INSTITUTION THEORY AND APPLICATIONS

RĂZVAN DIACONESCU

Habilitation

Simion Stoilow Institute of Mathematics of the Romanian Academy

January 2014 –

SUMMARY

Institution theory is a very general mathematical study of formal logical systems, with emphasis on semantics, that is not committed to any particular concrete logical system. It is based upon a mathematical definition for the informal notion of logical system, called *institution*, which includes both syntax and semantics as well as the relationship between them. Because of its very high level of abstraction this definition accommodates not only well established logical systems but also very unconventional ones and moreover it has served and it may serve as a template for defining new ones. Institution theory was introduced by Joseph Goguen and Rod Burstall in the late seventies as a response to the explosion in the population of logical systems in use in formal specification theory and practice and it has become part of the *universal logic* trend which approaches logic from a relativistic, non-substantialist perspective, quite different from the common reading of logic, both in philosophy and in exact sciences.

Institution theory is generally used in two main ways, as mathematical foundations for computer science studies (especially formal specification, but not only) and as an abstract approach to logical model theoretic studies. Between these two trends there is a strong interdependency, for example if the underlying logic enjoys good model theoretic properties then the respective language has good specification features. In this habilitation thesis we present authored developments over the past two decades in the institution theoretic approach to model theory and to formal specification.

The institution theoretic approach to model theory is based upon the development of most important conventional model theory methods and concepts at the level of abstract institutions. These include semantics of logical connectives and quantifiers, the methods of diagrams, of ultraproducts, of saturated models, concepts of axiomatizability, interpolation, definability. Important results in these areas are developed, sometimes representing high generalisations of well established results, and some of them leading to new results even in well extensively studied topics in concrete conventional logic. On the other hand all these provide a very convenient path to developing model theories to unconventional less studied concrete logical systems, including modal and many valued logics but not only. We also present proof theoretic developments within institution theory which play an important role in the institution theoretic approach to completeness results. An alternative powerful way to establishing logical results is of borrowing them across theoroidal comorphisms, which are encoding mappings between institutions. Important results obtained by borrowing include interpolation and definability theorems, existence of saturated models, etc. We also present a hierarchical way to logic combination that we apply to obtain many-valued or modal extensions of logical systems, as well as a combination between specification logics within the context of the semantics of the CafeOBJ language.

On the algebraic specification side, institutions provide a solid conceptual framework for the definition of coherent hidden algebra, the logic underlying the behavioural specification paradigm of CafeOBJ. The institution theoretic studies of the structuring of specifications or programs include here an axiomatic ap-

proach to structured specifications, module algebra studies, parameterized specifications. Due to the special compositionality properties of hidden algebra signatures, the module algebra of behavioural specification presents some particular problems. Specification over logically heterogeneous environments represents an important modern trend. Its foundations are given by a Grothendieck construction on institutions. Other developments presented are institution theoretic foundations for logic programming and for structural induction as a basis for formal verification of inductive properties. Many of the theoretical developments presented here have been motivated by the design of the algebraic specification language CafeOBJ, its main features being also presented here.

REZUMAT

Teoria instituțiilor este un studiu matematic foarte general al sistemelor logice formale, cu accent pe semantică, și care nu este legat de nici un sistem logic particular. Ea se bazează pe o definiție matematică pentru noțiunea informală de sistem logic, numită *instituție*, care include atât sintaxa cât și semantica, precum și relația dintre acestea. Din cauza nivelului foarte ridicat de abstractizare, această definiție poate curpinde nu numai sisteme logice bine stabilite, dar, de asemenea, și sisteme logice neconvenționale și în plus a servit și servește ca șablon pentru definirea altelor noi. Teoria instituțiilor a fost introdusă de către Joseph Goguen și Rod Burstall la sfârșitul anilor șaptezeci, ca răspuns la explozia de sisteme logice utilizate în teoria și practica specificațiilor formale, între timp devenind parte din curentul *logicii universale*, curent care abordează logica din perspectivă relativistă, non-substanțialistă, diferită de abordarea comună a logicii atât în filozofie cât și în științele exacte.

