From Universal Logic to Computer Science, and back

Răzvan Diaconescu

Simion Stoilow Institute of Mathematics of the Romanian Academy, Romania

ICTAC 2014

Message

- Computer Science has been long viewed as a consumer of mathematics in general, and of logic in particular.
- Few and minor contributions back (attributed to Computer Science).
- Also often Computing Science blamed for poor intelectual and academic quality.
  (Rather than immediately dismissed this should be considered seriously.)
- Here we are challenging these views with the case of the relationship between *specification theory* and the *universal* trend in *logic*.

From Universal Logic...

The universal trend in logic

...to Computer Science...

Origins of Institution Theory

The concept of Institution

The expanse of Institution Theory

...and Back

On logical languages

- Signature morphisms and language extensions
- Variables and Quantifiers

On Interpolation

- From single sentences to sets of sentences.
- From language extensions to signature morphisms.
- From Craig to Craig-Robinson Interpolation.

Part I

From Universal Logic...

The universal trend in logic

Origins and history of Universal Logic

- Recognised as trend in (mathematical) logic since about one decade; some consider it as a true renaissance of mathematical logic.

- Its name coined by J.-Y. Beziau et al.

- But had a presence since much much longer (eg 1922).

  J.-Y. Beziau
  Universal Logic: an Anthology.
  Springer Basel (2012).

- Important names include Paul Herz, Paul Bernays, Kurt Gödel, Alfred Tarski, Haskell Curry, Jerzy Łoś, Roman Suszko, Saul Kripke, Dana Scott, Dov Gabbay, etc.

J.-Y. Béziau

What is (not) Universal Logic?

- It is a response to the (recent) great multitude of logical systems from
  - *mathematical logic* (ie. intuitionistic, modal, many valued, paraconsistent, non-monotonic, etc.) and
  - *computer science* (a lot of them!),

  most of them non-classical and/or unconventional.
- It is *not* a new super logic.
- It is rather a (dispersed) body of general mathematical theories of logical structures, that shares the following principles:
  - no commitment to particular logical systems, and
  - logic phenomena considered from a non-substantialist perspective.

Aims of Universal Logic

- To develop general concepts and method applicable to a great variety of logical systems.
- To determine and clarify the scope of important results (such as completeness, interpolation, etc.) and to produce general formulations and proofs of such results.
- To provide a toolkit for defining specific logics for specific situations.
- Clarification of basic important concepts, eg. logic translation, interpolation, etc.

Example:
Tarski's abstract theory of consequence

Famous Tarski's axiomatisation of logical consequence:

- $\{\rho\} \vdash \rho$;
- $E \subseteq E', E \vdash \rho$ implies $E' \vdash \rho$;
- $E \vdash \gamma$ for each $\gamma \in \Gamma, \Gamma \vdash \rho$ implies $E \vdash \rho$.

Academic infrastructure of Universal Logic

- a dedicated book series: *Studies in Universal Logic*, Springer Basel;
- a dedicated journal: *Logica Universalis*, Springer Basel;
- a dedicated corner of *Journal of Logic and Computation*, Oxford Univ. Press;
- a dedicated series of world congresses and schools: *UNILOG* (Switzerland 2005, China 2007, Portugal 2010, Brazil 2013, Turkey 2015; see `www.uni-log.org`).

Part II

...to Computer Science...

Institution Theory

The single most developed mathematical theory in universal logic is the *institution theory* (Goguen, Burstall).

J. Goguen and R. Burstall.
Institutions: Abstract Model Theory for Specification and Programming,
J.ACM 39(1):95–146, 1992.

D. Sannella, A. Tarlecki.
Foundations of Algebraic Specifications and Formal Software Development.
Springer (2012).

R. Diaconescu.
Institution-independent Model Theory.
Springer Basel (2008).

Origins of Institution Theory

The concept of Institution

The expanse of Institution Theory

Original motivation

- Around 80's there was already a population explosion of logical systems in use computer science, esp. formal specification.
- Feeling 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.
- The example of Clear (Goguen, Burstall approx. 1980), a generic language for structuring formal specifications.
- A generic development would bring uniformity, clarity, etc. to the theory and practice.

Theoretical sources

The first step to achieve this logical genericity would be to come up with a a very general model oriented formal definition for the informal concept of logical system.

