From Universal Logic to Computer Science, and back

Răzvan Diaconescu

Simion Stoilow Institute of Mathematics of the Romanian Academy
Razvan.Diaconescu@imar.ro

Abstract. Computer Science has been long viewed as a consumer of mathematics in general, and of logic in particular, with few and minor contributions back. In this article we are challenging this view with the case of the relationship between specification theory and the universal trend in logic.

1 From Universal Logic...

Although universal logic has been clearly recognised as a trend in mathematical logic since about one decade only, mainly due to the efforts of Jean-Yves Béziau and his colleagues, it had a presence here and there since much longer. For example the anthology [9] traces universal logic ideas back to the work of Paul Herz in 1922. In fact there is a whole string of famous names in logic that have been involved with universal logic in the last century, including Paul Bernays, Kurt Gödel, Alfred Tarski, Haskell Curry, Jerzy Łoś, Roman Suszko, Saul Kripke, Dana Scott, Dov Gabbay, etc.

Universal logic is not a new super-logic, but it is rather a body of general theories of logical structures, in the same way universal algebra is a general theory of algebraic structures (see [8] for a discussion about what universal logic is and is not). Within the last century mathematical logic has witnessed the birth of a multitude of unconventional logical systems, such as intuitionistic, modal, multiple valued, paraconsistent, non-monotonic logics, etc. Moreover, a big number of new logical systems have appeared in computer science, especially in the area of formal methods. The universal logic trend constitutes a response to this new multiplicity by the development of general concepts and methods applicable to a great variety of logical systems. One of the aims of universal logic is to determine the scope of important results (such as completeness, interpolation, etc.) and to produce general formulations and proofs of such results. This is very useful in the applications and helps with the distinction between what is and what is not really essential for a certain particular logical system, thus leading to a better understanding of that logical system. Universal logic may also be regarded as a toolkit for defining specific logics for specific situations; for example, temporal deontic paraconsistent logic. It also helps with the clarification of basic concepts by explaining what is an extension, or what is a deviation, of a given logic, what it means for a logic to be equivalent or translatable to another logic, etc. Paramount researchers in mathematical logic consider universal logic as a true renaissance of the study of logic, that is based on very modern principles and methodologies and that responds to the new mathematical logic perspectives. The dynamism of this area, its clear identity, and its high potential have been materialized through a dedicated new book series (*Studies in Universal Logic*, Springer Basel), a dedicated new journal (*Logica Universalis*, Springer Basel), a dedicated corner of *Journal of Logic and Computation* (Oxford Univ. Press), and through a dedicated series of world congresses and schools (UNILOG: Switzerland 2005, China 2007, Portugal 2010, Brazil 2013; see `www.uni-log.org`).

The analogy between universal algebra and universal logic however fails in the area of the supporting mathematical structures. While the former is in fact a mathematical theory based upon a relatively small set of core mathematical definitions, this is not the case with the latter. There is not a single commonly accepted mathematical base for universal logic. Instead the universal trend in logic includes several theories each of them supported by adequate mathematical structures that share a non-substantialist view on logic phenomena, free of commitment to particular logical systems, and consequently a top-down development methodology. One of the most famous such theories is Tarski's general approach to logical consequence via closure operators [60]. And perhaps now the single most developed mathematical theory in universal logic is the *institution theory* of Goguen and Burstall [39,40].

2 ...to Computer Science,...

2.1 Origins of institution theory

Around 1980's there was already a population explosion of logical systems in use computer science, especially in the logic-based areas such as specification theory and practice. People felt that many of the theoretical developments (concepts, results, etc.), and even aspects of implementations, are in fact independent of the details of the actual logical systems. Especially in the area of structuring of specifications or programs, it would be possible to develop the things in a completely generic way. The benefit would be not only in uniformity, but also in clarity since for many aspects of specification theory the concrete details of actual logical systems may appear as irrelevant, with the only role being to suffocate the understanding. The first step to achieve this was to come up with a very general model oriented formal definition for the informal concept of logical system. The model theoretic orientation is dictated by formal specification in which semantics plays a primary role. Due to their generality, category theoretic concepts appeared as ideal tools. However there is something else which makes category theory so important for this aim: its deeply embedded non-substantialist thinking which gives prominence to the relationships (morphisms) between objects in the detriment of their internal structure. Moreover, category theory was at that time, and continues even now to be so, the mathematical field of the upmost importance for computer science. In fact, it was computer science that recovered the status of category theory, at the time much diminished in conventional mathematical areas. The article [37] that Joseph Goguen wrote remains one of the most relevant and beautiful essays on the significance of category theory for computer science and not only.

The categorical model theories existing at the time, although quite deep theoretically, were however unsatisfactory from the perspective of a universal logic approach to specification. Sketches of [35,44,62] had just developed another language for expressing (possibly infinitary) first-order logic realities. The satisfaction as cone injectivity [1,2,3,49,47,46], whilst considering models as objects of abstract categories, lacks a multi-signature aspect given by the signature morphisms and the model reducts, which leads to severe methodological limitations. Moreover in both these categorical model theory frameworks, the satisfaction of sentences by the models is defined rather than being axiomatized, which give them a strong taste of concreteness in contradiction with universal logic aims and ideals. On the other hand, the model theory trend known as 'abstract model theory' [4,5] had an axiomatic approach to the satisfaction relation, it also had a multi-signature aspect, but it was still only concerned with extensions of conventional logic in that the signatures and the models are concrete, hence it lacked a fully universal aspect.

2.2 The concept of institution

The definition of institution [14,40] can be seen as representing a full generalisation of 'abstract model theory' of [4,5] in a true universal logic spirit by also considering the signatures and models as abstract objects in categories.

