\mathrm{VIS}}}\rangle \in \mathrm{Mod}(\mathcal{L})\}$. We say that $t \approx t'$ is *behaviorally valid over* $\mathcal{L}$, in symbols $\models^{beh}_{\mathcal{L}} t \approx t'$, if for every $A \in \mathrm{Mod}(\mathcal{L})^{=}$, $\models^{beh}_{\langle A, id_{A_{\mathrm{VIS}}}\rangle} t \approx t'$. If $\mathcal{L}$ is clear from the context we simply write, $\models^{beh} t \approx t'$.

In the example of $\mathcal{L}_{\mathrm{eflag}}$, we have just sketched the proof that equation $\mathrm{rev}(\mathrm{rev}(F)) \approx F$ is behaviorally satisfied in each algebra in $\mathrm{Mod}(\mathcal{L}_{\mathrm{eflag}})^{=}$.

Lemma 3.9. *Let $\mathcal{A} = \langle A, F\rangle$ be a data structure over a hidden signature Σ and S be an arbitrary sort. Then, $\models^{beh}_{\mathcal{A}} t \approx t'$ iff for every $h : X \to A$ and every visible S-context $\varphi(z{:}S, x_1, \ldots, x_n){:}V$,*

$$h(\varphi(t, x_1, \ldots, x_n)) \equiv h(\varphi(t', x_1, \ldots, x_n))\ (F_V)$$

We are going to introduce some notation.

Definition 3.10. Let $t \approx t'$ be an equation of type S. We define $\Delta[t \approx t']$ to be the set $\{t \approx t'\}$, if $S \in \mathrm{VIS}$; and $\Delta[t \approx t']$ to be the set $\{\varphi(t, x_1, \ldots, x_n) \approx \varphi(t', x_1, \ldots, x_n) | \varphi(z{:}S, x_1, \ldots, x_n) \in (\mathrm{Te}_\Sigma(X))_{\mathrm{VIS}}\}$, if $S \notin \mathrm{VIS}$.

The following lemma is a useful characterization for an equation to be behaviorally satisfied in an algebra A (the version for equality models was proved by Roşu in [50]).

Lemma 3.11. *Let $\mathcal{A} = \langle A, F\rangle$ be a data structure over a hidden signature Σ and S be an arbitrary sort. Then the following conditions are equivalent:*

(i) $\models^{beh}_{\mathcal{A}} t \approx t'$;
(ii) $\models_{\mathcal{A}} \Delta[t \approx t']$.

In the special case of data structures of form $\langle \mathrm{Te}_\Sigma(X), T\rangle$, with T a theory, we call the equations in $\Omega(T)$, *behavioral consequences* of T. Behavioral consequences of a substitution invariant theory T may be characterized in the following way.

Corollary 3.12. *Let $\mathcal{L}$ be a* HEL, *T a substitution invariant theory of $\mathcal{L}$ and $t, t' \in (\mathrm{Te}_\Sigma(X))_S$. Then the following are equivalent:*

(i) $t \equiv t'$ ($\Omega(T)_S$);
(ii) *for every visible S-context $\varphi(z{:}S, x_1, \ldots, x_n)$,*
$T \vdash_{\mathcal{L}} \varphi(t, x_1, \ldots, x_n) \approx \varphi(t', x_1, \ldots, x_n)$.

A detailed study of the properties of $\Omega(T)$ can be found in [39].

3.1. **Formal Behavioral Consequence Relation.** In this section we present a characterization of the Leibniz congruence in terms of the consequence relation of $\mathcal{L}^{\mathrm{UH}}$. This characterization justifies the name we have been using for the elements in $\Omega(T)$ - behavioral consequences of T.

This result generalizes, in two directions, the work of Leavens and Pigozzi in [33]. On the one hand, we allow conditional equations as axioms (instead of equations only); on the other hand, we characterize $\Omega(T)$ for all theories (instead of just $\Omega(\mathrm{Thm}(\mathcal{L}))$).

In the second part of this section, we present a simpler characterization for the special case of strict HEL, by dropping the condition that G has to range over all theories.

Now we consider the following generalization of the definition of formal behavioral consequence (introduced in [33]).

Definition 3.13. Let $\mathcal{L}$ be a HEL, $T \in \mathrm{Th}(\mathcal{L})$ and F a set of equations over Σ. Then we say that F is a *global formal behavioral consequence* of T, in symbols $T \vdash^{GFB}_{\mathcal{L}} F$, if for every $G \in \mathrm{Th}(\mathcal{L}^{\mathrm{UH}})$, such that $T \subseteq G$, and every visible equation $s \approx s'$, $G \cup F \vdash_{\mathcal{L}^{\mathrm{UH}}} s \approx s'$ implies that $G \vdash_{\mathcal{L}^{\mathrm{UH}}} s \approx s'$. If $F = \{t \approx t'\}$, then we say that $t \approx t'$ is a *global formal behavioral consequence* of T and we write $T \vdash^{GFB}_{\mathcal{L}} t \approx t'$.

Let $\mathcal{L}$ be a HEL and $T \in \mathrm{Th}(\mathcal{L})$. We define the following relation on the term algebra. For each sort S, $\mathcal{GFB}(T)_S$ is the set of all pairs (t, t') of formulas of type S, such that $T \vdash^{GFB}_{\mathcal{L}} t \approx t'$. Thus, $\mathcal{GFB}(T) = \langle \mathcal{GFB}(T)_S : S \in \mathrm{SORT}\rangle$.

