Available online at www.sciencedirect.com
ScienceDirect
Electronic Notes in Theoretical Computer Science
ELSEVIER
Electronic Notes in Theoretical Computer Science 259 (2009) 105–121
www.elsevier.com/locate/entcs

Refinement by Interpretation in a General Setting

Manuel A. Martins[1]

Dep. Mathematics, Aveiro University, Aveiro, Portugal

Alexandre Madeira[2]

CCTC, Minho University and Dep. Mathematics, Aveiro University and Critical Software S.A., Portugal

Luis S. Barbosa[3]

Dep. Informatics & CCTC, Minho University, Portugal

Abstract

Refinement by interpretation replaces signature morphisms by logic interpretations as a means to translate specifications and witness refinements. The approach was recently introduced by the authors [13] in the context of equational specifications, in order to capture a number of relevant transformations in software design, reuse and adaptation. This paper goes a step forward and discusses the generalization of this idea to deductive systems of arbitrary dimension. This makes possible, for example, to refine sentential into equational specifications and the latter into modal ones. Moreover, the restriction to logics with finitary consequence relations is dropped which results in increased flexibility along the software development process.

Keywords: Refinement; algebraic specification; logic interpretation.

1 Introduction

In the framework of algebraic specification [17,11,18,15,19], *signature morphisms* are traditionally used to translate specifications and, in particular, to witness refinements. This enables renaming, adding, removing and grouping together various signature components which is very useful during the specification and development

[1] Email: martins@ua.pt
[2] Email: madeira@ua.pt
[3] Email: lsb@di.uminho.pt

doi:10.1016/j.entcs.2009.12.020

processes. In a number of situations, however, transformations based in signature morphisms are too rigid to be useful. This is the case in the context of software reuse. But also the emergence of new computing paradigms entails the need for more flexible approaches to what is taken as a valid transformation of specifications (see, for example, [4]).

In a recent paper [13] the authors introduced an alternative approach to refinement of equational specifications in which signature morphisms are replaced by *logic interpretations*. Intuitively, an interpretation is a logic translation which preserves meaning. Originally defined in the area of algebraic logic, in particular as a tool for studying equivalence semantics (see, *e.g.*, [2,1,3,5]), the notion proved effective to capture a number of transformations difficult to deal with in classical terms. Examples include *data encapsulation* and the *decomposition of operations into atomic transactions*.

If there exists a translation τ interpreting a specification SP such that $SP \models \xi \Rightarrow SP' \models \tau(\xi)$ for all relevant conditional equations ξ, we say that SP' *refines the specification* SP *via the interpretation* τ. The following example, adapted from [13], illustrates the approach.

Example 1 *Consider the following fragment of a specification of a bank accoun management system whose signature* Σ_1 *is defined by*

sorts

Ac;
Int;

ops

$bal : Ac \to Int$;
$cred : Ac \times Int \to Ac$
$deb : Ac \times Int \to Ac$

involving account deposits (operation cred), withdrawals (deb) and a balance query (bal). The specification is given as the axiomatic extension of the free equational logic EQ_{Σ_1} *and the traditional specification of* integers*:*

spec BAMS $=$ *enrich* EQ_{Σ_1} *and* INT *with*
axioms
$\langle bal(cred(x, n)), bal(x) + n\rangle$;
$\langle bal(deb(x, n)), bal(x) + (-n)\rangle$.

where the pair of terms $\langle t, t'\rangle$ *represents equation* $t \approx t'$*, a kind of representation discussed latter in this paper.*

Consider, now, an implementation where transactions operations affecting account balances are previously validated. This is achieved through a signature Σ_2 *which extends* Σ_1 *with a new operation symbol*

ops

$\ldots$
$val : Ac \to Ac$

and axioms

spec BAMS2 $=$ *enrich* EQ_{Σ_2} *and* INT *with*
axioms

$\ldots$
$\langle bal(val(cred(x,n)), bal(x) + n\rangle;$
$\langle bal(val(deb(x,n)), bal(x) + (-n)\rangle.$

Clearly,

$$\tau : Eq(\Sigma_1) \longrightarrow Eq(\Sigma_2) \quad = \quad \{\langle \mathsf{op}(x), y\rangle \mapsto \{\langle val(\mathsf{op}(x)), y\rangle\} |\ \mathsf{op} \in \mathsf{userOp}\}$$

where $\mathsf{userOp} = \{cred, deb\}$*, is the required interpretation.*

Can this approach be generalized to *arbitrary* logic systems? What properties will remain valid? What can be expected from such a generalization relevant to the pragmatics of algebraic specification? Such are the questions addressed in the present paper, motivated by a well-known engineering concern: often changes in the application requirements enforce change in the underlying specification logics. The envisaged generalization resorts to a notion of *k-dimensional* logic to represent arbitrary deductive systems. Refinement by interpretation in such a general setting enables, for example, to refine sentential into equational specifications and into modal ones.

This observation answers our last question: the paper's contribution is placed at the meta-level, addressing refinement across different specification logics. The example above, as well as all the results in [13], comes simply from a reduction to the equational setting. How this is achieved is discussed in the following sections. Section 2 introduces *k-dimensional* deductive systems and their semantics, following [1] and paving the way to the formulation of refinement by interpretation in such a general setting in sections 3 and 4. Additionally, section 4 presents a number of examples and discusses further properties of this notion of refinement. Finally, section 5 concludes and suggests some problems deserving further attention.

2 Background

2.1 k-dimensional deductive systems

An *equation*, usually represented by a formal expression $t \approx t'$, can be regarded as a pair of formulas $\langle t, t'\rangle$. This, in turn, is an instance of a binary predicate standing for the equality of two formulas. Similarly, a unary predicate representing the assertion of a formula is enough to represent a proposition. In general, adding a k-ary predicate to a strict universal Horn theory without equality, gives rise to a representation of a k-dimensional logic, thus providing a suitable context to deal si-

multaneously with different specification logics (*e.g.*, assertional, equational, modal, ...).

The syntactic support is that of a k-term for any nonzero natural number k: A *k-term* of sort s over a signature Σ is a sequence of k Σ-terms all of the same sort s. We indicate k-terms by overlining, *i.e.*, $\bar{\varphi}{:}s = \langle \varphi_0{:}s, \ldots, \varphi_{k-1}{:}s \rangle$, or omitting references to sorts, $\bar{\varphi}$. A *k-variable of sort s* is a sequence of k variables all of the same sort s. $\mathrm{Te}^k_\Sigma(X)$ is the sorted set of all k-terms over Σ with variables in X. Thus $\mathrm{Te}^k_\Sigma(X) = \langle (\mathrm{Te}_\Sigma(X))^k_s : s \text{ is a sort in } \Sigma \rangle$.

Let us now introduce some streamlined notation and terminology: if A is a Σ-algebra and $\bar{\varphi}(x_0{:}T_0, \ldots, x_{n-1}{:}T_{n-1})$ is a k-term over Σ and $a_0 \in A_{T_0}, \ldots, a_{n-1} \in A_{T_{n-1}}$, then we denote by $\bar{\varphi}^A(a_0, \ldots, a_{n-1})$ the value $\bar{\varphi}$ takes in A when the variables $x_0, \ldots, x_{n-1}$ are interpreted respectively by $a_0, \ldots, a_{n-1}$. More precisely, if

$$\bar{\varphi}(x_0, \ldots, x_{n-1}) = \langle \varphi_0(x_0, \ldots, x_{n-1}), \ldots, \varphi_{k-1}(x_0, \ldots, x_{n-1}) \rangle,$$

then $\bar{\varphi}^A(a_0, \ldots, a_{n-1}) = h(\bar{\varphi}) := \langle h(\varphi_0), \ldots, h(\varphi_{k-1}) \rangle$, where h is any homomorphism from $\mathrm{Te}_\Sigma(X)$ to A such that $h(x_i) = a_i$ for all $i < n$.