Definition 1 (Institutions). *An* institution $\mathcal{I} = (Sig^{\mathcal{I}}, Sen^{\mathcal{I}}, Mod^{\mathcal{I}}, \models^{\mathcal{I}})$ *consists of*

1. *a category* $Sig^{\mathcal{I}}$*, whose objects are called* signatures*,*
2. *a functor* $Sen^{\mathcal{I}} : Sig^{\mathcal{I}} \to \mathbf{Set}$ *(to the category of sets), giving for each signature a set whose elements are called* sentences *over that signature,*
3. *a functor* $Mod^{\mathcal{I}} : (Sig^{\mathcal{I}})^{\mathrm{op}} \to \mathbf{CAT}$ *(from the opposite of* $Sig^{\mathcal{I}}$ *to the category of categories) giving for each signature* Σ *a category whose objects are called* Σ-models*, and whose arrows are called* Σ*-(*model*)* homomorphisms*, and*
4. *a relation* $\models^{\mathcal{I}}_{\Sigma} \subseteq |Mod^{\mathcal{I}}(\Sigma)| \times Sen^{\mathcal{I}}(\Sigma)$ *for each* $\Sigma \in |Sig^{\mathcal{I}}|$*, called* Σ-satisfaction*,*

such that for each morphism $\varphi : \Sigma \to \Sigma'$ *in* $Sig^{\mathcal{I}}$*, the* satisfaction condition

$$M' \models^{\mathcal{I}}_{\Sigma'} Sen^{\mathcal{I}}(\varphi)(\rho) \quad \text{if and only if} \quad Mod^{\mathcal{I}}(\varphi)(M') \models^{\mathcal{I}}_{\Sigma} \rho$$

holds for each $M' \in |Mod^{\mathcal{I}}(\Sigma')|$ *and* $\rho \in Sen^{\mathcal{I}}(\Sigma)$*.*

The functions $Sen^{\mathcal{I}}(\varphi)$ *are called* sentence translation functions *and the functors* $Mod^{\mathcal{I}}(\varphi)$ *are called* model reduct functors*.*

The literature (e.g. [22,57]) shows myriads of logical systems from computing or from mathematical logic captured as institutions. In fact, an informal thesis underlying institution theory is that any logic may be captured by the above definition. While this should be taken with a grain of salt, it certainly applies to any logical system based on satisfaction between sentences and models of any kind. However the very process of formalising logical systems as institutions is not a trivial one as it has to provide precise and consistent mathematical definitions for basic concepts that are commonly considered in a rather naive style. Moreover these definitions have to obey the axioms of institution. For example we will see below how the template given by Def. 1 shapes a drastically reformed understanding of logical languages (signatures) and variables.

The following example may convey an understanding about the process of capturing of a logical system as institution.

Example 1 (Many sorted algebra as institution). This is a very common logical system in computer science, and constitutes the logical basis of traditional algebraic specification. It is also used frequently in the literature as an example of the definition of institution; however there are some slight differences between various formalisations of many sorted algebra as institution. Here we sketch this institution in accordance with [22] and other papers of the author in the recent years.

The signatures (S, F) consist of a set of sorts (types) S and a family F of functions typed by arities (finite strings of sorts) and sorts, i.e. $F = (F_{w\to s})_{w\in S^*, s\in s}$. Signature morphisms map symbols such that arities are preserved; they can be presented as families of functions between corresponding sets of function symbols. Given a signature, its models M are many-sorted algebras interpreting sorts s as sets M_s, and function symbols $\sigma \in F_{s1\dots sn\to s}$ as functions $M_\sigma : M_{s1} \times \cdots \times M_{sn} \to M_s$. Model homomorphisms are many-sorted algebra homomorphisms. Model reduct means reassembling the models components according to the signature morphism, i.e. for any signature morphism $\varphi : (S, F) \to (S', F')$ and any (S', F')-model M' we have $Mod(\varphi)(M')_x = M'_{\varphi(x)}$ for each x sort in S or function symbol in F. The sentences are first-order formulæ formed from atomic equations (i.e. equalities between well formed terms having the same sort) by iteration of logical connectives $(\wedge, \neg)$ and (first-order) quantifiers $\forall X$ (where X is a finite block of S-sorted variables). Sentence translation means replacement of the translated symbols, for example for variables the sort is changed accordingly.

Satisfaction is the usual Tarskian satisfaction of a first-order sentence in a many-sorted algebra that is defined by induction on the structure of the sentences.

When working out the details of this definition, the *Sig*, *Mod* and $\models$ components are straightforward. Less so is *Sen* that requires a careful management of the concept of variable, an issue that will be discussed below in some detail. The proof of the Satisfaction Condition is done by induction on the structure of the sentences, the only non-trivial step corresponding to the quantifications. This involves some mild form of model amalgamation (see [22]).

2.3 The expanse of institution theory

Def. 1 constitutes the starting concept of institution theory. Institution theory currently comprises a rather wide (both in terms of internal developments and applications) and constitutes a dynamic research area. The relationship between the concept of institution and institution theory is somehow similar to that between the concept of group and group theory in algebra. The definition of group is very simple and abstract and it does not convey the depth and expanse of group theory; the same holds for institutions and institution theory. The theory of institutions has gradually emerged as the most fundamental mathematical tool underlying algebraic specification theory (in its wider meaning) [57], also being increasingly used in other areas of computer science. And a lot of model theory has been gradually developed at the level of abstract institutions (see [22]), with manyfold consequences including a systematic supply of model theories to (sometimes sophisticated) non-conventional logical systems, but also new deep results in conventional model theory.

We refrain here from discussing in some details the rather long list of achievements of institution theory, instead we refer to the survey [26] that gives a brief account of the development of institution theory both in computer science and in mathematical logic (model theory).