The global formal behavioral consequence relation, as a property of a set, can actually be reduced to a property of its individual members.

Theorem 3.14. *Let $\mathcal{L}$ be a specifiable* HEL *and F be a set of equations. Then for every $T \in \mathrm{Th}(\mathcal{L})$,*

$$T \vdash^{GFB}_{\mathcal{L}} F \Leftrightarrow (T \vdash^{GFB}_{\mathcal{L}} t \approx t',\ \textit{for all}\ \ t \approx t' \in F) \tag{9}$$

Proof. The implication from left to right is obvious. To prove the converse we first show that it holds for any finite F, using induction on the number of elements in F.

Let $G \in \mathrm{Th}(\mathcal{L}^{\mathrm{UH}})$ such that $T \subseteq G$ and $s, s' \in (\mathrm{Te}_\Sigma(X))_{\mathrm{VIS}}$. Suppose that $G \cup F \vdash_{\mathcal{L}^{\mathrm{UH}}} s \approx s'$.

If F has only one equation, we have, by hypothesis, that $G \vdash_{\mathcal{L}^{\mathrm{UH}}} s \approx s'$.

Let now F be the union of F' with $\{t \approx t'\}$. Suppose that $G \cup F' \cup \{t \approx t'\} \vdash_{\mathcal{L}^{\mathrm{UH}}} s \approx s'$. Hence, $\mathrm{Cn}_{\mathcal{L}^{\mathrm{UH}}}(G \cup F') \cup \{t \approx t'\} \vdash_{\mathcal{L}^{\mathrm{UH}}} s \approx s'$. Since, $T \vdash^{GFB}_{\mathcal{L}} t \approx t'$, we have that $\mathrm{Cn}_{\mathcal{L}^{\mathrm{UH}}}(G \cup F') \vdash_{\mathcal{L}^{\mathrm{UH}}} s \approx s'$. Hence, $G \cup F' \vdash_{\mathcal{L}^{\mathrm{UH}}} s \approx s'$. Finally, by the induction hypothesis, $G \vdash_{\mathcal{L}^{\mathrm{UH}}} s \approx s'$. So, we have just proved that $T \vdash^{GFB}_{\mathcal{L}} F$.

Let now F be any set of equations. Let $G \in \mathrm{Th}(\mathcal{L}^{\mathrm{UH}})$ such that $T \subseteq G$ and $s, s' \in (\mathrm{Te}_\Sigma(X))_{\mathrm{VIS}}$. Suppose that $G \cup F \vdash_{\mathcal{L}^{\mathrm{UH}}} s \approx s'$. Then, there is a finite subset F_0 of F such that $G \cup F_0 \vdash_{\mathcal{L}^{\mathrm{UH}}} s \approx s'$. From the discussion above, $G \vdash_{\mathcal{L}^{\mathrm{UH}}} s \approx s'$. □

The following lemma shows that the global formal behavioral consequence is closed under ordinary equational deduction.

Lemma 3.15. *Let $\mathcal{L}$ be a specifiable* HEL, *$T \in \mathrm{Th}(\mathcal{L})$ and F a set of equations. Then $T \vdash^{GFB}_{\mathcal{L}} F$ and $F \vdash_{\mathcal{L}^{\mathrm{UH}}} t \approx t'$ implies that $T \vdash^{GFB}_{\mathcal{L}} t \approx t'$.*

Proof. Let $G \in \mathrm{Th}(\mathcal{L}^{\mathrm{UH}})$, such that $T \subseteq G$, and $s \approx s'$ be a visible equation. Suppose that $G \cup \{t \approx t'\} \vdash_{\mathcal{L}^{\mathrm{UH}}} s \approx s'$. Then, $G \cup F \vdash_{\mathcal{L}^{\mathrm{UH}}} s \approx s'$. Since $T \vdash_{\mathcal{L}}^{GFB} F$, we have that $G \vdash_{\mathcal{L}^{\mathrm{UH}}} s \approx s'$. Therefore, $T \vdash_{\mathcal{L}}^{GFB} t \approx t'$. □

Corollary 3.16. *Let $\mathcal{L}$ be a specifiable* HEL *and $T \in \mathrm{Th}(\mathcal{L})$. Then $\mathcal{GFB}(T)$ is a theory of $\mathcal{L}^{\mathrm{UH}}$.*

Theorem 3.17. *Let $\mathcal{L}$ be a specifiable* HEL *and $T \in \mathrm{Th}(\mathcal{L})$. Then*

(1) *$\mathcal{GFB}(T)_V = T_V$, for all $V \in$* VIS.
(2) *$\mathcal{GFB}(T) \subseteq \Omega(T)$.*

Proof. Obviously, $T \subseteq \mathcal{GFB}(T)_{\mathrm{VIS}}$. To prove the other inclusion, suppose that $(t, t') \in \mathcal{GFB}(T) \cap \mathrm{Te}^2_{\Sigma}(X)_{\mathrm{VIS}}$. Since $T \cup \{t \approx t'\} \vdash_{\mathcal{L}^{\mathrm{UH}}} t \approx t'$, then by definition of $\mathcal{GFB}(T)$, $T \vdash_{\mathcal{L}^{\mathrm{UH}}} t \approx t'$. Therefore, since $T \vdash_{\mathcal{L}} t \approx t'$ iff $T \vdash_{\mathcal{L}^{\mathrm{UH}}} t \approx t'$, for any $t, t' \in \mathrm{Te}_{\Sigma}(X)_{\mathrm{VIS}}$, we have $T \vdash_{\mathcal{L}} t \approx t'$, i.e. $t \approx t' \in T$.