Two main theoretical sources:

1. Category theory (Mac Lane, Eilenberg) – already intensely used at the time in the theory of ADTs;
2. Abstract model theory (Barwise, Feferman)

Origins of Institution Theory

The concept of Institution

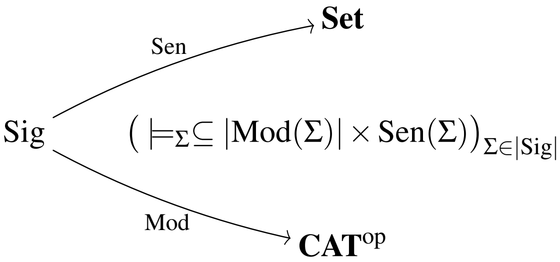
The expanse of Institution Theory

J. Goguen, R. Burstall

Institution I

'Logical system' as mathematical object:

$\mathscr{I} = (\mathrm{Sig}, \mathrm{Sen}, \mathrm{Mod}, \models)$:



Institution II

Satisfaction Condition

$$
\begin{array}{lll}
\Sigma & \rho \in \mathrm{Sen}(\Sigma) & \mathrm{Mod}(\varphi)(M') \models_{\Sigma} \rho \\
\downarrow \varphi & & \Updownarrow \\
\Sigma' & M' \in |\mathrm{Mod}(\Sigma')| & M' \models_{\Sigma'} \mathrm{Sen}(\varphi)(\rho)
\end{array}
$$

- Expresses the invariance of truth with respect to change of notation;
- inspired from Barwise-Feferman's abstract model theory.

Abstract abstract model theory

Institution theory is the only fully abstract and fully axiomatic approach to model theory.

	language	**sentences**	**models**	**satisfaction**
Institution Theory *(Goguen-Burstall)*	abstract	abstract	abstract	abstract
sketches *(Ehresmann)*	defined	defined	defined	defined
cone injectivity *(Andreka-Nemeti)*	defined	defined	abstract	defined
Abstract model th. *(Barwise-Feferman)*	defined	abstract	defined	abstract

Myriads of logics as institutions

- Countless logical system from computing science and mathematical logic have been captured as institutions; classical or non-classical, conventional or unconventional.
- *Thesis*: any logical system based on satisfaction between models and sentences of any kind may be captured as institution.
- The process of formalising concrete logics as institutions is non-trivial, sometimes shapes a reformed understanding of the respective logical system.

Example (brief):
Many-sorted Algebra as institution

- many-sorted signatures (S, F);
- first-order sentences based on many-sorted terms and variables;
- first-order models interpreting sorts s as sets M_s and function symbols $\sigma \in F_{w \to s}$ as functions $M_\sigma : M_w \to M_s$;
- usual Tarski styled satisfaction (by induction on the structure of sentences).
- the proof of the Satisfaction Condition not immediate, involves a certain (mild) form of model amalgamation.

Origins of Institution Theory

The concept of Institution

The expanse of Institution Theory

An analogy with group theory

Definition of group very simple and abstract, yet group theory developed into a very wide, deep and sophisticated research area.

The same applies to institution theory.

Significance in computer science

- Emerged as the most fundamental mathematical theory underlying logic-based formal [algebraic] specification theory (in its wider meaning).
- Uniform and generic development of concepts and methods in specification and programming independently of the underlying formalism.
- Its relevance expanded to other computer science areas, eg. ontologies, etc.

Significance in logic

- A lot of model theory has been gradually developed at the level of abstract institutions, including many deep results (ie. completeness, interpolation, axiomatizability, etc).
- Reformed understanding of important traditional concepts in logic and model theory, often leading to new results even in well studied classical areas.
- Uniform supply of model theories to non-conventional logical systems.
- Provide a systematic method for *logic by translation*.
- **All these can be regarded as contributions of computing science to logic.**

A huge literature

The institution theory literature both in logic and computing science comprises of hundreds of research papers spread over a wide spectrum of conferences and journals.

A brief survey of the expanse of institution theory in computing science and logic is

R. Diaconescu.
Three decades of institution theory,
in **Universal logic: an anthology**, Springer (2012).

Part III

...and Back

The wide body of abstract abstract model theory developed within institution theory can be regarded as important contributions of computing science to logic and model theory.

However we will not do this here.

Instead our focus will be on some more subtle and striking contributions, namely the reform of some important basic concepts in logic.

Logical languages

- They are primary syntactic concepts in mathematical logic, representing structured collections of symbols used in two ways;
  1. as extra-logical symbols in the composition of sentences/formulæ; and
  2. interpreted (often in set theory) in order to get semantics.
- In institution theory they are called *signatures* and correspond to the objects of the category Sig.