3 ...and back

The wide body of abstract model theory results developed within institution theory (many of them collected in [22]) can be regarded as an important contribution of computer science to logic and model theory in general, and to universal logic in particular. However here we will set this aside and instead will focus on something else, which is more basic and subtle in the same time, namely on the reformed understanding of some important basic concepts in logic. Through our analysis we will see that this has been made possible not only because of the universal logic aspect of institution theory, but especially because of its computer science origins. Computer science in general, and formal methods in particular, cannot afford a naive informal treatment of logical entities for the simple reason that often these have to be realised directly in implementations. It is thus no surprise that in many situation issues arising from implementation of formal specification languages can be very consonant with issues regarding the mathematical rigor imposed by the definition of institution and the corresponding solutions are highly convergent.

In this section we will discuss the new understanding of the concepts of logical language, variables, quantifiers, interpolation brought in by institution theory. We conclude with a brief discussion challenging the common view on many sorted logics.

3.1 On logical languages

Logical languages are the primary syntactic concept in mathematical logic. Informally they represent structured collections of symbols that, on the one hand are used as extra-logical symbols[1] in the composition of the sentences or formulæ, and on the other hand are interpreted, often in set theory, in order to get semantics. In institution theory the logical languages correspond to the objects of *Sig* and are called *signatures*, a terminology that owes to computer science. Institution theory leads to a more refined understanding of two aspects of logical languages, namely mappings between languages and variables.

Signature morphisms and language extensions. In Def. 1, *Sig* is a category rather than a class; this means that morphisms between signatures play a primary role. In fact the category theoretic thinking leans towards morphisms rather than towards objects, objects are somehow secondary to morphisms. Some early and courageous presentations of category theory [34] even do it without the concept of object since objects can be assimilated to identity morphisms. In concrete situations the fact that *Sig* is only required to be category gives a lot of freedom with respect to the choice for an actual concept of signature morphism. One extreme choice is not to have proper signature morphisms at all or even that *Sig* has only one object. The latter situation is common to logical studies in which no variation in the language is necessary. A less extreme choice is made in the traditional model theory practice, namely to have only language (signature) extensions as morphisms. However, mathematically this may be quite an unconventional choice since usually, in concrete situations, morphisms are structure preserving mappings between objects and from this perspective signature extensions represent a rather strong restriction.

With respect to signature morphisms the practice of formal specification is quite different than that of mathematical logic in that it considers more sophisticated concepts of mappings between languages. The example of the many-sorted algebra institution given above is quite illustrative in this respect. The practice of algebraic specification (especially in the area of parameterised specifications) requires much more than signature extensions, it requires at least the fully general structure-preserving morphisms as in the aforementioned example. Moreover the literature (e.g. [57]) considers also an even more complex concept of signature morphism, the so-called *derived signature morphisms* that are in fact second-order substitutions replacing function symbols by terms. These of course are also immediately accommodated by the *Sig* part of Def. 1. This widening of the concept of language extension to various forms of signature morphisms has manyfold implications in all areas that involve the use of language extensions. For example paramount logical concepts such as interpolation and definability get a much more general formulation (see [22,59,20,51] etc.) with important consequences in the applications.

The case of the derived signature morphisms shows that in some situations the simple criterion of preserving the mathematical structure is not enough for defining a fully usable concept of signature morphism. There is also another famous case that comes from the behavioural specification trend [52,53,41,42,10,45,29,54].

[1] Logical symbols are connectors such as $\wedge$, $\neg$,..., or quantifiers $\forall, \exists$, or modalities $\Box, \Diamond$, etc. Sorts (types), function, relations symbols, etc. are extra-logical symbols.

When defining the corresponding institution(s), the use of the mere structure-preserving mappings for the signature morphisms leads to the failure of the Satisfaction Condition of Def. 1. In order to get that holding, an additional condition has to be imposed on the signature morphisms known in the literature as the 'encapsulation condition' and which in the concrete applications corresponds clearly to an object-orientation aspect. In both [38] and [41] the authors remark that the derivation of the encapsulation condition on the morphisms of signatures from the meta-principle of invariance of truth under change of notation (the Satisfaction Condition of institutions) seems to confirm the naturalness of each of the principles. We may add here that this shows an inter-dependency between the abstract logic level and pragmatical computer science aspects.

Variables and quantifiers. The concept of variable is primary when having to deal with quantifications. Mathematical logic has a common way to treat variables which has a *global* aspect to it. A typical example is the following quotation from [15] that refers to the language of first-order logic.

> "To formalize a language $\mathcal{L}$, we need the following *logical symbols*
> parentheses), (;
> variables $v_0, v_1, \ldots, v_n, \ldots$;
> connectives $\wedge$ (and), $\neg$ (not);
> quantifier $\forall$ (for all);
> and one binary relation symbol $\equiv$ (identity). We assume, of course, that no symbol in $\mathcal{L}$ occurs in the above list."

Upon analysis of this text we can easily understand that variables are considered as logical rather than extra-logical symbols which also implies that, as a collection, they are invariant with respect to the change of the signature. Moreover they have to be disjoint from the signatures. And of course, this collection of variables ought to be infinite.

While such treatment of variables may work well when having to deal only with ad-hoc signature extensions, as it is the case with conventional model theory. However it rises a series of technical difficulties with the institution theoretic approach.