Since $\mathcal{GFB}(T)$ is a congruence which coincides with T in the visible part and $\Omega(T)$ is the largest congruence with this property, we have that $\mathcal{GFB}(T) \subseteq \Omega(T)$. □

The following theorem is one of the main results of this section.

Theorem 3.18. *Let $\mathcal{L}$ be a specifiable* HEL *and $T \in \mathrm{Th}(\mathcal{L})$. Then*

$$\Omega(T) = \mathcal{GFB}(T)$$

Proof. Suppose that $t \equiv t'\ (\Omega(T))$. Let $s, s' \in \mathrm{Te}_{\Sigma}(X)_{\mathrm{VIS}}$ and $G \in \mathrm{Th}(\mathcal{L}^{\mathrm{UH}})$, such that $T \subseteq G$. Suppose that

$$G \cup \{t \approx t'\} \vdash_{\mathcal{L}^{\mathrm{UH}}} s \approx s' \tag{10}$$

Since $T \subseteq G_{\mathrm{VIS}}$ and the HEL logics are monotonic (see Corollary 3.7), $\Omega(T) \subseteq \Omega(G_{\mathrm{VIS}})$. Hence, $t \equiv t'\ (\Omega(G_{\mathrm{VIS}}))$. Since $G \subseteq \Omega(G_{\mathrm{VIS}})$ (see Corollary 3.6), (10) implies $s \equiv s'\ (\Omega(G_{\mathrm{VIS}}))$. Moreover, as $s \approx s'$ is visible, $s \approx s' \in G_{\mathrm{VIS}}$. So, $G_{\mathrm{VIS}} \vdash_{\mathcal{L}^{\mathrm{UH}}} s \approx s'$. Since the visible $\mathcal{L}$-consequences and the $\mathcal{L}^{\mathrm{UH}}$-consequences of a set of visible equations are the same, we have $G \vdash_{\mathcal{L}} s \approx s'$. That is, $T \vdash_{\mathcal{L}}^{GFB} t \approx t'$.

The other inclusion is Part 2 of Theorem 3.17. □

3.2. **Strict equational** HEL. In this section we define a simpler notion of behavioral consequence of a theory -*formal behavioral consequence of* T. It considers only the theory T instead of considering all theories of $\mathcal{L}^{\mathrm{UH}}$ that contain T. For strict equational HEL's, we obtain simpler results.

First we define a relation on the term algebra that will play an important role in the sequel. A similar relation was already considered by Leavens and Pigozzi in the context of equational reasoning with subtyping (cf. [33]). The difference is that here we do not restrict the type of the arguments of the term r and we do not consider the substitution instances of the axioms.

Definition 3.19. Let E be a set of equations and $\widetilde{E} = \{t' \approx t : t \approx t' \in E\}$. We define the SORT-sorted relation $\equiv_E$ in the following way:

for each sort S, $t\,(\equiv_E)_S\,t'$ iff there is a term $r(z\!:\!S, y_1\!:\!T_1, \ldots, y_m\!:\!T_m)$ and an equation $e(x_1\!:\!S_1, \ldots, x_n\!:\!S_n) \approx e'(x_1\!:\!S_1, \ldots, x_n\!:\!S_n) \in E \cup \widetilde{E}$ of type S such that $t = r(e(x_1, \cdots, x_n), y_1, \ldots, y_m)$ and $t' = r(e'(x_1, \ldots, x_n), y_1, \ldots, y_m)$.

Finally, we define $\equiv^*_E$ as the reflexive, transitive closure of $\equiv_E$.

In the sequel we will use the following immediate consequence of the definition of $\equiv_E^*$:

Lemma 3.20. *$t \equiv_E^* t'$ iff there are terms $s_1, \dots, s_n$ such that $t = s_1 \equiv_E \cdots \equiv_E s_i \equiv_E s_{i+1} \equiv_E \cdots \equiv_E s_n = t'$.*

By reformulating Lemma 2.21 of [33] we obtain the following characterization of equational consequence:

Lemma 3.21. *Let $\mathcal{L}$ be a strict equational* HEL *with set of equations E. Let F be a set of equations and $t \approx t'$ an equation. Then*

$$F \vdash_{\mathcal{L}^{\mathrm{UH}}} t \approx t' \Leftrightarrow t\,(\equiv_{F\,\cup\,E^{\mathcal{L}}})^*\,t', \tag{11}$$

where $E^{\mathcal{L}}$ is the set of all substitution instances of equations in E.

Proof. [$\Leftarrow$] Suppose that $t\,(\equiv_{F\,\cup\,E^{\mathcal{L}}})^*\,t'$. Then, there are terms $s_0, \dots s_n$ such that $t = s_0 \equiv_{F\,\cup\,E^{\mathcal{L}}} s_1 \equiv_{F\,\cup\,E^{\mathcal{L}}} \cdots s_i \equiv_{F\,\cup\,E^{\mathcal{L}}} s_{i+1} \cdots \equiv_{F\,\cup\,E^{\mathcal{L}}} s_n = t'$. So, for each $i \leq n$, there is a S-context $r(z{:}S, y_1, \dots, y_m)$ and an equation $e(x_1, \dots, x_n) \approx e'(x_1, \dots, x_n) \in (F \cup E^{\mathcal{L}}) \cup (\widetilde{F} \cup \widetilde{E^{\mathcal{L}}})$ such that:

- $s_i = r(e(x_1, \dots, x_n), y_1, \dots, y_m)$ and
- $s_{i+1} = r(e'(x_1, \cdots, x_n), y_1, \dots, y_m)$.