Let $\mathrm{Va} = \langle \mathrm{Va}_s \rangle_{s \in S}$ be an arbitrary but fixed family of countably infinite disjoint sets Va_s of variables of sort $s \in S$. In the sequel, we assume Va fixed for every set of sorts S. As it is usually adopted in logical frameworks we will refer to formulas (k-formulas) as a synonymous of terms (k-terms respectively). For each nonzero natural number k, given a sorted signature Σ, a *k-formula* of sort s over Σ is an member of $(\mathrm{Te}^k_\Sigma(\mathrm{Va}))_s$. The set of all k-formulas will be denoted by $\mathrm{Fm}^k(\Sigma)$. Also note that an S-sorted subset Γ of k-formulas will be identified with the unsorted set $\bigcup_{s \in S} \Gamma_s$, which allows writing $\bar{\varphi} \in \Gamma$ to mean that $\bar{\varphi} \in \Gamma_s$, for some sort s. A set $\Gamma \subseteq \mathrm{Fm}^k(\Sigma)$ is said to be *globally finite* when Γ_s is a finite set for each sort s of Σ and $\Gamma_s = \emptyset$ except for a finite number of them, *i.e.*, when $\bigcup_{s \in S} \Gamma_s$ is finite. In this setting a k-dimensional deductive system, or a k-logic, for short, is defined as a special consequence relation on the set of k-formulas, independently of any specific choice of axioms and rules of inference. More precisely, as a substitution invariant consequence relation on the set of k-formulas. Formally,

Definition 2.1 A *k-dimensional deductive system* is a pair $\mathcal{L} = \langle \Sigma, \vdash_{\mathcal{L}} \rangle$, where Σ is sorted signature and $\vdash_{\mathcal{L}} \subseteq \mathcal{P}(\mathrm{Fm}^k(\Sigma)) \times \mathrm{Fm}^k(\Sigma)$ is a relation that satisfies for all $\Gamma \cup \Delta \cup \{\bar{\gamma}, \bar{\varphi}\} \subseteq \mathrm{Fm}^k(\Sigma)$ 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.

A k-logic is *specifiable* if $\vdash_{\mathcal{L}}$ is finitary, *i.e.*, if $\Gamma \vdash_{\mathcal{L}} \bar{\varphi}$ implies $\Delta \vdash_{\mathcal{L}} \bar{\varphi}$ for some globally finite subset Δ of Γ. The relation $\vdash_{\mathcal{L}}$ is called *the consequence relation of* $\mathcal{L}$; when $\mathcal{L}$ is clear from the context we simple write $\vdash$.

It is easy to see that, for any $\Gamma \cup \Delta \cup \{\bar{\gamma}, \bar{\varphi}\} \subseteq \mathrm{Fm}^k(\Sigma)$, it follows that

$$\Gamma \vdash \bar{\gamma} \text{ and } \Gamma \subseteq \Delta \text{ implies } \Delta \vdash \bar{\gamma}.$$

Every consequence relation $\vdash$ has a natural extension to a relation, also denoted by $\vdash$, between sets of k-formulas; 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 $\Gamma \cup \{\bar{\varphi}_0, \ldots, \bar{\varphi}_{n-1}\} \vdash \bar{\varphi}$ and $\Gamma_0 \cup \cdots \cup \Gamma_{n-1} \vdash \bar{\varphi}$ by $\Gamma, \bar{\varphi}_0, \ldots, \bar{\varphi}_{n-1} \vdash \bar{\varphi}$ and $\Gamma_0, \ldots, \Gamma_{n-1} \vdash \bar{\varphi}$ respectively.

Let $\mathcal{L}$ be a (not necessarily specifiable) k-logic. By a *theorem* of $\mathcal{L}$ we mean a k-formula $\bar{\varphi}$ such that $\vdash_{\mathcal{L}} \bar{\varphi}$, *i.e.*, $\emptyset \vdash_{\mathcal{L}} \bar{\varphi}$. The set of all theorems is denoted by $\mathrm{Thm}(\mathcal{L})$. By an inference rule we mean a pair $\langle \Gamma, \bar{\varphi} \rangle$ with Γ a global finite set of k-formulas and $\bar{\varphi}$ a k-formula, usually we represent an inference rule $\langle \Gamma, \bar{\varphi}_n \rangle$ in the general form

$$\frac{\bar{\varphi}_0, \ldots, \bar{\varphi}_{n-1}}{\bar{\varphi}_n}, \tag{1}$$

where $\Gamma = \{\bar{\varphi}_0, \ldots, \bar{\varphi}_{n-1}\}$. A rule such as (1) is said to be a *derivable rule* of $\mathcal{L}$ if $\{\bar{\varphi}_0, \ldots, \bar{\varphi}_{n-1}\} \vdash_{\mathcal{L}} \bar{\varphi}_n$. A set of k-formulas T closed under the consequence relation, *i.e.*, $T \vdash_{\mathcal{L}} \bar{\varphi}$ implies $\bar{\varphi} \in T$, is called a *theory* of $\mathcal{L}$. 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. Given any set of k-formulas Γ, the set of all consequences of Γ, in symbols $\mathrm{Cn}_{\mathcal{L}}(\Gamma)$, is the smallest theory that contains Γ. It is easy to see that $\mathrm{Cn}_{\mathcal{L}}(\Gamma) = \{\, \bar{\varphi} \in \mathrm{Fm}^k(\Sigma) : \Gamma \vdash_{\mathcal{L}} \bar{\varphi}\}$. Often, a specifiable k-logic is presented in the so called Hilbert style, *i.e.*, by a set of axioms (k-formulas) and inference rules. We say that a k-formula $\bar{\psi}$ is *directly derivable* from a set Γ of k-formulas by a rule such as (1) if there is a substitution $h : \mathrm{Va} \to \mathrm{Fm}^k(\Sigma)$ such that $h(\bar{\varphi}_n) = \bar{\psi}$ and $h(\bar{\varphi}_0), \ldots, h(\bar{\varphi}_{n-1}) \in \Gamma$.

Given a set AX of k-formulas 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 $\Gamma \vdash_{\mathrm{AX,IR}} \bar{\varphi}$ if there is a finite sequence of k-formulas, $\bar{\psi}_0, \ldots, \bar{\psi}_{n-1}$ such that $\bar{\psi}_{n-1} = \bar{\psi}$, and for each $i < n$ one of the following conditions holds:

- $\bar{\psi}_i \in \Gamma$;
- $\bar{\psi}_i$ is a substitution instance of a k-formula in AX;
- $\bar{\psi}_i$ 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 k-logic. Moreover, a 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 k-formulas $\bar{\psi}$ and any set Γ of k-formulas, $\Gamma \vdash_{\mathcal{L}} \bar{\psi}$ iff $\Gamma \vdash_{\mathrm{AX,IR}} \bar{\varphi}$. Hence we will present our examples of specifiable logics by their set of axioms and inference rules. If $\mathcal{L} = \langle \Sigma, \vdash_{\mathrm{AX,IR}} \rangle$, for some sets AX and IR with $|\mathrm{AX} \cup \mathrm{IR}| < \omega$, we say that $\mathcal{L}$ is *finitely aximatizable*. A k-logic $\mathcal{L}' = \langle \Sigma, \vdash_{\mathcal{L}'} \rangle$ is an *extension* of the k-logic $\mathcal{L} = \langle \Sigma, \vdash_{\mathcal{L}} \rangle$ if, $\Gamma \vdash_{\mathcal{L}'} \bar{\varphi}$ whenever $\Gamma \vdash_{\mathcal{L}} \bar{\varphi}$ for all $\Gamma \cup \{\varphi\} \subseteq \mathrm{Fm}^k(\Sigma)$ (*i.e.*$\vdash_{\mathcal{L}} \subseteq \vdash'_{\mathcal{L}}$). A k-logic $\mathcal{L}'$ is *an extension by axioms and rules of* an specifiable k-logic $\mathcal{L}$ if it can be axiomatized by adding axioms and inference rules to the axioms and rules of some axiomatization of $\mathcal{L}$.