1. Having a set of variables χ as logical symbols means that the respective institution has χ as a parameter. Therefore, strictly speaking, it is improper to talk, for example, about *the* institution of first-order logic.
2. In the concrete situations the category *Sig* is usually defined in the style of Ex. 1, which means that the individual signatures are set theoretic structures that are not restricted in any way on the basis of the fixed set of variables. This of course cannot guarantee the principle of disjointness between the signatures and the variables. For example it is possible that some signatures may contain some of the variables as constants.
3. Moreover, the institution-theoretic approach to quantifiers [22,59,19] etc. abstracts blocks of variables just to signature morphisms $\varphi : \Sigma \to \Sigma'$, where in the concrete situations φ stands for the extension of Σ with a respective

block of variables. This means that while the variables have to be disjoint from the signature Σ, *they are actually part of* Σ'.

Unfortunately much of the institution theory literature is quite sloppy about these issues and adopts the common logic view of variables. However starting with [32] a series of works in institution theory adopts a view on variables that responds adequately to the aforementioned issues and therefore is mathematically rigorous. This is based on a *local* rather than the common global view of variable, drawing inspiration from the actual implementations of specification languages. For many sorted algebra (Ex. 1) it goes like this. Given a signature (S, F), a block of variables for (S, F) consists of a finite set of triples $(x, s, (S, F))$ where x is the name of the variable and $s \in S$ is its sort. It is also required that in any block of variables different variables have different names. Because of the qualification by the signature (the third component), by axiomatic set theory arguments we get that a variable for a signature is disjoint from the respective signature. On the other hand, they can be adjoined to the signature. So, given a block X of variables for a many sorted signature (S, F) let $(S, F + X)$ denote the new signature obtained by adding the variables of sort s as new constants of sorts s. Then for any $(S, F + X)$-sentence ρ we have that $(\forall X)\rho$ is a (S, F)-sentence. In this way satisfaction of quantified sentences can be defined only in terms of model reducts, without having to resort to traditional concepts such as valuations of variables that have a strong concrete aspect. An (S, F)-model M satisfies $(\forall X)\rho$ if and only if for each $(S, F + X)$-model M' such that $Mod(\varphi)(M') = M$ we have that M' satisfies ρ, where φ denotes the signature expansion $(S, F) \to (S, F + X)$. Note that this definition is institution theoretic since it does not depend on the many sorted algebra case, it can be formulated in exactly the same way in abstract institutions. Moreover, our local concept of variable also behaves well with respect to the sentence translations induced by signature morphisms. Given a signature morphism $\chi\colon\ (S, F) \to (S', F')$, any block of variables X for (S, F) translates to a block of variables X' for (S', F') by mapping each variable $(x, s, (S, F))$ to $(x, \chi(s), (S', F'))$. "Behaves well" here means two things: (1) that we get a block of variables as required, and (2) that the translation is functorial. Then latter aspect is crucial for the functor axioms for Sen.

It is very interesting to note that this local view on variables, necessary to meet the mathematical rigor of the definition of institution, fits the way logical variables are treated in actual implementations of specification languages (e.g. CafeOBJ [28], etc.). There variables are declared explicitly and their scope is restricted to the module in which they are declared. The way this fits exactly the aforementioned approach to logical variables is explained by the fact that, according to works such as [24] the institutions underlying specification languages have structured specifications or modules as their signatures, so in this case the qualification by the signature of the institution means qualification by a corresponding module.

The mathematical properties underlying our local approach to logical variables are axiomatised by the following abstract notion which has been used in a series of works (e.g. [23,48,27,31], etc.) for building explicit quantifications in abstract institutions in a way that it yields another sentence functor (and consequently another institution that shares the signatures and the models with the original institution).

Definition 2 (Quantification space). *For any category Sig a subclass of arrows $\mathcal{D} \subseteq Sig$ is called a* quantification space *if, for any $(\chi : \Sigma \to \Sigma') \in \mathcal{D}$ and $\varphi : \Sigma \to \Sigma_1$, there is a designated pushout*

$$\begin{array}{ccc} \Sigma & \xrightarrow{\varphi} & \Sigma_1 \\ {\scriptstyle \chi}\downarrow & & \downarrow{\scriptstyle \chi(\varphi)} \\ \Sigma' & \xrightarrow[\varphi[\chi]]{} & \Sigma'_1 \end{array}$$

with $\chi(\varphi) \in \mathcal{D}$ and such that the 'horizontal' composition of such designated pushouts is again a designated pushout, i.e. for the pushouts in the following diagram

$$\begin{array}{ccccc} \Sigma & \xrightarrow{\varphi} & \Sigma_1 & \xrightarrow{\theta} & \Sigma_2 \\ {\scriptstyle \chi}\downarrow & & \downarrow{\scriptstyle \chi(\varphi)} & & \downarrow{\scriptstyle \chi(\varphi)(\theta)} \\ \Sigma' & \xrightarrow[\varphi[\chi]]{} & \Sigma'_1 & \xrightarrow[\theta[\chi(\varphi)]]{} & \Sigma'_2 \end{array}$$

$\varphi[\chi]; \theta[\chi(\varphi)] = (\varphi; \theta)[\chi]$ *and* $\chi(\varphi)(\theta) = \chi(\varphi; \theta)$*, and such that* $\chi(1_\Sigma) = \chi$ *and* $1_\Sigma[\chi] = 1_{\Sigma'}$.

The use of designated pushouts is required by the fact that quantified sentences ought to have a unique translation along a given signature morphism. The coherence property of the composition is required by the functoriality of the translations. For example, in the aforementioned concrete case of many sorted algebra, $\mathcal{D}$ consists of the signature extensions $\varphi : (S, F) \to (S, F + X)$ where X is a finite block of variables for (S, F). For any signature morphism $\chi : (S, F) \to (S', F')$ we define $X' = \{(x, \chi(s), (S', F')) \mid (x, s, (S, F)) \in X\}$, $\varphi[\chi]$ to be signature extension $(S', F') \to (S', F' + X')$ and $\chi(\varphi) : (S, F + X) \to (S', F' + X')$ to be the canonical extension of χ that maps each variable $(x, x, (S, F))$ to $(x, \chi(s), (S', F'))$.