Teoria instituților este folosită în general în două moduri principale, ca fundamente matematice pentru studii în informatică (în special în domeniul specificațiilor formale, dar nu numai), și ca o abordare abstractă a studiilor de teoria modelelor. Între aceste două tendințe există o interdependență puternică, de exemplu, în cazul în care logica de bază se bucură de proprietăți bune de teoria modelelor, atunci limbajul respectiv are caracteristici bune de specificații. În această teză de abilitare prezentăm dezvoltări din ultimele două decenii în abordarea instituțională a teoriei modelelor și a specificațiilor formale.

Abordarea instituțională a teoriei modelelor se bazează pe dezvoltarea celor mai importante metode și concepte din teoria convențională a modelelor la nivelul instituțiilor abstracte. Acestea includ semantica conectorilor logici și ai cuantificatorilor, metoda de diagramelor, a ultraproduselor, a modelelor saturate, concepte de axiomatizabilitate, interpolare, definabilitate. Sunt dezvoltate rezultate importante în aceste domenii, acestea uneori reprezentând generalizări înalte ale unor rezultate bine stabilite, iar unele dintre acestea conducând la rezultate noi chiar și în cazul unor probleme studiate extensiv în logica concretă convențională. Pe de altă parte, toate acestea oferă o cale foarte convenabilă pentru dezvoltarea de teorii ale modelelor pentru sisteme logice concrete neconventionale mai puțin studiate, inclusiv logici modale și multi-valuate, și nu numai. Deasemeni prezentăm dezvoltări de teoria demonstrației care joacă un rol important în abordarea instituțională a rezultatelor de completitudine. O modalitate alternativă eficientă de stabilire a rezultatelor logice este împrumutul de-a lungul comorfismelor theoroidale, care reprezintă codări între instituții. Rezultatele importante obținute prin împrumuturi includ teoreme de interpolare și definabilitate, existența de modele saturate, etc. Deasemeni se prezintă o metodă de combinație ierarhică de sisteme logice, aplicată pentru obținerea sistematică de extensii multi-valuate și de extensii modale a sistemelor logice, precum și combinarea de sisteme logice de specificații în contextul semanticii limbajului CafeOBJ.

Pe partea de specificații algebrice, instituțiile oferă un cadru conceptual solid pentru definirea de logicii algebrelor cu sorturi ascunse, logică care se află la baza paradigmei de specificații comportamentale în

CafeOBJ. Studiile instituționale de structurare a specificațiilor și programelor prezentate aici includ o abordare axiomatică a specificațiilor structurate, studii de algebre de module, de parametrizări de specificații. Datorită proprietăților speciale de compoziționalitate a signaturilor de algebre cu sorturi ascunse, algebra respectivă de module ridică niște probleme particulare. Specificațiile peste medii eterogene logic reprezintă o tendință modernă importantă. Fundamentele sale sunt date de o construcție Grothendieck peste instituții. Alte dezvoltări prezentate sunt abordări instituționale de semantică denotațională pentru programarea logică, și de inducție structurală ca o bază pentru verificarea formală a proprietăților inductive. Multe dintre evoluțiile teoretice prezentate aici sunt motivate de proiectarea limbajului de specificații algebrice CafeOBJ, principalele sale caracteristici fiind, de asemenea, prezentate aici.

ACKNOWLEDGEMENT

My deepest gratitude is towards my Professor, late Joseph Goguen, who brought me up as scientist, who taught me institution theory, and who put a lot of trust into my thinking. Secondly, I have benefited a lot from collaboration over many years with a number of colleagues, which I consider all to be also my friends. They have been supporting my work in a wide variety of ways, many of them being co-authors of some of my publications. This list includes Marc Aiguier, Jean-Yves Beziau, Rod Burstall, Carlos Caleiro, Kokichi Futatsugi, Daniel Găină, Shuusaku Iida, Alexandre Madeira, Manuel-Antonio Martins, Till Mossakowski, Kazuhiro Ogata, Marius Petria, Andrei Popescu, Grigore Roșu, Petros Stefaneas, Andrzej Tarlecki, Ionuț Țuțu, Uwe Wolter, etc. Thirdly I would like to thank to so many anonymous referees of my peer-reviewed publications. Their very dedicated and competent work represents a crucial contribution to my scientific achievements; in fact I consider them as hidden co-authors of my works. The last but not the least, I am indebted to my SNSB students from 2002 to 2011 with whom I learned a lot.