If $e \approx e' \in E^{\mathcal{L}} \cup \widetilde{E}^{\mathcal{L}}$ then obviously $F \vdash_{\mathcal{L}^{\mathrm{UH}}} s_i \approx s_{i+1}$. Otherwise, by the congruence inference rule, $F \vdash_{\mathcal{L}^{\mathrm{UH}}} s_i \approx s_{i+1}$. Therefore, by transitivity, we obtain $F \vdash_{\mathcal{L}^{\mathrm{UH}}} t \approx t'$.

[$\Rightarrow$] We are going to prove that all equations in F and all substitution instances of the extralogical axioms are in $(\equiv_{F\,\cup\,E^{\mathcal{L}}})^*$ and $(\equiv_{F\,\cup\,E^{\mathcal{L}}})^*$ is closed under equation deduction. It is clear that all substitution instances of the equations in E and all equations in F, considered as ordered pairs, are in $(\equiv_{F\,\cup\,E^{\mathcal{L}}})^*$ (by considering the context $r = z{:}S$). By definition, $(\equiv_{F\,\cup\,E^{\mathcal{L}}})^*$ is obviously closed under the rules of reflexivity, symmetry and transitivity. To show that $\equiv_{F\,\cup\,E^{\mathcal{L}}}$ is closed under the congruence rule, let g be an operation symbol of type $S_1, \dots, S_n \to S$ and $t_1, \dots, t_n$ and $t'_1, \dots, t'_n$ of type $S_1, \dots, S_n$ respectively. Suppose that $t_i \equiv_{F\,\cup\,E^{\mathcal{L}}} t'_i$, for all $i \leq n$. To prove that $g(t'_1, \cdots, t'_{i-1}, t'_i, \dots, t'_n) \equiv_{F\,\cup\,E^{\mathcal{L}}} g(t_1, \dots, t_i, t_{i+1}, \dots, t_n)$ it is enough to prove that

$$g(t'_1, \dots, t'_{i-1}, t_i, \dots, t_n) \equiv_{F\,\cup\,E^{\mathcal{L}}} g(t'_1, \dots, t'_i, t_{i+1}, \dots, t_n), \text{ for } i \leq n.$$

We know that $t_i = r(e(x_1, \dots, x_n), y_1, \dots, y_m)$ and $t'_i = r(e'(x_1, \dots, x_n), y_1, \dots, y_m)$, for some S_i-context r and some $e(x_1, \dots, x_n) \approx e'(x_1, \dots, x_n) \in (F \cup E^{\mathcal{L}}) \cup (\widetilde{F} \cup \widetilde{E^{\mathcal{L}}})$.

So,
$g(t'_1, \dots, t'_{i-1}, t_i, \dots, t_n) = g(t'_1, \dots, t'_{i-1}, r(e(x_1, \dots, x_n), y_1, \dots, y_m),\ t_{i+1}, \dots, t_n)$
and
$g(t'_1, \dots, t'_i, t_{i+1}, \dots, t_n) = g(t'_1, \dots, t'_{i-1}, r(e'(x_1, \dots, x_n), y_1, \dots, y_m), t_{i+1}, \dots, t_n)$.
Hence,

$$g(t'_1, \dots, t'_{i-1}, r(e(x_1, \dots, x_n), y_1, \dots, y_m), t_{i+1}, \dots, t_n) = \varphi(r(e(x_1, \dots, x_n), y_1, \dots, y_m), y_1, \dots, y_m)$$

and

$$g(t'_1, \dots, t'_{i-1}, r(e'(x_1, \dots, x_n), y_1, \dots, y_m), t_{i+1}, \dots, t_n) = \varphi(r(e'(x_1, \dots, x_n), y_1, \dots, y_m), y_1, \dots, y_m),$$

where $\varphi(z, y_1, \dots, y_m) = g(t'_1, \dots, t'_{i-1}, z{:}T_i, t_{i+1}, \dots, t_n)$.

Therefore, by the definition of $\equiv_{F \cup E^{\mathcal{L}}}$,

$$g(t'_1, \dots, t'_{i-1}, t_i, \dots, t_n) \equiv_{F \cup E^{\mathcal{L}}} g(t'_1, \dots, t'_i, t_{i+1}, \dots, t_n).$$

Now, we show that $(\equiv_{F \cup E^{\mathcal{L}}})^*$ is also closed under the congruence rule. Suppose now that $t_i \, (\equiv_{F \cup E^{\mathcal{L}}})^* t'_i$, for all $i \leq n$. By definition of $(\equiv_{F \cup E^{\mathcal{L}}})^*$ and reflexivity, for all $i \leq n$, there are terms $s_i^1, \dots, s_i^m$ such that

$$t_i = s_i^1 \equiv_{F \cup E^{\mathcal{L}}} \cdots \equiv_{F \cup E^{\mathcal{L}}} s_i^m = t'_i.$$

Hence, by the previous discussion,

$$g(s_1^1, \dots, s_n^1) \equiv_{F \cup E^{\mathcal{L}}} \cdots \equiv_{F \cup E^{\mathcal{L}}} g(s_1^m, \dots, s_n^m).$$