3.2 On interpolation

Because of its many applications in logic and computer science, interpolation is one of the most desired and studied properties of logical systems. Although it has a strikingly simple and elementary formulation as follows,

> given sentences ρ_1 and ρ_2, if ρ_2 is a consequence of ρ_1 (written $\rho_1 \vdash \rho_2$) then there exists a sentence ρ (called *interpolant*) in the common language of ρ_1 and ρ_2 such that $\rho_1 \vdash \rho$ and $\rho \vdash \rho_2$,

in general it is very difficult to establish. The famous result of Craig [16] marks perhaps the birth of the study of interpolation, proving it for (single sorted) first-order logic. The actual scope of Craig's result has been gradually extended to many other logical systems (for example in the world of modal logics, see [36]), a situation that meets the universal character of interpolation that can be easily detected from its formulation that does not seem to commit inherently to any particular logical system.

The institution theoretic approach to interpolation has lead to a multi dimensional reformation of this important concept that will be discussed below. However, before that, we note that within institution theory the consequence relation $\vdash$ from the above formulation of interpolation is interpreted as the semantic consequence $\models$, i.e. for a given signature Σ and sets E, Γ of Σ-sentences, $E \models \Gamma$ when for each Σ-model M if M satisfies each sentence in E then it satisfies each sentence in Γ too.

From single sentences to sets of sentences. It has been widely believed that equational logic, the logical system underlying traditional algebraic specification, lacks interpolation; likewise for Horn-clause logic and other such fragments of first-order logic. As far as we know, Piet Rodenburg [55,56] was the first to point out that this is a misconception due to a basic misunderstanding of interpolation, rooted in the heavy dependency of logic culture on classical first-order logic with all its distinctive features taken for granted. Then it follows the grave general fault of exporting a coarse understanding of concepts dependent on details of a particular logical system to other logical systems of a possibly very different nature, where some detailed features may not be available. In the case of interpolation, the gross confusion has to do with looking for an interpolant as a single sentence. In first-order logic, which has conjunction, looking for interpolants as finite sets of sentences ($\{\rho_1, \ldots, \rho_n\}$) is just the same as looking for interpolants as single sentences ($\rho_1 \wedge \cdots \wedge \rho_n$). Hence, the common formulation of interpolation requires single sentences as interpolants. However, this is not an adequate formulation for equational logic which lacks conjunction, i.e., conjunction $\rho_1 \wedge \rho_2$ of universally quantified equations ρ_1 and ρ_2 cannot be captured as a universally quantified equation in general. Rodenburg [55,56] proved that equational logic has interpolation with the interpolant being a finite set of sentences, and this apparently weaker interpolation property is quite sufficient in both computer science and logic applications.

From language extensions to signature morphisms. The relationship between signatures Σ_1 (of ρ_1), Σ_2 (of ρ_2) and their union $\Sigma_1 \cup \Sigma_2$ (where the consequence $\rho_1 \vdash \rho_2$ happens) and intersection $\Sigma_1 \cap \Sigma_2$ (the signature of the interpolant), is depicted by the following diagram where arrows indicate the obvious inclusions:

$$\begin{array}{ccc} \Sigma_1 \cap \Sigma_2 & \xrightarrow{\subseteq} & \Sigma_1 \\ {\scriptstyle\subseteq}\big\downarrow & & \big\downarrow{\scriptstyle\subseteq} \\ \Sigma_2 & \xrightarrow[\subseteq]{} & \Sigma_1 \cup \Sigma_2 \end{array}$$

While intersections $\cap$ and unions $\cup$ are more or less obvious for signatures as used in first-order logic and in many other standard logics, they are not so in some other logical systems, and certainly not at the level of abstract institutions where signatures are just objects of an arbitrary category. One immediate response to this problem would be to add an infrastructure to the abstract category of signatures that would support concepts of $\cap$ and $\cup$; in fact this is already available in the institution theoretic literature and is called *inclusion system* [30,22]. Another solution would be, at the abstract level to use arbitrary signature morphisms and in the applications to restrict the signature morphisms only to those that are required to be, for example, inclusions (i.e. language extensions). Due to the abstraction involved, this means a lot of flexibility. For instance, in many computer science applications it is very meaningful to consider non-inclusive signature morphisms in the role of inclusions in the square above. An example comes from the practice of parameterised specifications (e.g. [57]) where instantiation of the parameters may involve signature morphisms that collapse syntactic entities. A generalised form of interpolation involving such non-injective signature morphisms is needed in order to get the completeness of formal verification for structured specifications (e.g. [12,11]). This generalisation of interpolation that relaxes language extensions to arbitrary signature morphisms has been introduced in [59]. The category-theoretic property of the above intersection-union square that makes things work is that it is a *pushout*. These considerations lead to the following abstract formulation of the interpolation property [22].

Definition 3 (Institution-theoretic Craig interpolation [22]). [2] *Given* $\mathcal{L}, \mathcal{R} \subseteq Sig$, *the institution has* Craig $(\mathcal{L},\mathcal{R})$-interpolation *when for each pushout square of signatures*

$$\begin{array}{ccc} \Sigma & \xrightarrow{\varphi_1 \in \mathcal{L}} & \Sigma_1 \\ {\scriptstyle\varphi_2 \in \mathcal{R}}\big\downarrow & & \big\downarrow{\scriptstyle\theta_1} \\ \Sigma_2 & \xrightarrow[\theta_2]{} & \Sigma' \end{array}$$

and any finite sets of sentences $E_1 \subseteq Sen(\Sigma_1)$ *and* $E_2 \subseteq Sen(\Sigma_2)$, *if* $\theta_1(E_1) \models \theta_2(E_2)$ *then there exists a finite set* E *of* Σ*-sentences such that* $E_1 \models \varphi_1(E)$ *and* $\varphi_2(E) \models E_2$.