CONTENTS

Part I

Scientific Achievements

Institution theory is a categorical abstract model theory that arose about three decades ago [73] as a response to the explosion of the population of logical systems used for formal software specification. Its original aim was to develop as much computing science as possible in a general, uniform way, independently of particular logical systems. This has been achieved to an extent even greater than originally envisaged. The theory of institutions became the most fundamental mathematical tool underlying algebraic specification theory (in its wider meaning) [137], also being increasingly used in other areas of computer science. Moreover, institution theory is a major trend in the so-called 'universal logic' (in the sense envisaged by Jean-Yves Béziau [7, 8]) which is considered by many a true renaissance of mathematical logic. A lot of model theory has been gradually developed at the level of abstract institutions (see [38]). A relatively recent survey of the vast area of institution theory is [48].

The starting concept of institution theory is the formal definition of a logical system; this includes the syntax, the semantics and the satisfaction relation between them. It plays the same role as, for example, the definition of group plays for group theory. Although the definition of group is very simple, group theory is a vast sophisticated mathematical area. The same with the definition of institution and institution theory.

Definition 0.1. *[73] An* institution *is a tuple* $(\mathrm{Sign}, \mathrm{Sen}, \mathrm{Mod}, (\models_\Sigma)_{\Sigma \in |\mathrm{Sign}|})$ *that consists of*

- *a category* Sign *whose objects are called* signatures*,*
- *a functor* $\mathrm{Sen} : \mathrm{Sign} \to \mathbf{Set}$ *(to the category of sets) giving for each signature a set whose elements are called* sentences *over that signature,*
- *a (contravariant) functor* $\mathrm{Mod} : (\mathrm{Sign})^{op} \to \mathbf{CAT}$ *(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*
- *a relation* $\models_\Sigma \subseteq |\mathrm{Mod}(\Sigma)| \times \mathrm{Sen}(\Sigma)$ *for each* $\Sigma \in |\mathrm{Sign}|$*, called the* satisfaction relation*,*

such that for each morphism $\varphi : \Sigma \to \Sigma' \in \mathrm{Sign}$*, the* Satisfaction Condition

$$M' \models_{\Sigma'} \mathrm{Sen}(\varphi)(\rho) \text{ if and only if } \mathrm{Mod}(\varphi)(M') \models_\Sigma \rho \tag{1}$$

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

The literature (e.g. [38, 137]) 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.

Concerning notations, this document in general follows a certain pattern of institution and category theoretic notations that can be found in the author's publications, such as [38]. Categorical composition is written in diagrammatic notation. For any signature morphism in an institution if $M = \mathrm{Mod}(\varphi)(M')$, we denote $\mathrm{Mod}(\varphi)(M')$ by $M{\restriction_\varphi}$ and may we say that M is the φ-*reduct* of M' and that M' is a φ-*expansion* of M. Also we may abbreviate $\mathrm{Sen}(\varphi)(\rho)$ simply by $\varphi(\rho)$. We may use the also use the following notations

– for any $E \subseteq \mathrm{Sen}(\Sigma)$, E^* denotes $\{M \in |\mathrm{Mod}(\Sigma)| \mid M \models_\Sigma \rho \text{ for each } \rho \in E\}$;
– for any $E, E' \subseteq \mathrm{Sen}(\Sigma)$, $E \models E'$ denotes $E^* \subseteq E'^*$ and $E \models\!\models E'$ denotes $E^* = E'^*$;
– for any $E \subseteq \mathrm{Sen}(\Sigma)$, $\mathrm{Mod}(\Sigma, E)$ is the full subcategory of $\mathrm{Mod}(\Sigma)$ whose objects are in E^*.

1

LOGIC AND MODEL THEORY

1.1 INSTITUTION-INDEPENDENT MODEL THEORY

An important trend within institution theory is motivated by model theory research. Several important model theory methods have been developed at the level of abstract institutions and a lot of very general and yet deep results have been developed. This has resulted in a very abstract form of model theory, often refereed to as *institution-independent model theory* or synonymously *institutional model theory*. Many of the institution-independent model theory results constitute high generalisation of well known results from conventional concrete model theory and can be used for obtaining easily corresponding results in less conventional logical systems. The same can be said for model theoretic concepts.

In this chapter we present achievements by the author in this area, many of them being collected in the monograph [38], which is widely considered as the authoritative reference of institution-independent model theory.