Finally, by transitivity we get $g(t_1, \dots, t_n) \, (\equiv_{F \cup E^{\mathcal{L}}})^* \, g(t'_1, \dots, t'_n)$. □

Definition 3.22. Let $\mathcal{L}$ be a HEL, $T \in \mathrm{Th}(\mathcal{L})$ and F a set of equations. We say that F is a *formal behavioral consequence* of T, in symbols $T \vdash_{\mathcal{L}}^{FBE} F$, if for every visible equation $s \approx s'$, $T \cup F \vdash_{\mathcal{L}^{\mathrm{UH}}} s \approx s'$ implies that $T \vdash_{\mathcal{L}^{\mathrm{UH}}} s \approx s'$. We say that an equation $t \approx t'$ is a *formal behavioral consequence* of T, if $\{t \approx t'\}$ is and we write $T \vdash_{\mathcal{L}}^{FBE} t \approx t'$.

Similarly to the global behavioral consequence relation, the formal behavioral consequence relation, as a property of a set, is actually a local property.

Theorem 3.23. *Let $\mathcal{L}$ be a strict equational* HEL *and F be a set of equations. Then for every $T \in \mathrm{Th}(\mathcal{L})$,*

$$T \vdash_{\mathcal{L}}^{FBE} F \Leftrightarrow (T \vdash_{\mathcal{L}}^{FBE} t \approx t', \textit{ for all } t \approx t' \in F) \tag{12}$$

Proof. The implication from right to left is straightforward. To prove the other implication, let $s, s' \in (\mathrm{Te}_\Sigma(X))_{\mathrm{VIS}}$ and assume that $T \cup F \vdash_{\mathcal{L}^{\mathrm{UH}}} s \approx s'$. Since $E^{\mathcal{L}} \subseteq T$, by Lemma 3.21, $s \equiv_{T \cup F}^* s'$. Hence, there are $s_1, \cdots, s_n \in \mathrm{Te}_\Sigma(X)$ such that
$s = s_1 \equiv_{T \cup F} \cdots \equiv_{T \cup F} s_m = s'$. Then, for every $i \leq n$, $s_i = r(e(x_1, \dots, x_n), y_1, \dots, y_m)$ and $s_{i+1} = r(e'(x_1, \dots, x_n), y_1, \dots, y_m)$ for some $r(z, y_1, \dots, y_m)$ and $e(x_1, \dots, x_n) \approx e'(x_1, \dots, x_n) \in T \cup F \cup \widetilde{F}$.

If $e(x_1, \dots, x_n) \approx e'(x_1, \dots, x_n) \in T$ then, by the congruence rule,
$T \vdash_{\mathcal{L}^{\mathrm{UH}}} r(e(x_1, \dots, x_n), y_1, \dots, y_m) \approx r(e'(x_1, \dots, x_n), y_1, \dots, y_m)$. I.e., $T \vdash_{\mathcal{L}^{\mathrm{UH}}} s_i \approx s_{i+1}$. Otherwise, $e(x_1, \dots, x_n) \approx e'(x_1, \dots, x_n) \in F \cup \widetilde{F}$. Then, by the congruence rule,
$\{e \approx e'\} \vdash_{\mathcal{L}^{\mathrm{UH}}} s_i \approx s_{i+1}$. So, $T \cup \{e \approx e'\} \vdash_{\mathcal{L}^{\mathrm{UH}}} s_i \approx s_{i+1}$. Since $T \vdash_{\mathcal{L}}^{FBE} e \approx e'$, we get $T \vdash_{\mathcal{L}^{\mathrm{UH}}} s_i \approx s_{i+1}$.

So, for every $i \leq n$, $T \vdash_{\mathcal{L}^{\mathrm{UH}}} s_i \approx s_{i+1}$. Hence, by transitivity rule,
$T \vdash_{\mathcal{L}^{\mathrm{UH}}} s \approx s'$. Therefore $T \vdash_{\mathcal{L}}^{FBE} F$ □

Let $\mathcal{L}$ be a HEL and $T \in \mathrm{Th}(\mathcal{L})$. We define a relation $\mathcal{FB}(T)$ on the term algebra in the following way: for each sort S, $\mathcal{FB}(T)_S$ is the set of all pairs (t, t') of formulas of type S, such that $T \vdash_{\mathcal{L}}^{FBE} t \approx t'$. Thus, $\mathcal{FB}(T) = \langle \mathcal{FB}(T)_S : S \in \mathrm{SORT} \rangle$.

Lemma 3.24. *Let $\mathcal{L}$ be a strict* HEL*. Then for every $T \in \mathrm{Th}(\mathcal{L})$, $\mathcal{GFB}(T) \subseteq \mathcal{FB}(T)$.*

Proof. Let $T \in \mathrm{Th}(\mathcal{L})$. Suppose that $t \approx t' \in \mathcal{GFB}(T)$. Let $s \approx s'$ be a visible equation. Suppose that $T \cup \{t \approx t'\} \vdash_{\mathcal{L}^{\mathrm{UH}}} s \approx s'$. Then, $\mathrm{Cn}_{\mathcal{L}^{\mathrm{UH}}}(T) \cup \{t \approx t'\} \vdash_{\mathcal{L}^{\mathrm{UH}}} s \approx s'$. Hence, by hypothesis, $\mathrm{Cn}_{\mathcal{L}^{\mathrm{UH}}}(T) \vdash_{\mathcal{L}^{\mathrm{UH}}} s \approx s'$. So, $T \vdash_{\mathcal{L}^{\mathrm{UH}}} s \approx s'$. I.e., $(t, t') \in \mathcal{FB}(T)$. □

The following corollary is a consequence of the previous Lemma and Theorem 3.18. It provides a necessary condition for an equation to be a behavioral consequence of a theory.