The (abstract) restriction to pre-defined classes of signature morphisms, $\mathcal{L}$ for φ_1 and $\mathcal{R}$ for φ_2, constitutes an essential parameter in the above definition. In its

[2] Given a signature morphism $\varphi : \Sigma \to \Sigma'$, we abbreviate $Sen(\varphi)$ as φ, and so for a set of sentences $E \subseteq Sen(\Sigma)$, $\varphi(E)$ is the image of E under $Sen(\varphi)$.

absence the interpolation concept would be unrealistically too rigid and strong (for example many-sorted first-order logic would not support it [43,13,22]).

A couple of typical examples of institution-theoretic Craig $(\mathcal{L}, \mathcal{R})$-interpolation are as follows:

– many-sorted first-order logic for either $\mathcal{L}$ or $\mathcal{R}$ consisting of the signature morphisms that are injective on the sorts [43,22]; and
– many-sorted Horn clause logic for $\mathcal{R}$ consisting of the signature morphisms that are injective [20,22].

From Craig to Craig-Robinson interpolation. There is a variety of situations in model theory (e.g. Beth definability [7,15]) and in computer science (e.g. complete calculi for structured specifications [12]) when Craig interpolation is used together with implication. The latter property is so obvious in some logics – such as first-order logic – that it is hardly ever mentioned explicitly in concrete contexts. Its definition at the level of abstract institutions is straightforward [59]: an institution *has implication* when for every signature Σ and Σ-sentences ρ_1, ρ_2, there exists a Σ-sentence ρ such that for each Σ-model M,

$$M \models \rho \text{ if and only if } M \models \rho_2 \text{ whenever } M \models \rho_1.$$

However, in many contexts we may render implication unnecessary by reformulating the interpolation property. Important applications are definability [51] in model theory and the completeness of calculus for structured specifications [22] in computer science. The trick is to 'parameterise' each instance of interpolation by a set of 'secondary' premises. In [33,58,61] this is called *Craig-Robinson interpolation*; it also plays an important role in specification theory, e.g. [6,30,33,22]. Let us recall here explicitly its institution-theoretic formulation.

Definition 4 (Institution-theoretic Craig-Robinson interpolation). *An institution has* Craig-Robinson $(\mathcal{L}, \mathcal{R})$-interpolation *when for each pushout square of signatures with $\varphi_1 \in \mathcal{L}$ and $\varphi_2 \in \mathcal{R}$*

$$\begin{array}{ccc} \Sigma & \xrightarrow{\varphi_1} & \Sigma_1 \\ {\scriptstyle \varphi_2}\downarrow & & \downarrow{\scriptstyle \theta_1} \\ \Sigma_2 & \xrightarrow[\theta_2]{} & \Sigma' \end{array}$$

and finite sets of sentences $E_1 \subseteq Sen(\Sigma_1)$ and $E_2, \Gamma_2 \subseteq Sen(\Sigma_2)$, if $\theta_1(E_1) \cup \theta_2(\Gamma_2) \models \theta_2(E_2)$ then there exists a finite set E of Σ-sentences such that $E_1 \models \varphi_1(E)$ and $\varphi_2(E) \cup \Gamma_2 \models E_2$.

Clearly, Craig-Robinson interpolation implies Craig interpolation. In any compact institution with implication, Craig-Robinson interpolation and Craig interpolation are equivalent [25,22] (so for instance within first-order logic, the two properties coincide). This means that Craig-Robinson interpolation alone in principle is weaker than Craig interpolation and implication. But is it properly so? Is there a significant example of an institution lacking implication but having Craig-Robinson interpolation? Through a rather sophisticated technique of so-called Grothendieck institutions [18,21], a result in [22] gives a general method to lift Craig interpolation to Craig-Robinson interpolation in institutions that may not have implication but are embedded in a certain way into institutions having implication. A concrete consequence of this result based on the Craig interpolation property of many-sorted first-order logic that was mentioned above, is as follows.

Corollary 1 (Craig-Robinson interpolation in many-sorted Horn-clause logic). *Many-sorted Horn-clause logic (with equality) has* $(\mathcal{L}, \mathcal{R})$*-Craig-Robinson interpolation when* $\mathcal{L}$ *consists only of signature morphisms* φ *that are injective on sorts and 'encapsulate' the operations.*[3]

One of the important significance of this result can be seen in conjunction with the upgrade in [22] of the completeness result for structured specifications of [12], that replaces Craig interpolation and implication by Craig-Robinson interpolation as one of the conditions. In the light of [12], the lack of implication has been used in the formal specification community as an argument against the adequacy of equational logic as a specification formalism. However we can see that this was only due to a couple of misunderstandings (1) that implication is not really needed for obtaining the completeness result of [12] and (2) that equational logic does satisfy the kind of interpolation that is really needed there and in a form that meets the requirements of the applications. In practice the only restriction involved by the conditions of Cor. 1 is that all information hidings have to be done with morphisms from $\mathcal{L}$, something that seem to accord well with practical intuitions underlying the concept of information hiding.