1.1.1 *The method of diagrams*

In conventional model theory the method of diagrams is one of the most important methods. The institution-independent method of diagrams pervades the development of a lot of model theoretic results at the level of abstract institutions, many of these being presented in [38]. In the form introduced in [32] it is significantly simpler than a previously introduced one in [141, 142].

Definition 1.1 (The method of diagrams [32])**.** *An institution I has* diagrams *when for each signature Σ and each Σ-model M, there exists a signature Σ_M and a signature morphism $\iota_\Sigma(M) : \Sigma \to \Sigma_M$, functorial in Σ and M, and a set E_M of Σ_M-sentences such that $\mathrm{Mod}(\Sigma_M, E_M)$ and the comma category $M/\mathrm{Mod}(\Sigma)$ are naturally isomorphic, i.e. the following diagram commutes by the isomorphism $i_{\Sigma,M}$ that is natural in Σ and M*

$$\begin{array}{ccc} \mathrm{Mod}(\Sigma_M, E_M) & \xrightarrow{i_{\Sigma,M}} & (M/\mathrm{Mod}(\Sigma)) \\ & \searrow^{\mathrm{Mod}(\iota_\Sigma(M))} & \downarrow \text{forgetful} \\ & & \mathrm{Mod}(\Sigma) \end{array}$$

The signature morphism $\iota_\Sigma(M) : \Sigma \to \Sigma_M$ is called the elementary extension of Σ via M *and the set E_M of Σ_M-sentences is called the* diagram *of the model M.*

The existence of institution theoretic diagrams (in the sense of Dfn. 1.1) in concrete logical systems is a mark of coherence between the syntax (the kind of sentences involved) and the semantics (the concept

of model morphism employed). Since [32] this institution theoretic concept of diagrams has been used rather intensively in many institution-independent model theory and computer science works including existence of co-limits of models [32], interpolation [91], definability [129], Tarski's elementary chain theorem [90], axiomatizability [38], saturated models [61], initial semantics [38, 88], abstract constraint logic programming [47], etc.

1.1.2 *Internal logic*

The semantics of Boolean connectives (such as $\wedge$, $\neg$, etc.) as well as of quantifiers (based on models reducts along signature morphisms) has been introduced earlier in the institution theory literature by Tarlecki [140]. In order to complete the semantics of first order sentences at the level of abstract institutions in [31] two concepts are introduced and developed: *basic* sentences that cover the atomic sentences in concrete institutions, and *representable* signature morphisms that represent a categorical capture of the first order property of the quantifiers.

Definition 1.2 (Basic sentence)**.** *[31] Given a signature Σ in any institution, a Σ-sentence e is* basic *if there exists a Σ-model M_e, called the* basic model of *e such that for each Σ-model M, $M \models_\Sigma e$ if and only if there exists a model homomorphism $M_e \to M$. The sentence e is* epic basic *when the homomorphism $M_e \to M$ is an epi.*

Definition 1.3 (Representable signature morphism)**.** *[31] In any institution, a signature morphism $\chi : \Sigma \to \Sigma'$ is* representable *if and only if there exists a Σ-model M_χ (called the* representation of χ*) and an isomorphism i_χ of categories such that the following diagram commutes:*

$$\begin{array}{ccc} \mathrm{Mod}(\Sigma') & \xrightarrow{i_\chi} & (M_\chi / \mathrm{Mod}(\Sigma)) \\ & \searrow^{\mathrm{Mod}(\chi)} & \downarrow \text{forgetful} \\ & & \mathrm{Mod}(\Sigma) \end{array}$$

In [38] several important compositionality properties of representable signature morphisms are developed.

Often the finiteness of the quantification plays an important role. This has been defined in [31] for representable signature morphisms and extended in [38] to quantifications by any signature morphisms.