Corollary 3.25. *Let $\mathcal{L}$ be a strict equational* HEL *and $t \approx t'$ an equation. Then, for every $T \in \mathrm{Th}(\mathcal{L})$*

$$t \equiv t' \; (\Omega(T)) \Rightarrow \; T \vdash_{\mathcal{L}}^{FBE} t \approx t'. \tag{13}$$

An important result of this paper is the converse of (13). First we show that, similarly to the global behavioral consequence, the formal behavioral consequence relation is closed under $\mathcal{L}^{\mathrm{UH}}$-consequences.

Lemma 3.26. *Let $\mathcal{L}$ be a strict equational* HEL, *$T \in \mathrm{Th}(\mathcal{L})$ and F a set of equations. Then $T \vdash_{\mathcal{L}}^{FBE} F$ and $F \vdash_{\mathcal{L}^{\mathrm{UH}}} t \approx t'$ implies that $T \vdash_{\mathcal{L}}^{FBE} t \approx t'$.*

Proof. Let $s \approx s'$ be a visible equation such that $T \cup \{t \approx t'\} \vdash_{\mathcal{L}^{\mathrm{UH}}} s \approx s'$. Then, $T \cup F \vdash_{\mathcal{L}^{\mathrm{UH}}} s \approx s'$. Since $T \vdash_{\mathcal{L}}^{FBE} F$, we have that $T \vdash_{\mathcal{L}^{\mathrm{UH}}} s \approx s'$. Therefore, $T \vdash_{\mathcal{L}}^{FBE} t \approx t'$. □

Therefore,

Corollary 3.27. *Let $\mathcal{L}$ be a strict equational* HEL *and $T \in \mathrm{Th}(\mathcal{L})$. Then $\mathcal{FB}(T)$ is a theory of $\mathcal{L}^{\mathrm{UH}}$.*

Theorem 3.28. *Let $\mathcal{L}$ be a strict equational* HEL *and $T \in \mathrm{Th}(\mathcal{L})$. Then*

(1) *$\mathcal{FB}(T)_V = T_V$, for all $V \in$* VIS.
(2) *$\mathcal{FB}(T) \subseteq \Omega(T)$.*

Proof. Clearly $T_V \subseteq \mathcal{FB}(T)_V$ for all $V \in$ VIS. Let $t \approx t' \in \mathcal{FB}(T)_V$. Then, by definition, for every visible equation $s \approx s'$, $T \cup \{t \approx t'\} \vdash_{\mathcal{L}^{\mathrm{UH}}} s \approx s'$ implies $T \vdash_{\mathcal{L}^{\mathrm{UH}}} s \approx s'$. In particular, $T \vdash_{\mathcal{L}^{\mathrm{UH}}} t \approx t'$, i.e., $T \vdash_{\mathcal{L}} t \approx t'$. Since T is a theory, $t \approx t' \in T$.

$\mathcal{FB}(T) \subseteq \Omega(T)$ is a consequence of Corollary 3.27 and the fact that $\Omega(T)$ is the largest congruence equal to T on the visible part. □

We now have gathered all the necessary results to prove the following characterization of the behavioral consequences of a theory.

Theorem 3.29. *Let $\mathcal{L}$ be a strict equational* HEL *and $T \in \mathrm{Th}(\mathcal{L})$. Then, for every $t, t' \in \mathrm{Te}_\Sigma(X)$,*

$$t \equiv t' \, (\Omega(T)) \Leftrightarrow T \vdash_{\mathcal{L}}^{FBE} t \approx t'. \tag{14}$$

Proof. The direct implication is just Corollary 3.25, and the converse follows from Part (2) of Theorem 3.28. □

4. Conclusion

The discussion of AAL in the many sorted case started in the early nineties. In [9, Section 15] Blok and Pigozzi presented some results for equality-test algebras, where notions such as reduced matrix are dealt with. However, one can credit Don Pigozzi with the seminal ideas presented in 1999 in a course at University of Lisbon, that opened this new area of research. This application of AAL to computer science, namely to behavioral equivalence, produced several results, based on the theory of hidden k-logic. We believe that there is space for further developments. Actually, we are currently trying to use tools from AAL to deal with behavioral transitions, another topic in computer science. Preordered algebra is the natural algebraic framework to specify and reason about transitions. In [17] Diaconescu studies a combination of preordered algebra and hidden algebra, which he calls *hidden preordered algebra*. A new concept appears in this context - *behavioral transition*. Behavioral transitions are already presented in CafeOBJ, however there are still several aspects that need more attention. For instance its methodological aspects remain unexplored. As shown in [17], the coinduction proof method for behavioral equivalence can be extended to proving behavioral transitions. This method is based on the fact that the behavioral preordered algebra congruence on an ordered algebra $(A, \leq)$ is the largest hidden preordered algebra congruence on $(A, \leq)$. This is similar to the case of hidden congruence (behavioral equivalence) vs. Leibniz congruence. Hence, we believe that a model with two filters, one for equations and another for transitions might prove a good fit as a semantics for this computer science paradigm. Raftery's work [47] on ordered algebraizable logics will probably play an important role in our intended application.