On signature morphisms

- Unlike in traditional logic, because of category theory, in institution theory morphisms between signatures play a primary role.
- Since (at abstract level) Sig is a fully abstract category, in concrete situations this means a lot of flexibility for the choice of actual signature morphisms:
  1. no signature morphisms at all, or even Sig being singleton –> no variation of the logical language;
  2. signature extensions –> common practice in conventional model theory;
  3. structure preserving mappings –> the most adequate from a mathematical viewpoint;
  4. *derived* signature morphisms –> similar to second order substitutions, used in instantiations of parameterised specifications (so-called *views*).

Then striking case of behavioural specifications

A general concept of signature morphisms accommodates also peculiar situations, eg. *behavioural signature morphisms*.

In the logics underlying behavioural specifications, morphisms just preserving the math structure of the signatures lead to failures of the Satisfaction Condition.

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*...

...which in the concrete applications corresponds clearly to an object-orientation aspect.

The derivation of the encapsulation condition from the meta-principle of invariance of truth under change of notation shows an inter-dependency between the abstract logic level and pragmatical computer science aspects.

Consequences of abstract signature morphisms

- Paramount concepts such as *interpolation* or *definability* get a more general formulation, with important applications.
- A generic treatment of *quantifiers* based on signature morphisms and model reducts.
- A generic *methods of diagrams* (one of the most important methods in model theory).
- etc.

The traditional treatment of logical variables

From

C.-C. Chang, H.J. Keisler.
Model Theory.
North Holland (1990).

(but very common and unchallenged):

> *To formalize a language $\mathscr{L}$, 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 $\mathscr{L}$ occurs in the above list.*

Remarks

Upon analysis of the traditional approach we can understand that:

- Variables are considered logical rather than extra-logical symbols.
- As a collection they are invariant with respect to change of signature (hence *global* aspect).
- They are disjoint from the signatures.
- The (fixed) collection of variables ought to be infinite.

Some problems

With institution theory the traditional approach to variables leads to some problems:

- The fixed collection (χ) of variables appears as a parameter of the logic.
- The disjointness between χ and signatures
  - imposes some unwholesome technical restrictions on the definition of Sig (which may affect its compositional properties);
  - clashes with the institution approach to quantifiers (as signature extensions).

The local approach to variables

- draws inspiration from implementations of specification languages (eg. CafeOBJ);
- **block of variables** X **for a signature** (S, F):
  - is a finite set of triples $(x, s, (S, F))$, called *variables*, where
  - x is the *name* of the variable and is unique (in the block), and
  - $s \in S$ is the *sort* of the variable,

Remark

Due to set theoretic arguments, X *and* (S, F) *are disjoint.*

Quantifications over local variables

- blocks X can be adjoined to (S, F) as by adding the variables as new constants; the result is the signature denoted $(S, F+X)$.
- $(\forall X)\rho$ is a (S, F)-sentence for each $(S, F+X)$-sentence ρ.
- The satisfaction $M \models (\forall X)\rho$ can be defined only in terms of model reducts, thus avoiding the concept of valuation:

$$M \models (\forall X)\rho \quad \text{if and only if} \quad M' \models \rho$$

for all $(S, F+X)$-expansions M' of M.

Translations along signature morphisms

- Any (S,F)-block X is translated along a signature morphism $\varphi\colon (S,F) \to (S',F')$ by

$$(x,s,(S,F)) \mapsto (x,\varphi(s),(S',F')).$$

- The result is an (S',F')-block.
- This also ensures the functoriality of the sentence functor.

Correspondence to actual implementations

- In specification languages logical variables are always considered *locally*.
- Their scope is restricted to the modules in which they are declared.
- This corresponds exactly to our qualification of variables by signatures since the institution of structured specifications has specification modules as its signatures.

A categorical axiomatization

The class of extensions $(S,F) \to (S,F+X)$ can be axiomatised as a class of arrows $\mathscr{D} \subseteq \mathrm{Sig}$ such that:

1. for any $(\chi : \Sigma \to \Sigma') \in \mathscr{D}$ and $\varphi : \Sigma \to \Sigma_1$, there is a designated pushout

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

2. designated pushouts compose horizontally

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

$$\Sigma' \xrightarrow{\ (\varphi;\theta)[\chi]\ } \Sigma'_2$$

What is Interpolation?