3.3 A short word on many-sortedness

Another significance of the aforementioned Craig-Robinson interpolation property of many-sorted Horn-clause logic is that, if we reduce the context to conditional equational logic by not considering predicate symbols – which is the logic underlying the equational logic programming paradigm (e.g. [17]) – it makes sense only in the many-sorted context. In a single sorted context it is clear that $\mathcal{L}$ collapses to nothing. This is just one of the examples that sharply refutes an idea that, in my opinion, is common among mathematical logicians, namely that many-sorted logics are "inessential" variations of their single-sorted versions (e.g. [50]). Another example is of course the case of generalised Craig interpolation in first-order logic; while the single-sorted variant supports it for all pushout squares of signature morphisms, we have seen that it is not so for the many-sorted variant.

[3] In the sense that no operation symbol outside the image of φ is allowed to have a sort in the image of φ.

References

1. Hajnal Andréka and István Németi. Łoś lemma holds in every category. *Studia Scientiarum Mathematicarum Hungarica*, 13:361–376, 1978.
2. Hajnal Andréka and István Németi. A general axiomatizability theorem formulated in terms of cone-injective subcategories. In B. Csakany, E. Fried, and E.T. Schmidt, editors, *Universal Algebra*, pages 13–35. North-Holland, 1981. Colloquia Mathematics Societas János Bolyai, 29.
3. Hajnal Andréka and István Németi. *Generalization of the concept of variety and quasivariety to partial algebras through category theory*, volume 204 of *Dissertationes Mathematicae*. Pańswowe Wydawnictwo Naukowe, 1983.
4. Jon Barwise. Axioms for abstract model theory. *Annals of Mathematical Logic*, 7:221–265, 1974.
5. Jon Barwise and Solomon Feferman. *Model-Theoretic Logics*. Springer, 1985.
6. Jan Bergstra, Jan Heering, and Paul Klint. Module algebra. *Journal of the Association for Computing Machinery*, 37(2):335–372, 1990.
7. Evert Willem Beth. On Padoa's method in the theory of definition. *Indagationes Mathematicæ*, 15:330–339, 1953.
8. Jean-Yves Béziau. 13 questions about universal logic. *Bulletin of the Section of Logic*, 35(2/3):133–150, 2006.
9. Jean-Yves Béziau, editor. *Universal Logic: an Anthology*. Studies in Universal Logic. Springer Basel, 2012.
10. Michel Bidoit, Rolf Hennicker, and Martin Wirsing. Behavioural and abstractor specifications. *Sci. Comput. Program.*, 25(2-3):149–186, 1995.
11. Tomasz Borzyszkowski. Generalized interpolation in CASL. *Information Processing Letters*, 76:19–24, 2001.
12. Tomasz Borzyszkowski. Logical systems for structured specifications. *Theoretical Computer Science*, 286(2):197–245, 2002.
13. Tomasz Borzyszkowski. Generalized interpolation in first-order logic. *Fundamenta Informaticæ*, 66(3):199–219, 2005.
14. Rod Burstall and Joseph Goguen. The semantics of Clear, a specification language. In Dines Bjorner, editor, *1979 Copenhagen Winter School on Abstract Software Specification*, volume 86 of *Lecture Notes in Computer Science*, pages 292–332. Springer, 1980.
15. Chen-Chung Chang and H. Jerome Keisler. *Model Theory*. North Holland, Amsterdam, 1990.
16. William Craig. Three uses of the Herbrand-Gentzen theorem in relating model theory and proof theory. *Journal of Symbolic Logic*, 22:269–285, 1957.
17. Răzvan Diaconescu. Category-based semantics for equational and constraint logic programming, 1994. DPhil thesis, University of Oxford.
18. Răzvan Diaconescu. Grothendieck institutions. *Applied Categorical Structures*, 10(4):383–402, 2002. Preliminary version appeared as IMAR Preprint 2-2000, ISSN 250-3638, February 2000.
19. Răzvan Diaconescu. Institution-independent ultraproducts. *Fundamenta Informaticæ*, 55(3-4):321–348, 2003.
20. Răzvan Diaconescu. An institution-independent proof of Craig Interpolation Theorem. *Studia Logica*, 77(1):59–79, 2004.
21. Răzvan Diaconescu. Interpolation in Grothendieck institutions. *Theoretical Computer Science*, 311:439–461, 2004.
22. Răzvan Diaconescu. *Institution-independent Model Theory*. Birkhäuser, 2008.