References

[1] S. Babenyshev and M. A. Martins. Deduction detachment theorem in hidden k-logics. *J. Log. Comput.*, 24(1): 233-255, 2014.

[2] S. Babenyshev and M. A. Martins. Behavioral equivalence of hidden logics: an abstract algebraic approach. Accepted in *Journal of Applied Logic.*

[3] N. Berreged, A. Bouhoula, and M. Rusinowitch. Observational proofs with critical contexts. In *Fundamental Approaches to Software Engineering, volume 1382 of LNCS, Springer*, pages 38–53. 1998.

[4] M. Bidoit and R. Hennicker. Behavioural theories and the proof of behavioural properties. *Theor. Comput. Sci.*, 165(1):3–55, 1996.

[5] M. Bidoit and R. Hennicker. Observer complete definitions are behaviourally coherent. In *Proc. OBJ/CafeOBJ/Maude Workshop at Formal Methods'99, Toulouse, France, Sep.*, pages 83–94. 1999. Preliminary long version available as Report LSV 99-4.

[6] M. Bidoit, R. Hennicker, and M. Wirsing. Behavioural and abstractor specifications. *Sci. Comput. Program.*, 25(2-3):149–186, 1995.

[7] W. J. Blok and D. Pigozzi. Protoalgebraic logics. *Stud. Logica*, 45:337–369, 1986.

[8] W. J. Blok and D. Pigozzi. *Algebraizable logics*, Memoirs of the American Mathematical Society, vol. 77, 396, 1989.

[9] W. J. Blok and D. Pigozzi. Algebraic semantics for universal Horn logic without equality, Universal algebra and quasigroup theory, Lect. Conf., Jadwisin/Pol. 1989, Res. Expo. Math. 19, 1-56 , 1992.

[10] A. Bouhoula and M. Rusinowitch.Observational proofs by rewriting, *Theor. Comput. Sci.*, 275, (2002), no. 1-2, 675–698.

[11] C. Caleiro, R. Gonçalves and M. A. Martins. Behavioral Algebraization of Logics, *Studia Logica*, 91(1): 63–111, 2009.

[12] J. Czelakowski. Equivalential logics. (I,II). *Stud. Logica*, 40:227–236 and 355–372, 1981.

[13] J. Czelakowski. *Protoalgebraic logics*. Trends in Logic–Studia Logica Library. 10. Dordrecht: Kluwer Academic Publishers, 2001.

[14] J. Czelakowski and R. Jansana. Weakly algebraizable logics, J. *Symb. Log.*, 65, (2000), no. 2, 641–668.

[15] J. Czelakowski and D. Pigozzi. Amalgamation and interpolation in abstract algebraic logic. In *Caicedo, Xavier et al. (ed.), Models, algebras, and proofs. Selected papers of the X Latin American symposium on mathematical logic, Bogotá, Colombia, June 24–29, 1995. New York, NY: Marcel Dekker. Lect. Notes Pure Appl. Math. 203*, pages 187–265. 1999.

[16] L. Descalço and M. A. Martins. On the injectivity of the Leibniz operator. *Bull. Sect. of Logic*, 34(4): 203–211, 2005.

[17] R. Diaconescu. Coinduction for preordered algebra. *Inf. Comput.*, 209(2): 108–117, 2011.

[18] R. Diaconescu and K. Futatsugi. CafeOBJ report: The language, proof techniques, and methodologies for object-oriented algebraic specification. In A. series in Computing, editor, *World Scientific*, volume 6, 1998.

[19] R. Diaconescu and K. Futatsugi. Behavioural coherence in object-oriented algebraic specification. *J. UCS*, 6(1):74–96, 2000.

[20] J. Font and R. Jansana. *A general algebraic semantics for sentential logics*. Lecture Notes in Logic, 7. Berlin: Springer, 1996.

[21] J. Font, R. Jansana and D. Pigozzi. A survey of abstract algebraic logic, *Studia Logica*, 74, (2003), 13–97.

[22] J. Goguen. Types as theories. In *Topology and category theory in computer science*, Oxford Sci. Publ., pages 357–390. Oxford Univ. Press, New York, 1989.

[23] J. Goguen, K. Lin, and G. Roşu. Conditional circular coinductive rewriting with case analysis, *Recent Trends in Algebraic Development Techniques, 16th International Workshop, WADT 2002*, Revised Selected Papers, 216–232, 2002.

[24] J. Goguen and G. Malcolm. Hidden coinduction: Behavioural correctness proofs for objects, *Math. Struct. in Comp. Science*, 9, (1999), no. 3, 287–319.

[25] J. Goguen and G. Malcolm. A hidden agenda, *Theor. Comput. Sci.*, 245, (2000), no. 1, 55–101.

[26] J. Goguen and G. Roşu. Hiding more of hidden algebra., Wing, Jeannette M. (ed.) et al., FM '99. Formal methods.*World congress on Formal methods in the development of computing systems. Toulouse, France, September 20-24, 1999.* Proceedings. In 2 vols. Berlin: Springer; 3-540-66588-9 (vol. 2)). Lect. Notes Comput. Sci. 1708; 1709, 1704-1719 (1999).