2.2 k-data structures

As discussed in [1], the semantics for arbitrary k-logics needs to go beyond the usual algebraic structures, resorting to algebras endowed with a set of k-tuples. Formally, a *k-data structure over* a signature Σ is a pair $\mathcal{A} = \langle A, F\rangle$ where A is a Σ-algebra and F is just a subset of A^k. The set F, of designated elements of A, can be regarded as the set of truth values on A: a formula holds if its interpretation is one of such elements. This is why F is called a *filter*: a filter for a deductive system representing the constructive propositional calculus, on a Boolean algebra is just a familiar, Boolean filter.

Let $\mathcal{A} = \langle A, F\rangle$ be a k-data structure. A k-formula $\bar{\varphi}{:}V$ is said to be a *semantic consequence* in $\mathcal{A}$ of a set of k-formulas Γ, in symbols $\Gamma \models_{\mathcal{A}} \bar{\varphi}$, if, for every assignment $h : \mathrm{Va} \to A$, $h(\bar{\varphi}) \in F_V$ whenever $h(\bar{\psi}) \in F_W$ for every $\bar{\psi}{:}W \in \Gamma$. A k-formula $\bar{\varphi}$ 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 (1) is a *validity*, or a *valid rule*, of $\mathcal{A}$, and conversely $\mathcal{A}$ is a *model* of the rule, if $\{\bar{\varphi}_0, \ldots, \bar{\varphi}_{n-1}\} \models_{\mathcal{A}} \bar{\varphi}_n$. A formula $\bar{\varphi}$ is a *semantic consequence* of a set of k-formulas 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 not always specifiable.

Similarly, a k-formula or rule is a *validity 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 k-logic $\mathcal{L}$ if every consequence of $\mathcal{L}$ is a semantic consequence of $\mathcal{A}$, *i.e.*, $\Gamma \vdash_{\mathcal{L}} \bar{\varphi}$ always implies $\Gamma \models_{\mathcal{A}} \bar{\varphi}$. The special models whose underlying algebra is the formula algebra, i.e., of the form $\langle \mathrm{Fm}^k(\Sigma), T\rangle$, 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 k-logic, then $\mathcal{A}$ is a model of $\mathcal{L}$ iff every axiom and rule of inference is a validity of $\mathcal{A}$.

A classe of k-data structures $\mathcal{M}$ is a *data structure semantics* of $\mathcal{L}$ if $\vdash_{\mathcal{L}} = \models_{\mathcal{M}}$. The classe of all models of $\mathcal{L}$ forms a data structure semantics of $\mathcal{L}$. This fact is expressed in the so called Completeness Theorem, i.e., for any k-logic $\mathcal{L}$, $\Gamma \vdash_{\mathcal{L}} \bar{\varphi}$ iff $\Gamma \models_{\mathrm{Mod}(\mathcal{L})} \bar{\varphi}$, for every $\Gamma \cup \{\bar{\varphi}\} \subseteq \mathrm{Fm}^k(\Sigma)$ (cf. [14]).

2.3 The case of equational logic

In the sequel these ideas are instantiated for the equational case. Recall that, as mentioned in section 1, a Σ-*equation* can be represented as a pair of formulas $\langle t, t'\rangle$ and the set of all equations over Va as $\mathrm{Fm}^2(\Sigma)$. Similar a Σ-*conditional equation* is a pair $\langle \Gamma, e\rangle$ where Γ is a globally finite subset of $\mathrm{Fm}^2(\Sigma)$ and $e \in \mathrm{Fm}^2(\Sigma)$. A conditional equation $\langle \{t_1 \approx t'_1, \ldots, t_n \approx t'_n\}, t \approx t'\rangle$ will be written as $t_1 \approx t'_1 \wedge \cdots \wedge t_n \approx t'_n \to t \approx t'$. An equation may be seen as a conditional equation without premisses, which justifies identifying equation $t \approx t'$ with the conditional equation $\langle \varnothing, t \approx t'\rangle$. The set of all Σ-conditional equations is denoted by Ceq_Σ.

Let $\Gamma \cup \{t \approx t'\} \subseteq \mathrm{Fm}^2(\Sigma)$ and A an algebra. We write $\Gamma \models_A \varphi \approx \psi$ if, for every homomorphism $h : \mathrm{Fm}(\Sigma) \to A$,

$$h(\xi) = h(\eta) \text{ for every } \xi \approx \eta \in \Gamma \text{ implies } h(t) = h(t').$$

If $\Gamma = \emptyset$, we write $\models_A t \approx t'$ instead of $\emptyset \models_A t \approx t'$.

An equation $t \approx t'$ is an *identity* of A if $\models_A t \approx t'$. Similarly, a conditional equation $\xi_0 \approx \eta_0 \wedge \cdots \wedge \xi_{n-1} \approx \eta_{n-1} \to \varphi \approx \psi$ is a *quasi-identity* of A if $\{\xi_0 \approx \eta_0, \ldots, \xi_{n-1} \approx \eta_{n-1}\} \models_A \varphi \approx \psi$.

Let K be a class of Σ-algebras. The (*semantic*) *equational consequence relation* $\models_K$ determined by K is the relation defined between sets of equations and single equations in the following way:

$$\Gamma \models_K t \approx t' \text{ iff, for every } A \in K \text{ we have } \Gamma \models_A t \approx t'.$$

In this case we say that $t \approx t'$ is a K-*consequence* of Γ.

The equational consequence relation $\models_K$ satisfies the conditions of definition 2.1. Hence it constitutes an example of a 2-logic (perhaps the most important one!) which we sometimes designate simply by K. All notions applicable to arbitrary 2-logics, do apply to equational logics.

It can be proved that, if K is a class of Σ-algebras axiomatized by a set of conditional equations then the relation $\models_K$ is specifiable (cf.[3] for the one-sorted case). In this case the relation can be defined in the Hilbert style by considering the set of Σ-equations in Φ together with the reflexivity axioms as the set of axioms, and the Σ-conditional equations in Φ together with the symmetry, transitivity and congruence rules as the inference rules. Actually, any specifiable equational logic over Σ is the natural extension (by axioms and rules) of the (2-dimensional) free equational logic (EQ_Σ) defined by the

Axioms: $\langle x{:}s, x{:}s\rangle$ for each sort s;

Inference rules:

$$(\mathrm{IR}_1) \quad \frac{\langle x{:}s, y{:}s\rangle}{\langle y{:}s, x{:}s\rangle} \quad \text{for each sort } s;$$

$$(\mathrm{IR}_2) \quad \frac{\langle x{:}s, y{:}s\rangle, \langle y{:}s, z{:}s\rangle}{\langle x{:}s, z{:}s\rangle} \quad \text{for each sort } s;$$

$$(\mathrm{IR}_3) \quad \frac{\langle x_0{:}s_0, y_0{:}s_0\rangle, \ldots, \langle x_{n-1}{:}s_{n-1}, y_{n-1}{:}s_{n-1}\rangle}{\langle \sigma(x_0, \ldots, x_{n-1}), \sigma(y_0, \ldots, y_{n-1})\rangle}$$

for each operation symbol $\sigma : s_0, \ldots, s_{n-1} \to s$ in Σ.

Note that EQ_Σ equals $\models_{\mathrm{A}lg(\Sigma)}$, where $Alg(\Sigma)$ is the class of all Σ-algebras.

As usual, an *algebraic specification* SP is a pair $\langle \Sigma, [[SP]]\rangle$ where Σ is a signature, denoted by $Sig(SP)$ and $[[SP]]$ is a class of Σ-algebras. This class of Σ-algebras is called the *model class of* SP, and each $Sig(SP)$-algebra in $[[SP]]$ a *model of* SP. If ξ is the conditional equation $\langle \Gamma, e\rangle$, we write $SP \models \xi$ for $\Gamma \models_{[[SP]]} e$. An algebraic specification SP is X-*flat* if there is a set $\Phi \subseteq \mathrm{Ceq}_\Sigma(X)$ such that $[[SP]] = \{A \in Alg(\Sigma) | A \models \Phi\}$. We represent an axiomatised specification $SP = \langle \Sigma, [[SP]]\rangle$ by a

pair $SP = \langle \Sigma, \Phi \rangle$ omitting explicit reference to variables X; X is assumed to be a set of variables for Σ such that $\Phi \subseteq \mathrm{Ceq}_\Sigma(X)$ and $[[SP]] = \{A \in Alg(\Sigma) | A \models \Phi\}$. When Φ is a set of equations, the specification $SP = \langle \Sigma, \Phi \rangle$ is called an *equational specification*.