Definition 1.4. *[38] A signature morphism $\chi : \Sigma \to \Sigma'$ is* finitary *when for each co-limit $(\mu_i)_{i \in I}$ of a directed diagram $(f_{i,j})_{(i<j) \in (I,\leq)}$ of Σ-models*

$$\begin{array}{ccc} A_i & \xrightarrow{f_{i,j}} & A_j \\ & {\scriptstyle \mu_i} \searrow \quad \swarrow {\scriptstyle \mu_j} & \\ & A & \end{array}$$

and for each χ-expansion A' of A

- *there exists an index $i \in I$ and a χ-expansion $\mu'_i : A'_i \to A'$ of μ_i, and*

– *any two different expansions as above can be 'unified' in the sense that for any* χ*-expansions* μ'_i *and* μ'_k *as above there exists an index* $j \in I$ *with* $i,k < j$*, a* χ*-expansion* μ'_j *as above and* $f'_{i,j}, f'_{k,j}$ χ*-expansions of* $f_{i,j}, f_{k,j}$ *such that the following commutes*

$$\begin{array}{ccccc} A'_i & \xrightarrow{f'_{i,j}} & A'_j & \xleftarrow{f'_{k,j}} & A'_k \\ & \searrow^{\mu'_i} & \downarrow{\scriptstyle \mu'_j} & \swarrow_{\mu'_k} & \\ & & A' & & \end{array}$$

Fact 1.5. *A representable signature morphism* $\chi : \Sigma \to \Sigma'$ *is finitary if and only if its representation* M_χ *is finitely presented.*

Substitutions in institution theory are given by the following concept introduced in [33] within the context of an abstract approach to the denotational semantics of logic programming and developed in extenso in [38]. This has been further used and refined in [44] within the context of a general institution theoretic approach to structural induction.

Definition 1.6 (Substitutions)**.** *[33] For any signature* Σ *of an institution, and any signature morphisms* $\chi_1 : \Sigma \to \Sigma_1$ *and* $\chi_2 : \Sigma \to \Sigma_2$*, a* Σ-substitution $\psi : \chi_1 \to \chi_2$ *consists of a pair* $(\mathrm{Sen}(\psi), \mathrm{Mod}(\psi))$*, where*

– $\mathrm{Sen}(\psi) : \mathrm{Sen}(\Sigma_1) \to \mathrm{Sen}(\Sigma_2)$ *is a function, and*

– $\mathrm{Mod}(\psi) : \mathrm{Mod}(\Sigma_2) \to \mathrm{Mod}(\Sigma_1)$ *is a functor*

such that both of them preserve Σ*, i.e., the following diagrams commute:*

$$\begin{array}{ccc} \mathrm{Sen}(\Sigma_1) & \xrightarrow{\mathrm{Sen}(\psi)} & \mathrm{Sen}(\Sigma_2) \\ & {\scriptstyle \mathrm{Sen}(\chi_1)}\nwarrow \quad \nearrow{\scriptstyle \mathrm{Sen}(\chi_2)} & \\ & \mathrm{Sen}(\Sigma) & \end{array} \qquad \begin{array}{ccc} \mathrm{Mod}(\Sigma_1) & \xleftarrow{\mathrm{Mod}(\psi)} & \mathrm{Mod}(\Sigma_2) \\ & {\scriptstyle \mathrm{Mod}(\chi_1)}\searrow \quad \swarrow{\scriptstyle \mathrm{Mod}(\chi_2)} & \\ & \mathrm{Mod}(\Sigma) & \end{array}$$

and such that the following satisfaction condition holds:

$$\mathrm{Mod}(\psi)(M_2) \models \rho_1 \text{ if and only if } M_2 \models \mathrm{Sen}(\psi)(\rho_1)$$

for each Σ_2*-model* M_2 *and each* Σ_1*-sentence* ρ_1*.*

For any class $\mathcal{D}$ *of signature morphisms in an institution, let us say that a* $\mathcal{D}$-substitution *is just a substitution between signature morphisms in* $\mathcal{D}$*.*

Other uses of this concept of substitution include works on completeness [24, 89], on institution theoretic semantics of services [26], etc.

1.1.3 *The method of ultraproducts*

The method of ultraproducts is a most important and remarkably powerful one in conventional model theory [21]. This has been realised at the abstract level of institution theory beginning with [31], on the basis of the previously established concept of categorical ultraproduct (introduced perhaps first time in [109]) and

applied to the categories of models $\mathrm{Mod}(\Sigma)$ in institutions. Let us consider a family $(M_i)_{i\in I}$ of Σ-models in an institution and a filter F on I. For any $J \in F$ let us denote the direct product of $(M_i)_{i\in J}$ by M_J. If $\mathrm{Mod}(\Sigma)$ has direct products then for any $J \subseteq J'$ in F there is a canonical projection $p_{J'\geq J} : M_{J'} \to M_J$. Then any colimit $\mu = \{\mu_J : M_J \to M_F \mid J \in F\}$ of the diagram $\{p_{J'\geq J} \mid J \subseteq J', J \in F\}$ is called an *F-product* of $(M_i)_{i\in I}$. When F is ultrafilter, F-products are called *ultraproducts*.