[27] E. Goriac, D. Lucanu, and G. Roşu. Automating coinduction with case analysis. In *Formal Methods and Software Engineering - 12th International Conference on Formal Engineering Methods, ICFEM 2010, Shanghai, China, November 17-19, 2010. Proceedings*, pages 220–236, 2010.

[28] R. Hennicker. *Structural specifications with behavioural operators: semantics, proof methods and applications*, Habilitationsschrift, 1997.

[29] R. Hennicker and M. Bidoit. Observational logic. In *Proc. AMAST '98, 7th International Conference on Algebraic Methodology and Software Technology. Lecture Notes in Computer Science, Berlin: Springer*, pag. 263–277. 1999.

[30] B. Herrmann. Equivalential and algebraizable logic. *Stud. Log.*, 57(2-3):419–436, 1996.

[31] J. Goguen, K. Lin and G. Roşu. Circular coinductive rewriting. In *Proceedings, Automated Software Engineering '00 (Grenoble France), IEEE Press*, pages 123–131. September 2000.

[32] G. T. Leavens and D. Pigozzi. A complete algebraic characterization of behavioral subtyping. *Acta Inf.*, 36(8):617–663, 2000.

[33] G. T. Leavens and D. Pigozzi. Equational reasoning with subtypes. Technical Report TR #02-07, Iowa State University, July 2002. Available at `ftp://ftp.cs.iastate.edu/pub/techreports/ TR02-07/TR.pdf`.

[34] G. Malcolm. Behavioural equivalence, bisimulation, and minimal realisation. In *Magne Haveraaen and Olaf Owe and Ole-Johan Dahl (eds.), Recent Trends in Data Type Specifications. 11th Workshop on Specification of Abstract Data Types*. Springer-Verlag Lecture Notes in Computer Science, 1996.

[35] M. A. Martins. *Behavioral reasoning in generalized hidden logics*. PhD thesis, University of Lisbon, 2004.

[36] M. A. Martins. Behavioral institutions and refinements in generalized hidden logics, *J. Univers. Comput. Sci.*, 12(8): 1020–1049, 2006.

[37] M. A. Martins. Closure properties for the class of behavioral models. *Theor. Comput. Sci.*, 379(1–2): 53–83, 2007.

[38] M. A. Martins. On the behavioral equivalence between k-data structures. *Computer Journal*, 51(2): 181–191, 2008.

[39] M. A. Martins and D. Pigozzi. Behavioural reasoning for conditonal equations. *Math. Struct. Comput. Sci.*, 17(5): 1075–1113, 2007.

[40] K. Meinke and J. V. Tucker. *Universal algebra*, *Handbook of logic in computer science*, Vol. 1, Handb. Log. Comput. Sci., vol. 1, Oxford Univ. Press, New York, 1992, pp. 189–411.

[41] B. Moore and G. Roşu. Program verification by coinduction. Technical Report. Available at `http://hdl.handle.net/2142/73177`, University of Illinois, February 2015.

[42] P. Padawitz. Swinging types=functions+relations+transition systems. *Theor. Comput. Sci.*, 243 (2000), no. 1-2, 93–165.

[43] K. M. Pałasińska. *Deductive systems and finite axiomatization properties.* PhD thesis, Iowa State University, 1994.

[44] D. Pigozzi. Equality-test and if-then-else algebras: Axiomatization and specification. *SIAM J. Comput.*, 20(4):766–805, 1991.

[45] D. Pigozzi. *Abstract algebraic logic and the specification of abstract data types*, Preprint, June 1999.

[46] D. Pigozzi. *Abstract algebraic logic*, Encyclopedia of Mathematics, Supplement III (M. Hazewinkel, ed.), Kluwer Academic Publishers, Dordrecht, December 2001, pp. 2–13.

[47] J.G. Raftery. Order algebraizable logics, *Annals of Pure and Applied Logic.* 164 (3), 251–283, 2013.

[48] H. Reichel. Behavioural validity of conditional equations in abstract data types. In *Proceedings of the Viena Conference* (1984).

[49] H. Reichel. An approach to object semantics based on terminal co-algebras. *Math. Struct. Comput. Sci.*, 5(2):129–152, 1995.

[50] G. Roşu. *Hidden Logic.* PhD thesis, University of California, San Diego, 2000.

[51] A. Tarski. Fundamentale begriffe der methodologie der deduktiven wissenschaften. i. *Monatshefte für Mathematik and Physik.* 37:361–404, 1930. English translaction in: Fundamental concepts of the methodology of the deductive systems.

[52] G. Voutsadakis. Categorical Abstract Algebraic Logic: Categorical Algebraization of Equational Logic. *Logic Journal of the IGPL*, 12(4), 313–333, 2004.

[53] G. Voutsadakis. Categorical Abstract Algebraic Logic: Behavioral π-Institutions. *Studia Logica*, 102 (3), 617-646, 2014.

[54] W. Wechler. *Universal algebra for computer scientists.* EATCS Monographs on Theoretical Computer Science. 25. Berlin: Springer-Verlag, 1992.

[55] R. Wójcicki. *Theory of logical caculi. Basic theory of consequence operations.* Synthese Library, 199. Dordrecht: Kluwer Academic Publishers, 1988.

Department of Mathematics, University of Lisbon, Portugal
E-mail address, I. Ferreirim: `imferreirim@ciencias.ulisboa.pt`

CIDMA- Center for R&D in Mathematics and Applications, Department of Mathematics, University of Aveiro, Portugal
E-mail address, M. Martins: `martins@ua.pt`