A class K of Σ-algebras that satisfies a given set of equations is called a *variety*. A variety can be characterized as a nonempty class K of Σ-algebras which is closed under homomorphic images, subalgebras and direct products. This result due to Birkhoff is very useful to show that a given algebraic specification is not flat.

3 Translations and Interpretations

3.1 Translations

A number of notions of translation between logical systems have been proposed in the literature (see, for example, [7,6,1,12]). In the sequel we adopt the following definition, assuming that all sets of variables are locally countable infinite.

Definition 3.1 [Translation] Let Σ and Σ' be two signatures. A (k,l)-*translation from* Σ *to* Σ' is a globally finite $S - S'$-sorted *multi-function* from $\mathrm{Fm}^k(\Sigma)$ to $\mathrm{Fm}^l(\Sigma')$, i.e., a $S - S'$-sorted *multi-function* such for any $\bar{\varphi} \in \mathrm{Fm}^k(\Sigma)$, $\tau(\bar{\varphi})$ be a globally finite set.

Whenever $\Sigma = \Sigma'$, τ is said a *self translation of* Σ. In this case, we say that τ *commutes with substitutions* if for every substitution σ and every formula $\bar{\varphi} \in \mathrm{Fm}^k(\Sigma)$ $\tau(\sigma(\bar{\varphi})) = \sigma(\tau(\bar{\varphi}))$. In this case, we present the translation just by giving, for each sort s, the image $\tau_s(\bar{x} : s)$ of a k-variable $\bar{x} : s$ (see Example 2). Given a (k,l)-translation τ and an inference rule $\xi = \langle \Gamma, \bar{\varphi} \rangle$, we write $\tau(\xi)$ for the set of inference rules $\{\langle \tau(\Gamma), \bar{\psi} \rangle : \bar{\psi} \in \tau(\bar{\varphi})\}$.

We say that a self (k,l)-translation τ is *schematic* if there is a S-sorted set Δ of l-formulas, where for each s, $\Delta_s(\bar{x})$ is a set of l-formulas over Σ' in the k-variable $\langle x_0 : s, \ldots, x_{n-1} : s \rangle$ such that, for any $\bar{\varphi} \in \mathrm{Fm}^k(\Sigma)_s$ $\tau_s(\bar{\varphi}) = \Delta_s(\varphi_0, \ldots, \varphi_{k-1})$.We say that a (k,l)-translation is *functional* if the image of each k-formula is a singleton. The schematic (2,2)-translations were used to translate algebraic specifications in the context of the *refinements via translations* [10].

3.2 Interpretations

Defined as a multi-function, in definition 3.1, a translation maps a formula into a set of formulas. This is exactly what makes translations interesting to establish relationships between specifications and the main source of flexibility of the approach proposed in this paper. Recall that, on the other hand, a signature morphism maps a formula into just another formula.

Not all translations, however, are suitable to base a suitable refinement relation. The following definition singles out the relevant ones:

Definition 3.2 [Interpretation]Let τ be a (k,l)-translation from Σ to Σ'. Let $\mathcal{L}$ be a k-logic over Σ. We say that τ *interprets* $\mathcal{L}$ if there is a l-logic $\mathcal{L}'$ over Σ' such

that, for any $\Gamma \cup \{\bar{\varphi}\} \subseteq \mathrm{Fm}^k(\Sigma)$, $\Gamma \vdash_{\mathcal{L}} \bar{\varphi}$ if and only if $\tau(\Gamma) \vdash_{\mathcal{L}'} \tau(\bar{\varphi})$. In this case we say that τ *interprets* $\mathcal{L}$ *in* $\mathcal{L}'$ and $\mathcal{L}'$ *is a* τ*-interpretation of* $\mathcal{L}$.

It can be proved that interpretations can be composed in the sense that if $\mathcal{L}'$ is a τ-interpretation of $\mathcal{L}$ and $\mathcal{L}''$ is a ρ-interpretation of $\mathcal{L}'$ then $\rho \circ \tau$ interprets $\mathcal{L}$ in $\mathcal{L}''$.

The special case of functional translation with $k = l = 1$, *i.e.*, between sentential languages, has been intensively studied by Feitosa and Ottaviano [7], where the interpretations were called *conservative translation*. Based on their work we have the following sufficient condition for a translation to be an interpretation, quite useful in practice.

Theorem 3.3 *Let τ be a (k,l)-translation from Σ to Σ', $\mathcal{L}$ a k-logic over Σ and $\mathcal{L}'$ a l-logic over Σ'. Suppose that τ is functional and injective. If $\tau(\mathrm{Cn}_{\mathcal{L}}(\Gamma)) = \mathrm{Cn}_{\mathcal{L}'}(\tau(\Gamma))$ for every set of formulas Γ, then τ interprets $\mathcal{L}$ in $\mathcal{L}'$.*

Proof. From the inclusion $\tau(\mathrm{Cn}_{\mathcal{L}}(\Gamma)) \subseteq \mathrm{Cn}_{\mathcal{L}'}(\tau(\Gamma))$ we have that $\Gamma \vdash_{\mathcal{L}} \bar{\varphi} \Rightarrow \tau(\Gamma) \vdash_{\mathcal{L}'} \tau(\bar{\varphi})$. Suppose now that $\tau(\bar{\varphi}) \in \mathrm{Cn}_{\mathcal{L}'}(\tau(\Gamma)) = \tau(\mathrm{Cn}_{\mathcal{L}}(\Gamma))$. Hence there is a $\bar{\psi} \in \mathrm{Cn}_{\mathcal{L}}(\Gamma)$ such that $\tau(\bar{\varphi}) = \tau(\bar{\psi})$. Since τ is injective $\bar{\varphi} = \bar{\psi}$, and so, $\bar{\varphi} \in \mathrm{Cn}_{\mathcal{L}}(\Gamma)$, i.e., $\Gamma \vdash_{\mathcal{L}} \bar{\varphi}$. □

The following are interesting examples of interpretations, relevant to capture the *change of logic* underlying the refinements one is interested in.

Example 2 (CPC **vs. Boolean algebras)** *The equational logic of Boolean algebras $\mathcal{L}_{\mathrm{BA}}$ interprets the classical propositional logic* (CPC)*, both over the one-sorted signature $\Sigma = \{\rightarrow, \wedge, \vee, \neg, \top, \bot\}$, under the self (1,2)-translation $\tau(p) = \{\langle p, \top\rangle\}$. Moreover, the equational logic $\mathcal{L}_{\mathrm{HA}}$, induced by the class of Heyting algebras* HA *also provides an interpretation of* CPC *under the translation $\nu(p) = \{\langle \neg\neg p, \top\rangle\}$. This translation also interprets* CPC *into $\mathcal{L}_{\mathrm{BA}}$ which shows that a interpretation may not to be unique* [3].