The foundation of the ultraproducts method in first order model theory is constituted by a result in [105] which gives a 'preservation' property for the satisfaction by ultraproducts of models that is common to all sentences in first order logic. This has been highly generalised in institution theory in [31] by decomposing it into a puzzle of general preservation results across Boolean connectors and quantifiers.

Definition 1.7. *[31] For a signature Σ in an institution, for each filter $F \in \mathcal{F}$ over a set I and for each family $\{A_i\}_{i\in I}$ of Σ-models, a Σ-sentence e is*

- preserved by $\mathcal{F}$-factors *if $A_F \models_\Sigma e$ implies $\{i \in I \mid A_i \models_\Sigma e\} \in F$, and*
- preserved by $\mathcal{F}$-products *if $\{i \in I \mid A_i \models_\Sigma e\} \in F$ implies $A_F \models_\Sigma e$.*

When $\mathcal{F}$ is the class of all ultrafilters, preservation by $\mathcal{F}$-factors, respectively products, is called preservation by ultrafactors*, respectively* ultraproducts*.*

Theorem 1.8 (Fundamental ultraproducts theorem)**.** *[31] In any institution:*

1. *The basic sentences are preserved by all filtered products.*
2. *The finitary basic sentences are preserved by all filtered products and all filtered factors.*

For any class $\mathcal{F}$ of filters closed under reductions:

3. *The sentences preserved by $\mathcal{F}$-products are closed under existential χ-quantification, when χ preserves $\mathcal{F}$-filtered products.*
4. *The sentences preserved by $\mathcal{F}$-factors are closed under existential χ-quantification, when χ lifts $\mathcal{F}$-filtered products.*
5. *The sentences preserved by $\mathcal{F}$-factors and the sentences preserved by $\mathcal{F}$-filtered products are both closed under conjunction.*
6. *The sentences preserved by $\mathcal{F}$-products are closed under infinite conjunctions.*
7. *If a sentence is preserved by $\mathcal{F}$-factors then its negation is preserved by $\mathcal{F}$-products.*

And finally, if we further assume that $\mathcal{F}$ contains only ultrafilters:

8. *If a sentence is preserved by $\mathcal{F}$-products then its negation is preserved by $\mathcal{F}$-filtered factors.*
9. *The sentences preserved by both $\mathcal{F}$-products and factors are closed under negation.*

Concrete instances of this result and of some of its extensions provide for free an ultraproducts method for a variety of logical systems, including unconventional ones for which such development was otherwise difficult to envisage.

An immediate important general application of the general institution theoretic ultraproducts is compactness in abstract institutions [31, 38].

Proposition 1.9. *[31] Any institution in which each sentence is preserved by ultraproducts is model compact. Moreover if in addition the institution has negations then it is compact too.*

This remarkably general result when adjoined to the institution theoretic generalisation of the fundamental ultraproducts result of [105] given by Thm. 1.8 gives a very general compactness result, which can be instantiated with little effort to a wide variety of concrete logical systems. The efficiency of this path to compactness has become transparent for example in [62] in the case of a wide class of quantified modal systems.

Other applications of institution theoretic ultraproducts include a general ultrapower embedding theorem [31, 38], a general isomorphism criterion for finitely sized models [38], an institution theoretic generalization of the famous Keisler-Shelah isomorphism theorem [61], general preservation and axiomatizability results [38], and interpolation [34] and definability [129] by axiomatizability.

1.1.4 *Saturated models*

A lot of deep results in model theory can be reached by the method of saturated models. The follow definition of saturated models in abstract institution was given in [61].