- extremely simple to formulate and understand:

  if $\rho_1 \vdash \rho_2$ then there exists ρ in the common language (signature) of ρ_1 and ρ_2 such that $\rho_1 \vdash \rho$ and $\rho \vdash \rho_2$.

- but in general very very difficult to establish;
- still a central topic in logic;
- for first order logic first established by Craig (1957);
- very many applications in logic and computer science.

On unnatural failures of interpolation

- Traditionally known to fail in important (esp for computing) logics such as equational, Horn clause, etc.
- The apparent cause of these failures is the lack of conjunction, i.e. for ρ_1, ρ_2 sentences, $\rho_1 \wedge \rho_2$ is no longer a sentence.
- The subtle cause is the traditional misconception that uses single rather than (finite) sets of sentences in the formulation of interpolation.
- Piet Rodenburg (1991), the first to remark that equational logic has interpolation with sets of sentences.
- *Lesson: grave mistake to export a coarse understanding of concepts dependent on details of a particular logical system to other ones where some detailed features may not be available.*

The signature square of conventional interpolation

The relationship between signatures Σ_1 (of ρ_1), Σ_2 (of ρ_2), $\Sigma_1 \cup \Sigma_2$ (where $\rho_1 \vdash \rho_2$ happens) and $\Sigma_1 \cap \Sigma_2$ (the signature of the interpolant), is depicted by the following square of inclusions:

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

The signature square of generalized interpolation

The former square can be abstracted to a *pushout* square in Sig, where $\mathscr{L}, \mathscr{R} \subseteq \mathrm{Sig}$ are designated sub-classes of signature morphisms.

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

Institution-theoretic Craig Interpolation

It now looks like this (including interpolant as set-of-sentences):

Definition

Given $\mathscr{L}, \mathscr{R} \subseteq \mathrm{Sig}$, the institution has
Craig $(\mathscr{L}, \mathscr{R})$*-interpolation*
when for each pushout square of signatures

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

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

Benefits of generalized interpolation

- *Simplicity*: no need to bother with technical concepts of inclusion, intersection ($\Sigma_1 \cap \Sigma_2$), union ($\Sigma_1 \cup \Sigma_2$).
- *Wider applicability*: formal specification practice needs interpolation with signature morphisms that are non-injective.
- *Flexibility*: by setting properly the parameters $\mathscr{L}, \mathscr{R}$ we obtain various possibilities for interpolation to hold, accordingly to the intended applications.

Some examples

- many-sorted first-order logic for either $\mathscr{L}$ or $\mathscr{R}$ consisting of the signature morphisms that are injective on the sorts; and
- many-sorted Horn clause logic for $\mathscr{R}$ consisting of the signature morphisms that are injective.

What is Craig-Robinson interpolation (CRi)?

- an extension of Craig interpolation that embeds implicitly a form of implication;
- it plays an important role in specification theory and in other areas of computing science;
- in first order logic, due to existence of (explicit) implication, CRi and Ci collapse to the same interpolation concept;
- however in general CRi strictly more general that Ci plus implication
  (since there are logics without implication but with CRi).

Institution-theoretic Craig-Robinson Interpolation

Definition

An institution has *Craig-Robinson* $(\mathcal{L},\mathcal{R})$*-interpolation* when for each pushout square of signatures

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

and finite sets $E_1 \subseteq \mathrm{Sen}(\Sigma_1)$ and $E_2, \Gamma_2 \subseteq \mathrm{Sen}(\Sigma_2)$, 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 $\varphi_2(E) \cup \Gamma_2 \models E_2$.

Beyond implication

- Obviously CRi => Ci.
- Formally, Ci + implication => CRi.
- Are there logics without implication but with CRi?
- YES! but within the context of generalised interpolation (with arbitrary signature morphisms).

Lifting Craig to Craig-Robinson interpolation

General method to lift Ci to CRi in institutions without implication through a sophisticated technique (eg. involving a Grothendieck construction on institutions, etc.).

Corollary

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

- For many years belief that Horn logic not suitable for modular formal specification because of lack of implication.
- However implication not needed in conjunction to Ci since both together can be replaced with CRi.
- $\mathscr{L}$ and $\mathscr{R}$ above fit very well the pragmatics of structuring specifications.

A short word on many-sortedness

- Without predicate symbols (eg. the logic of the equational logic programming paradigm) in the single sorted case $\mathscr{L}$ collapses to empty.
- Hence many sortedness is crucial for the above result.
- This is one of the many situations sharply refuting the common idea among logicians that many-sorted logics are "inessential" variants of the their single-sorted variants.

THANKS!