23. Răzvan Diaconescu. Quasi-boolean encodings and conditionals in algebraic specification. *Journal of Logic and Algebraic Programming*, 79(2):174–188, 2010.
24. Răzvan Diaconescu. An axiomatic approach to structuring specifications. *Theoretical Computer Science*, 433:20–42, 2012.
25. Răzvan Diaconescu. Borrowing interpolation. *Journal of Logic and Computation*, 22(3):561–586, 2012.
26. Răzvan Diaconescu. Three decades of institution theory. In Jean-Yves Béziau, editor, *Universal Logic: an Anthology*, pages 309–322. Springer Basel, 2012.
27. Răzvan Diaconescu. Quasi-varieties and initial semantics in hybridized institutions. *Journal of Logic and Computation*, DOI:10.1093/logcom/ext016.
28. Răzvan Diaconescu and Kokichi Futatsugi. *CafeOBJ Report: The Language, Proof Techniques, and Methodologies for Object-Oriented Algebraic Specification*, volume 6 of *AMAST Series in Computing*. World Scientific, 1998.
29. Răzvan Diaconescu and Kokichi Futatsugi. Behavioural coherence in object-oriented algebraic specification. *Universal Computer Science*, 6(1):74–96, 2000. First version appeared as JAIST Technical Report IS-RR-98-0017F, June 1998.
30. Răzvan Diaconescu, Joseph Goguen, and Petros Stefaneas. Logical support for modularisation. In Gerard Huet and Gordon Plotkin, editors, *Logical Environments*, pages 83–130. Cambridge, 1993. Proceedings of a Workshop held in Edinburgh, Scotland, May 1991.
31. Răzvan Diaconescu and Alexandre Madeira. Encoding hybridized institutions into first order logic. *Mathematical Structures in Computer Science*, To appear.
32. Răzvan Diaconescu and Marius Petria. Saturated models in institutions. *Archive for Mathematical Logic*, 49(6):693–723, 2010.
33. Theodosis Dimitrakos and Tom Maibaum. On a generalized modularization theorem. *Information Processing Letters*, 74:65–71, 2000.
34. Charles Ehresmann. *Catégories et strcutures*. Dunod Paris, 1965.
35. Charles Ehresmann. Esquisses et types des structures algébriques. *Buletinul Institutului Politehnic Iaşi*, 14(18):1–14, 1968.
36. Dov M. Gabbay and Larisa Maksimova. *Interpolation and Definability: modal and intuitionistic logics*. Oxford University Press, 2005.
37. Joseph Goguen. A categorical manifesto. *Mathematical Structures in Computer Science*, 1(1):49–67, March 1991. Also, Programming Research Group Technical Monograph PRG–72, Oxford University, March 1989.
38. Joseph Goguen. Types as theories. In George Michael Reed, Andrew William Roscoe, and Ralph F. Wachter, editors, *Topology and Category Theory in Computer Science*, pages 357–390. Oxford, 1991. Proceedings of a Conference held at Oxford, June 1989.
39. Joseph Goguen and Rod Burstall. Introducing institutions. In Edward Clarke and Dexter Kozen, editors, *Proceedings, Logics of Programming Workshop*, volume 164 of *Lecture Notes in Computer Science*, pages 221–256. Springer, 1984.
40. Joseph Goguen and Rod Burstall. Institutions: Abstract model theory for specification and programming. *Journal of the Association for Computing Machinery*, 39(1):95–146, 1992.
41. Joseph Goguen and Răzvan Diaconescu. Towards an algebraic semantics for the object paradigm. In Hartmut Ehrig and Fernando Orejas, editors, *Recent Trends in Data Type Specification*, volume 785 of *Lecture Notes in Computer Science*, pages 1–34. Springer, 1994.
42. Joseph Goguen and Grant Malcolm. A hidden agenda. *Theoretical Computer Science*, 245(1):55–101, 2000.

43. Daniel Găină and Andrei Popescu. An institution-independent proof of Robinson consistency theorem. *Studia Logica*, 85(1):41–73, 2007.
44. René Guitart and Christian Lair. Calcul syntaxique des modèles et calcul des formules internes. *Diagramme*, 4, 1980.
45. Rolf Hennicker and Michel Bidoit. Observational logic. In A. M. Haeberer, editor, *Algebraic Methodology and Software Technology*, number 1584 in LNCS, pages 263–277. Springer, 1999. Proc. AMAST'99.
46. Michael Makkai. Ultraproducts and categorical logic. In C.A. DiPrisco, editor, *Methods in Mathematical Logic*, volume 1130 of *Lecture Notes in Mathematics*, pages 222–309. Springer Verlag, 1985.
47. Michael Makkai and Gonzolo Reyes. *First order categorical logic: Model-theoretical methods in the theory of topoi and related categories*, volume 611 of *Lecture Notes in Mathematics*. Springer, 1977.
48. Manuel-Antonio Martins, Alexandre Madeira, Răzvan Diaconescu, and Luis Barbosa. Hybridization of institutions. In Andrea Corradini, Bartek Klin, and Corina Cîrstea, editors, *Algebra and Coalgebra in Computer Science*, volume 6859 of *Lecture Notes in Computer Science*, pages 283–297. Springer, 2011.
49. Günter Matthiessen. Regular and strongly finitary structures over strongly algebroidal categories. *Canadian Journal of Mathematics*, 30:250–261, 1978.
50. J. Donald Monk. *Mathematical Logic*. Springer-Verlag, 1976.
51. Marius Petria and Răzvan Diaconescu. Abstract Beth definability in institutions. *Journal of Symbolic Logic*, 71(3):1002–1028, 2006.
52. Horst Reichel. Behavioural equivalence – a unifying concept for initial and final specifications. In *Proceedings, Third Hungarian Computer Science Conference*. Akademiai Kiado, 1981. Budapest.
53. Horst Reichel. *Initial Computability, Algebraic Specifications, and Partial Algebras*. Clarendon, 1987.
54. Grigore Roşu. *Hidden Logic*. PhD thesis, University of California at San Diego, 2000.
55. Pieter-Hendrik Rodenburg. Interpolation in conditional equational logic, 1989. Preprint from Programming Research Group at the University of Amsterdam.
56. Pieter-Hendrik Rodenburg. A simple algebraic proof of the equational interpolation theorem. *Algebra Universalis*, 28:48–51, 1991.
57. Donald Sannella and Andrzej Tarlecki. *Foundations of Algebraic Specifications and Formal Software Development*. Springer, 2012.
58. Joseph Shoenfield. *Mathematical Logic*. Addison-Wesley, 1967.
59. Andrzej Tarlecki. Bits and pieces of the theory of institutions. In David Pitt, Samson Abramsky, Axel Poigné, and David Rydeheard, editors, *Proceedings, Summer Workshop on Category Theory and Computer Programming*, volume 240 of *Lecture Notes in Computer Science*, pages 334–360. Springer, 1986.
60. Alfred Tarski. On some fundamental concepts of metamathematics. In *Logic, Semantics, Metamathematics*, pages 30–37. Oxford University Press, 1956.
61. Paulo Veloso. On pushout consistency, modularity and interpolation for logical specifications. *Information Processing Letters*, 60(2):59–66, 1996.
62. Charles F. Wells. Sketches: Outline with references. Unpublished draft.