Reciprocally, as one would expect, CPC *also interprets $\mathcal{L}_{\mathrm{BA}}$, now under the (2,1)-translation $\rho(\langle p, q\rangle) = \{p \rightarrow q, q \rightarrow p\}$.*

Example 3 (Semilattices into posets) *A semilattice can be regarded either as an algebra or as a partially order structure. Such a duality, often useful in specifications, can be expressed, in a natural way, by an interpretation, actually an equivalence between two 2-logics over the one-sorted signature $\Sigma = \{\wedge\}$ (see* [1]*). Consider the logics*

spec SLV $=$ *enrich* EQ_{Σ} *with*
axioms
$\langle p, p \wedge p\rangle$;
$\langle p \wedge q, q \wedge p\rangle$;
$\langle p \wedge (q \wedge r)$;
$(p \wedge q) \wedge r\rangle$;

and SLP*, the specifiable 2-logic defined by the following axioms and inference rules:*

spec SLP $=$ *enrich* EQ_Σ *with*
axioms
$\langle p, p\rangle$;
$\langle p, p \wedge p\rangle$;
$\langle p \wedge q, p\rangle$;
$\langle p \wedge q, q\rangle$;
inference rules
$\dfrac{\langle x, y\rangle, \langle y, z\rangle}{\langle x, z\rangle}$;
$\dfrac{\langle x_0, y_0\rangle, \langle x_1, y_1\rangle}{\langle x_0 \wedge x_1, y_0 \wedge y_1\rangle}$;

The translation τ defined by the multifunction $\tau(\langle p, q\rangle) = \{\langle p, q\rangle, \langle q, p\rangle\}$ witnesses that SLP *interprets* SLV*.*

3.3 Towards algebraic semantics

There are k-logics to which an algebraic specification can be associated, thus providing an alternative semantics (called *algebraic semantics* in the context of algebraic logic). It is well known [3] that this association is not unique and may not exist.

Definition 3.4 [τ-model] Let τ be a (k, l)-translation from Σ to Σ' and $\mathcal{L}$ a k-logic over Σ. A l-data structure $\mathcal{A}$ is a *τ-model of $\mathcal{L}$* if for any $\Gamma \cup \{\bar{\varphi}\} \subseteq \mathrm{Fm}^k(\Sigma)$, $\Gamma \vdash_{\mathcal{L}} \bar{\varphi}$ implies $\tau(\Gamma) \models_{\mathcal{A}} \tau(\bar{\varphi})$. The classe of all τ-models of $\mathcal{L}$, denoted by $\mathrm{Mod}^\tau(\mathcal{L})$, is called *$\tau$-model class of $\mathcal{L}$.*

As mentioned above, the semantic consequence associated to a class of k-data structures, defined over $\mathrm{Fm}^k(\Sigma)$, is always a k-logic, even if it fails to be specifiable. Hence, $\models_{\mathrm{Mod}^\tau(\mathcal{L})}$ is a logic which we will denote by $\mathcal{L}^\tau$. Furthermore,

Theorem 3.5 *Let τ be a (k, l)-translation from Σ to Σ' and $\mathcal{L}$ a k-logic over Σ. If τ interprets $\mathcal{L}$, then the l-logic $\mathcal{L}^\tau$ interprets $\mathcal{L}$; moreover, it is the τ-interpretation of $\mathcal{L}$ with the largest class of models.*

Proof. Suppose that τ interprets $\mathcal{L}$. Let $\mathcal{L}'$ be a specification that is a τ-interpretation of $\mathcal{L}$. Then for any $\Gamma \cup \{\bar{\varphi}\} \subseteq \mathrm{Fm}^k(\Sigma)$, $\Gamma \vdash_{\mathcal{L}} \bar{\varphi}$ iff $\tau(\Gamma) \vdash_{\mathcal{L}'} \tau(\bar{\varphi})$ iff $\tau(\Gamma) \models_{\mathrm{Mod}(\mathcal{L}')} \tau(\bar{\varphi})$. Hence all models of $\mathcal{L}'$ are τ-models of $\mathcal{L}$. Thus, $\mathrm{Mod}(\mathcal{L}') \subseteq \mathrm{Mod}^\tau(\mathcal{L})$.
So, it is enough to prove that $\mathcal{L}^\tau$ is a τ-interpretation of $\mathcal{L}$. Let $\Gamma \cup \{\bar{\varphi}\} \subseteq \mathrm{Fm}^k(\Sigma)$. It is clear that $\Gamma \vdash_{\mathcal{L}} \bar{\varphi}$ implies $\tau(\Gamma) \vdash_{\mathcal{L}^\tau} \tau(\bar{\varphi})$. Suppose now that $\tau(\Gamma) \vdash_{\mathcal{L}^\tau} \tau(\bar{\varphi})$. Let $\mathcal{L}'$ be a specification that is a τ-interpretation of $\mathcal{L}$ (it exists since τ interprets $\mathcal{L}$). Since, $\mathrm{Mod}(\mathcal{L}') \subseteq \mathrm{Mod}^\tau(\mathcal{L})$, $\tau(\Gamma) \vdash_{\mathcal{L}'} \tau(\bar{\varphi})$. Thus $\Gamma \vdash_{\mathcal{L}} \bar{\varphi}$ because $\mathcal{L}'$ is a τ-interpretation of $\mathcal{L}$. □

Algebraic specifications are the most common way to specify a software system. Let us then explore the special case where τ is a $(k, 2)$-translation. This kind of translations allows us to establish important relationships between a k-logic and an appropriate algebraic specification. Note that a $(k, 2)$-translation maps a k-formula

in a 2-formula. In the remaining of this section we will consider a 2-formula $\langle t, t' \rangle$ as an equation $t \approx t'$. Therefore, let τ be a $(k, 2)$-translation from Σ to Σ' and $\mathcal{L}$ a k-logic. A class K of Σ-algebras is said to be a *τ-algebraic semantics of $\mathcal{L}$* if τ interprets $\mathcal{L}$ in $\models_K$. Based on the notion of τ-model, we define the algebraic specification $SP^{\tau}_{\mathcal{L}}$, over Σ', whose class of algebras is the class of algebraic reducts of the *τ-model of $\mathcal{L}$* having as filter the identity. Specifically,

$$[[SP^{\tau}_{\mathcal{L}}]] = \{A : \langle A, \triangle_A \rangle \text{ is a } \tau\text{-model}\}$$

It can be proved that, given a $(k, 2)$-translation τ from Σ to Σ', and a k-logic $\mathcal{L}$ over Σ, if there is a τ-algebraic semantics of $\mathcal{L}$, then the algebraic specification $SP^{\tau}_{\mathcal{L}}$ is the largest τ-algebraic semantics of $\mathcal{L}$, i.e., with the largest class of models. Moreover, $SP^{\tau}_{\mathcal{L}}$ is finitely axiomatized whenever SP is.

Consider now the following mapping $\tau_{\mathcal{L},K} : \mathrm{Th}(\mathcal{L}) \to \mathrm{Th}(K)$ defined by $\tau_{\mathcal{L},K}(T) = \mathrm{Cn}_K(\tau(T))$, for all $T \in \mathrm{Th}(\mathcal{L})$. Sometimes the algebraic specification $SP^{\tau}_{\mathcal{L}}$ is too wide for our purposes, namely to discuss implementations. The following theorem gives a sufficient and necessary condition for a subclass of $SP^{\tau}_{\mathcal{L}}$ being a τ-algebraic semantics of $\mathcal{L}$. It should be mentioned that similar results are well known for sentential logics [3]. In this paper we reformulate them for k-dimensional and many sorted logics, since they give interesting conditions (sufficient and necessary) for a logical system to have an algebraic semantics.

Lemma 3.6 *Let $\mathcal{L}$ be a logic, τ a self $(k, 2)$-translation of Σ that commutes with substitutions and $K \subseteq [[SP^{\tau}_{\mathcal{L}}]]$. The following conditions are equivalent:*

(i) *K is a τ-algebraic semantics of $\mathcal{L}$.*

(ii) *$\tau_{\mathcal{L},K}$ is injective.*

Proof. Let $T_1, T_2 \in \mathrm{Th}(\mathcal{L})$ and $\bar{\alpha} \in T_1$. Suppose $\tau_{\mathcal{L},K}(T_1) = \tau_{\mathcal{L},K}(T_2)$. We have that $\tau(\bar{\alpha}) \subseteq \tau(T_1) \subseteq \tau_{\mathcal{L},K}(T_1) = \tau_{\mathcal{L},K}(T_2)$, i.e., $\tau(T_2) \models_K \tau(\bar{\alpha})$. Since K is an τ-algebraic semantics of $\mathcal{L}$, we have $T_2 \vdash_{\mathcal{L}} \bar{\alpha}$, i.e., $\bar{\alpha} \in T_2$. Thus $T_1 \subseteq T_2$. In analogous way, we can proved that $T_2 \subseteq T_1$. We conclude that $\tau_{\mathcal{L},K}$ is injective.

Conversely, let $\Gamma \cup \{\bar{\alpha}\} \subseteq \mathrm{Fm}^k(\Sigma)$. Since K is a class of of algebraic reducts of τ-models of $\mathcal{L}$, we have that $\Gamma \vdash_{\mathcal{L}} \bar{\alpha}$ implies $\tau(\Gamma) \models_K \tau(\bar{\alpha})$. Now, suppose $\tau(\Gamma) \models_K \tau(\bar{\alpha})$. Thus, $\mathrm{Cn}_K(\tau(\Gamma)) = \mathrm{Cn}_K(\tau(\Gamma \cup \{\bar{\alpha}\}))$. Since $\Gamma \subseteq \mathrm{Cn}_{\mathcal{L}}(\Gamma)$, we have that $\tau(\Gamma) \subseteq \tau(\mathrm{Cn}_{\mathcal{L}}(\Gamma))$. Thus $\mathrm{Cn}_K(\tau(\Gamma)) \subseteq \mathrm{Cn}_K(\tau(\mathrm{Cn}_s(\Gamma))) = \tau_{\mathcal{L},K}(\mathrm{Cn}_{\mathcal{L}}(\Gamma))$. To prove the reverse inclusion, let $t \approx t' \in \tau_{\mathcal{L},K}(\mathrm{Cn}_{\mathcal{L}}(\Gamma))$. Thus $\{\tau(\xi) : \Gamma \vdash_{\mathcal{L}} \xi\} \models_K t \approx t'$. Again, since K is a class of of algebraic reducts of τ-models of $\mathcal{L}$, for all $t \approx t' \in \mathrm{Fm}^2(\Sigma)$ we have that $\Gamma \vdash_{\mathcal{L}} \xi$ implies $\tau(\Gamma) \models_K \tau(\xi)$. Hence $\tau(\Gamma) \models_K t \approx t'$, i.e., $t \approx t' \in \mathrm{Cn}_K(\tau(\Gamma))$. Therefore, for all $\Gamma \subseteq \mathrm{Fm}^k(\Sigma)$, $\tau_{\mathcal{L},K}(\mathrm{Cn}_{\mathcal{L}}(\Gamma)) = \mathrm{Cn}_K(\tau(\Gamma))$. Thus by these results, we have that $\tau_{\mathcal{L},K}(\mathrm{Cn}_{\mathcal{L}}(\Gamma)) = \tau_{\mathcal{L},K}(\mathrm{Cn}_s(\Gamma \cup \{\bar{\alpha}\}))$. Since $\tau_{\mathcal{L},K}$ is injective, $\mathrm{Cn}_{\mathcal{L}}(\Gamma) = \mathrm{Cn}_{\mathcal{L}}(\Gamma \cup \{\bar{\alpha}\})$, i.e., $\Gamma \vdash_{\mathcal{L}} \bar{\alpha}$. □

Lemma 3.6 and the fact that class $[[SP^{\tau}_{\mathcal{L}}]]$ is a τ-algebraic semantics, entails another important result: if $\mathcal{L}$ has a τ-algebraic semantics, then any extension of $\mathcal{L}$ also has a τ-algebraic semantics, for τ a self $(k, 2)$-translation of Σ commuting with substitutions.

4 Refinement via interpretation

Refinement is a systematic process along which specifications are transformed from an abstract level towards concrete implementations. In each step new requirements can be added (for example, forcing operations to become deterministic), but without denying the properties explicitly stated in the original specification. Refinement proceeds in a stepwise way leading to a chain of specifications

$$SP_0 \rightsquigarrow SP_1 \rightsquigarrow SP_2 \rightsquigarrow \cdots \rightsquigarrow SP_{n-1} \rightsquigarrow SP_n,$$

where for all $1 \leq i \leq n$ $SP_{i-1} \rightsquigarrow SP_i$ means a valid refinement step, entailing $[[SP_i]] \subseteq [[SP_{i-1}]]$. Transitivity of relation $\rightsquigarrow$, often referred to as *vertical composition*, assures that $SP_0 \rightsquigarrow SP_n$.

What counts for a valid refinement step is precisely what is under discussion in this paper. Our starting point is the following syntactic-grounded notion,

Definition 4.1 [(Syntactic) refinement] Let Σ and Σ' be two signatures such that $\Sigma \subseteq \Sigma'$, with identical set of sorts, $\mathcal{L}$ and $\mathcal{L}'$ be two k-logics over Σ and Σ' respectively. We say that $\mathcal{L}'$ is a *(syntactic) refinement of* $\mathcal{L}$, in symbols $\mathcal{L} \rightsquigarrow \mathcal{L}'$, if for any $\Gamma \cup \{\bar{\varphi}\} \subseteq \mathrm{Fm}^k(\Sigma)$,

$$\Gamma \vdash_{\mathcal{L}} \bar{\varphi} \Rightarrow \Gamma \vdash_{\mathcal{L}'} \bar{\varphi}.$$

Note that when $\mathcal{L}$ is specifiable, $\mathcal{L} \rightsquigarrow \mathcal{L}'$ if all the axioms of $\mathcal{L}$ are theorems of $\mathcal{L}'$ and the theories of $\mathcal{L}'$ are compatible with the inference rules of $\mathcal{L}$.

Example 4 *Modal logic* $\mathrm{S5^G}$ *forms a (syntactic) refinement of* CPC. *Consider the modal signature* $\Sigma = \{\rightarrow, \wedge, \vee, \neg, \top, \bot, \Box\}$. *The modal logic* K *is defined as an extension of* CPC *by adding the axiom* $\Box(p \rightarrow q) \rightarrow (\Box p \rightarrow \Box q)$ *and the inference rule* $\dfrac{p}{\Box p}$. *Logic* $\mathrm{S5^G}$*, on the other hand, enriches the signature of* K *with the symbol* $\Diamond$*, and* K *itself with the axioms* $\Box p \rightarrow p$, $\Box p \rightarrow \Box\Box p$ *and* $\Diamond p \rightarrow \Box\Diamond p$ (*cf.* [1]). *Hence, since the signature of both systems contain the signature of* CPC *and their presentations result from the introduction of extra axioms and inference rules to the* CPC *presentation, we have, by the previous fact that* CPC $\rightsquigarrow$ K *and* CPC $\rightsquigarrow$ $\mathrm{S5^G}$ *(actually,* CPC $\rightsquigarrow$ K $\rightsquigarrow$ $\mathrm{S5^G}$*). Hence, refining* CPC *in this way, we acquire expressivity sufficient to express properties over propositions like* it is necessary that ϕ *(by* $\Box\phi$*) and* it is possible that ϕ *(by* $\Diamond\phi$*). This kind of refinement makes possible the accommodation of a new type of requirements, modally expressed, along the refinement process.*

Theorem 4.2 *Let* Σ *be a signature and* $\mathcal{L}$ *and* $\mathcal{L}'$ *two* k*-logics over* Σ*. Then the following conditions are equivalent*

(i) $\mathcal{L} \rightsquigarrow \mathcal{L}'$

(ii) $\mathrm{Mod}(\mathcal{L}') \subseteq \mathrm{Mod}(\mathcal{L})$.

Proof. (i) $\Rightarrow$ (ii). Suppose $\mathcal{L} \rightsquigarrow \mathcal{L}'$. Let $\mathcal{A} \in \mathrm{Mod}(\mathcal{L}')$ and $\Gamma \cup \{\bar{\varphi}\} \subseteq \mathrm{Fm}^k(\Sigma)$. Suppose $\Gamma \vdash_{\mathcal{L}} \bar{\varphi}$. We have by (i) that $\Gamma \vdash_{\mathcal{L}'} \bar{\varphi}$ and hence $\Gamma \vdash_{\mathcal{A}} \bar{\varphi}$. Therefore $\mathcal{A} \in \mathrm{Mod}(\mathcal{L})$.

(ii) $\Rightarrow$ (i). Suppose $\mathrm{Mod}(\mathcal{L}') \subseteq \mathrm{Mod}(\mathcal{L})$. Let $\Gamma \cup \{\bar{\varphi}\} \subseteq \mathrm{Fm}^k(\Sigma)$. Suppose $\Gamma \vdash_{\mathcal{L}} \bar{\varphi}$. Let $\mathcal{A} \in \mathrm{Mod}(\mathcal{L}')$. By (ii) we have $\mathcal{A} \in \mathrm{Mod}(\mathcal{L})$ and hence $\Gamma \vdash_{\mathcal{A}} \bar{\varphi}$. Therefore, by completeness, $\Gamma \vdash_{\mathcal{L}'} \bar{\varphi}$. □

A coarser and more flexible definition of refinement, however, is provided by the notion of logic interpretation. Formally,

Definition 4.3 [Refinement via interpretation] Let $\mathcal{L}$ be a k-logic over Σ and τ a (k,l)-translation from Σ to Σ', which interprets $\mathcal{L}$ and such that the empty set does not belong to its codomain. We say that a l-logic $\mathcal{L}'$ over Σ' *refines the logic* $\mathcal{L}$ *via the interpretation* τ, in symbols $\mathcal{L} \rightharpoonup_{\tau} \mathcal{L}'$, if for any $\Gamma \cup \{\bar{\varphi}\} \subseteq \mathrm{Fm}^k(\Sigma)$,

$$\Gamma \vdash_{\mathcal{L}} \bar{\varphi} \;\Rightarrow\; \tau(\Gamma) \vdash_{\mathcal{L}'} \tau(\bar{\varphi}).$$

The condition that τ has to interpret $\mathcal{L}$ is required in order to have some control under the class of models of the logic $\mathcal{L}'$. In particular this guarantees that $\mathrm{Mod}(\mathcal{L}')$ has to be smaller than SP^{τ}.

The following two examples illustrate the *refinement via interpretation* approach at work.

Example 5 *Any subclass of the class of Boolean algebras induces a τ- refinement of* CPC *with τ the usual (1,2)-translation defined by $\tau(p) = \{\langle p, \top\rangle\}$).*

Example 6 *Consider the fragment of the specification* BAMS*, of a bank account management system, given in Example 1. Suppose we intend to refine this system by imposing that the* balance of each account has to be positive. *Naturally, this cannot be expressed in a (strict) equational logic. However, this situation can be encompassed using our formalization of refinement. Actually, consider now the following 2-logic over Σ:*

spec ORDBAMS $=$ *enrich* INT *with*
axioms
$\langle x{:}s, x{:}s\rangle, s \in \{Ac, int\}$;
$\langle bal(cred(x,n)), bal(x)+n\rangle; \langle bal(x)+n, bal(cred(x,n))\rangle$;
$\langle bal(deb(x,n)), bal(x)+(-n)\rangle; \langle bal(x)+(-n), bal(deb(x,n))\rangle$
inference rules
$$\frac{\langle x{:}Ac, y{:}Ac\rangle}{\langle y{:}Ac, x{:}Ac\rangle};$$
$$\frac{\langle x,y\rangle;\langle w,z\rangle}{\langle x+w, y+z\rangle}; \frac{\langle x,y\rangle}{\langle -y,-x\rangle}$$
$$\frac{\langle x{:}s, y{:}s\rangle; \langle y{:}s, z{:}s\rangle}{\langle x{:}s, z{:}s\rangle}; s \in \{Ac, int\};$$
$$\frac{\langle x,y\rangle}{\langle bal(x), bal(y)\rangle}; \frac{\langle x,y\rangle}{\langle bal(y), bal(x)\rangle}; \frac{\langle x,y\rangle}{\langle cred(x), cred(y)\rangle}; \frac{\langle x,y\rangle}{\langle deb(x), deb(y)\rangle}$$

Let us take a fixed-semantics approach. Consider the fixed-semantics of the above logics by fixing the Int component of the domains as the integer numbers (endowed

with its operations) and fixing the Int component of their filters as the identity relation. In fact, consider the following subclasses of model class of the above 2-logics:

$$\mathrm{BAMS}^{\mathbb{Z}} = \{\langle\langle A_{Ac}, A_{Int}\rangle, \langle F_1, F_2\rangle\rangle \in \mathrm{Mod}(\mathrm{BAMS}) : A_{Int} = \mathbb{Z} \,\&\, F_2 = id_{\mathbb{Z}}\}$$

$$\mathrm{ORDBAMS}^{\mathbb{Z}} = \{\langle\langle A_{Ac}, A_{Int}\rangle, \langle G_1, G_2\rangle\rangle \in \mathrm{Mod}(\mathrm{ORDBAMS}) : A_{Int} = \mathbb{Z} \,\&\, G_2 = \leq\}.$$

Let now τ be the (2,2)-translation defined schematically by $\tau_{Int}(\langle x, y\rangle) = \{\langle x, y\rangle, \langle y, x\rangle\}$ and $\tau_{Ac}(\langle x, y\rangle) = \{\langle x, y\rangle\}$ (the idea behind is that an equation $x \approx y$ of sort Int is translated in the two inequalities $x \leq y$ and $y \leq x$). Intuitively, we can accept that $\models_{\mathrm{ORDBAMS}^{\mathbb{Z}}}$ is a τ-interpretation of $\models_{\mathrm{BAMS}^{\mathbb{Z}}}$. Now, adding the axiom $\langle 0, balance(x)\rangle$ we obtain the announced specification as a τ-refinement of the original one.

Having illustrated some typical applications of the notion of refinement put forward in this paper, it is legitimate to ask now how does it relate to the traditional ones. From the previous theorem we achieve at the following characterization of the refinement via interpretation:

Theorem 4.4 *Let $\mathcal{L}$ and $\mathcal{L}'$ be a k-logic over Σ and l-logic over Σ' respectively. Let τ be a (k, l)-translation from Σ to Σ'. Then the following conditions are equivalent*

(i) $\mathcal{L} \prec_\tau \mathcal{L}'$;

(ii) $\mathcal{L}'$ *is a refinement of some τ-interpretation of $\mathcal{L}$,*
(i.e., there is a l-logic $\mathcal{L}^0$ which τ-interprets $\mathcal{L}$ and $\mathcal{L}^0 \rightsquigarrow \mathcal{L}'$.)

Proof. Suppose $\mathcal{L} \prec_\tau \mathcal{L}'$. Then $\mathrm{Mod}(\mathcal{L}')$ is a subclasse of τ-models of $\mathcal{L}$. By Theorem 3.5 $\mathrm{Mod}(\mathcal{L}') \subseteq \mathrm{Mod}(\mathcal{L}^\tau)$. Therefore, by Theorem 4.2, $\mathcal{L}^\tau \rightsquigarrow \mathcal{L}'$. So, condition (ii) holds for $\mathcal{L}^0 = \mathcal{L}^\tau$.

Suppose now there is a l-logic $\mathcal{L}^0$ which τ-interprets $\mathcal{L}$ and $\mathcal{L}^0 \rightsquigarrow \mathcal{L}'$. Let $\Gamma \cup \{\bar{\varphi}\} \subseteq \mathrm{Fm}^k(\Sigma)$. Then

$$\Gamma \vdash_{\mathcal{L}} \bar{\varphi} \Leftrightarrow \tau(\Gamma) \vdash_{\mathcal{L}^0} \tau(\bar{\varphi}) \Rightarrow \tau(\Gamma) \vdash_{\mathcal{L}'} \tau(\bar{\varphi}).$$

The equivalence holds since τ interprets $\mathcal{L}$ in $\mathcal{L}^0$. The implication holds since $\mathcal{L}^\tau \rightsquigarrow \mathcal{L}'$. Therefore, $\mathcal{L} \prec_\tau \mathcal{L}'$. □

Example 7 *Suppose a requirements specification is provided in the* CPC *specification logic, but one would like to obtain an implementation in which the properties of the system must be shown in a constructive way, for example by resorting to some kind of theorem prover. This entails the need for the specification refactoring within some variant of intuitionist logic. Based on Theorem 4.4 we get* CPC $\prec_\tau$ HA $\prec_\rho$ IPC, *with $\tau(p) = \{\langle \neg\neg p, \top\rangle\}$ and $\rho(\langle p, q\rangle) = \{p \to q, q \to p\}$ doing the job.*

The discussion concerning the composition of refinements via interpretation is not straightforward. For *vertical composition* one gets

Theorem 4.5 *Let $\mathcal{L}$, $\mathcal{L}'$ and $\mathcal{L}''$ be k, l and m-logics over Σ, Σ' and Σ'' respectively. Let τ be a (k,l)-translation from Σ to Σ' and ρ a (l,m)-translation from Σ' to Σ''. Suppose that $\mathcal{L} \rightharpoonup_{\tau} \mathcal{L}'$, $\mathcal{L}' \rightharpoonup_{\rho} \mathcal{L}''$ and ρ interprets $\mathcal{L}^{\tau}$. Then $\mathcal{L} \rightharpoonup_{\rho\circ\tau} \mathcal{L}''$*

Proof. Directly from the fact that $\mathcal{L} \rightharpoonup_{\tau} \mathcal{L}'$ and $\mathcal{L}' \rightharpoonup_{\rho} \mathcal{L}''$ we have that $\Gamma \vdash_{\mathcal{L}} \bar{\varphi}$ implies $\rho(\tau(\Gamma)) \vdash_{\mathcal{L}''} \rho(\tau(\bar{\varphi}))$ for any $\Gamma \cup \{\bar{\varphi}\} \subseteq \mathrm{Fm}^k(\Sigma)$.

On the other hand, we have by hypothesis and the previous theorem, that for any $\Gamma, \{\bar{\varphi}\} \subseteq \mathrm{Fm}^k(\Sigma)$,

$$\Gamma \vdash_{\mathcal{L}} \bar{\varphi} \Leftrightarrow \tau(\Gamma) \vdash_{\mathcal{L}^{\tau}} \tau(\bar{\varphi}) \Leftrightarrow \rho(\tau(\Gamma)) \vdash_{\mathcal{L}^{\tau\rho}} \rho(\tau(\bar{\varphi})),$$

and hence, $\rho \circ \tau$ interprets $\mathcal{L}$. Therefore, $\mathcal{L} \rightharpoonup_{\rho\circ\tau} \mathcal{L}''$. □

On the other hand, *horizontal composition* of refinements via interpretations is still a topic of current research, which leads us to the conclusions of this paper.

5 Conclusions and future work

The paper succeeded in providing a smooth generalization of a rather new approach to refinement based on *logic interpretations*, a powerful tool used in algebraic logic, to arbitrary logic systems. As one could expect, this generic account of *refinement by interpretation* is mainly useful at the specification meta-level, *i.e.*, whenever an implementation step requires a change in the underlying logic. This often arises, in formal software development, with the need for accommodating new requirements (as in Example 6) or when a particular theorem prover, embodying a specific logic, is to be used for validating design (as in Example 7).

A lot of questions, however, remain to be answered. For example, one can intuitively accept that a refinement via signature morphism, in the usual sense, can be regarded as a refinement via interpretation. However, in the framework introduced in this paper, this is not achieved in a straightforward way since signature morphisms implicitly define a translation of the variables. Thus in order to accommodate in our framework the classical refinement procedure, a logic has to be parameterized by the variables used to generate its formulas.

On the other hand, we believe this approach has an enormous application potential, when tuned to the many variants of pure algebraic specification of software, namely the approaches of observational logic [8], hidden logic [16] and [14] and behavioral logic [9]. In all of these cases the satisfaction of requirements is discussed up to some particular satisfaction relation and their verification is checked with respect to relations obtained by replacing, in the standard satisfaction relation, strict equality by its underlying notion of satisfaction. In this context, the adoption of a semantics based on k-data structures, offers theoretical support to the unification of all of these approaches since models in all of those cases consist of algebras whose k-data structures are of the form $\langle A, \theta \rangle$ where θ captures the particular satisfaction relation in each formalisms.

Naturally, most of the models (and τ-models) of software specifications are not admissible choices as implementations. For example, the structure $\langle A, \nabla_A \rangle$ is a

model of any logic over the corresponding specification signature. Therefore, the choice of adequate filters along the implementation process becomes a crucial, although not trivial task. It should be done according to the system nature (for example, adopting observational equality to deal with objects with encapsulated data). A similar concern is, moreover, shared by other general approaches, to formal development, as, for example, within the behavioral logic of [9].

Acknowledgement

This research was developed in the context of the project MONDRIAN (under the contract PTDC/EIA-CCO/108302/2008) and was supported by FCT (Portuguese Foundation for Science and Technology) under the contracts SFRH/BDE/33650/2009, PTDC/EIA/73252/2006, at Minho University, as well as PTDC/MAT/68723/2006 and the Unidade de Investigação Matemática e Aplicações of University of Aveiro.

References

[1] W. Blok and D. Pigozzi. Abstract algebraic logic and the deduction theorem. Preprint. To appear in the Bulletin of Symbolic Logic. Available at http://www.math.iastate.edu/dpigozzi/papers/aaldedth.pdf.

[2] W. Blok and D. Pigozzi. Algebraizable logics. *Memoirs of the American Mathematical Society*, 396, Amer. Math. Soc., Providence, 1989.

[3] W. Blok and J. Rebagliato. Algebraic semantics for deductive systems. *Studia Logica*, 74(1-2):153–180, 2003.

[4] Don Batory, J. N. Sarvela, and A. Rauschmayer. Scaling step-wise refinement. *IEEE Trans. in Sofware Engineering*, 30(6):355–371, 2004.

[5] J. Czelakowski. *Protoalgebraic Logics*. Trends in logic, Studia Logica Library, Kluwer Academic Publishers, 2001.

[6] H. A. Feitosa and I. M. L. D'Ottaviano. Conservative translations. *Ann. Pure Appl. Logic*, 108(1-3):205–227, 2001.

[7] H. Feitosa. *Traduções Conservativas*. PhD thesis, Universidade Federal de Campinas, Instituto de Filosofia e Ciências Humanas, 1997.

[8] Rolf Hennicker and Michel Bidoit. Observational logic. In *AMAST '98: Proceedings of the 7th International Conference on Algebraic Methodology and Software Technology*, pages 263–277, London, UK, 1999. Springer-Verlag.

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

[10] A. Madeira. Observational refinement process. *Electr. Notes Theor. Comput. Sci.*, 214:103–129, 2008.

[11] M. A. Martins. Behavioral institutions and refinements in generalized hidden logics. *Journal of Universal Computer Science*, 12(8):1020–1049, 2006.

[12] T. Mossakowski, R. Diaconescu, and A. Tarlecki. What is a logic translation ? *Logica Universalis*, 2009.

[13] M. A. Martins, A. Madeira, and L. S. Barbosa. Refinement via interpretation. In *Proc. of 7th IEEE Int. Conf. on Software Engineering and Formal Methods*, Hanoi, Vietnam, November 2009. IEEE Computer Society Press.

[14] M. A. Martins and D. Pigozzi. Behavioural reasoning for conditional equations. *Mathematical. Structures in Comp. Sci.*, 17(5):1075–1113, 2007.

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

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

[17] D. Sannella and A. Tarlecki. *Foundations of Algebraic Specifications and Formal Program Development*. Cambridge University Press, To appear.

[18] A. Tarlecki. Abstract specification theory: An overwiew. In *Models, Algebras, and Logics of Engineering Software,M. Broy, M. Pizka eds.*, NATO Science Series, Computer and Systems Sciences, VOL 191, pages 43–79. IOS Press, 2003.

[19] M. Wirsing. Algebraic specification. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science (volume B)*, pages 673–788. Elsevier - MIT Press, 1990.