Definition 1.10 (Saturated model)**.** *[61] For each signature morphism $\chi : \Sigma \to \Sigma'$, a Σ-model M* χ-realizes *a set E' of Σ'-sentences, if there exists a χ-expansion M' of M which satisfies E'. It* χ-realizes *finitely E' if it realizes every finite subset of E'.*

A Σ-model M is $(\lambda, \mathcal{D})$-saturated *for λ a cardinal and $\mathcal{D}$ a class of signature morphisms when for each ordinal $\alpha < \lambda$ and each $(\alpha, \mathcal{D})$-chain $(\Sigma_i \xrightarrow{\varphi_{i,j}} \Sigma_j)_{i<j\leq\alpha}$ with $\Sigma_0 = \Sigma$, for each $(\Sigma_\alpha \xrightarrow{\chi} \Sigma') \in \mathcal{D}$, each $\varphi_{0,\alpha}$-expansion of M χ-realizes any set of sentences if and only if it χ-realizes it finitely.*

An institution has $\mathcal{D}$-saturated models *if for any cardinal λ and for each Σ-model M there exists a Σ-homomorphism $M \to N$ such that M is elementary equivalent to N and N is $(\lambda, \mathcal{D})$-saturated.*

Two of the most useful properties of saturated models are their existence and their uniqueness [21]. The existence means that each model can be elementarily extended to a saturated model, while uniqueness holds when the model is 'sufficiently' small. Both properties have been developed at the level of abstract institutions in [61].

An important class of applications can be developed in conjunction with the method of ultraproducts. In [61] we have lifted at the level of abstract institutions an important result from conventional model theory [21], namely that for certain ultrafilters, the corresponding ultraproducts of models are always saturated. Assuming the Generalized Continuum Hypothesis, this leads to one of the most beautiful applications of saturated models to first order model theory, the Keisler-Shelah isomorphism theorem [21] saying that "two models are elementary equivalent if and only if they have isomorphic ultrapowers". Apart from its theoretical significance it has several applications, such as to axiomatizability and to interpolation. In [61] this is also developed at the level of abstract institutions.

1.1.5 *Axiomatizability*

In [34] axiomatizability is used as a cause for interpolation, and in [129] as a cause for definability. These are based upon the following institution theoretic generalization of Birkhoff style axiomatizability introduced in [34] (to which the notorious Birkhoff axiomatizability theorem in equational logic is a concrete instance).

Definition 1.11 (Birkhoff institution)**.** *[34]* $(\mathrm{Sign}, \mathrm{Sen}, \mathrm{Mod}, \models, \mathcal{F}, \mathcal{B})$ *is a* Birkhoff institution *when*

- $(\mathrm{Sign}, \mathrm{Sen}, \mathrm{Mod}, \models)$ *is an institution such that for each signature* $\Sigma \in |\mathrm{Sign}|$ *the category* $\mathrm{Mod}(\Sigma)$ *of* Σ*-models has* $\mathcal{F}$*-filtered products,*
- $\mathcal{F}$ *is a class of filters with* $\{\{*\}\} \in \mathcal{F}$*, and*
- $\mathcal{B}_\Sigma \subseteq |\mathrm{Mod}(\Sigma)| \times |\mathrm{Mod}(\Sigma)|$ *is a binary relation for each signature* $\Sigma \in |\mathrm{Sign}|$*, which is closed under isomorphisms, i.e.,* $(\mathcal{B}_\Sigma; \cong_\Sigma) = \mathcal{B}_\Sigma = (\cong_\Sigma; \mathcal{B}_\Sigma)$*,*

such that

$$\mathbb{M}^{**} = \mathcal{B}_\Sigma^{-1}(\mathcal{F}\mathbb{M})$$

for each signature Σ *and each class of* Σ*-models* $\mathbb{M} \subseteq |\mathrm{Mod}(\Sigma)|$*, and where* $\mathcal{F}\mathbb{M}$ *is the class of all* F*-filtered products of models from* $\mathbb{M}$ *for all filters* $F \in \mathcal{F}$*.*

In [34, 38] numerous concrete instances of this concept of axiomatizability are discussed. The monography [38] develops also general axiomatizability results for varieties and quasi-varieties of models, that constitute direct generalizations of corresponding results by Birkhoff [11] and Malcev [107], respectively.

1.1.6 *Interpolation*

Interpolation is one of the most studied properties in mathematical logic [139, 21], a recent monography dedicated to interpolation in modal and intuitionistic being [68]. Interpolation has numerous applications in computing science especially in formal specification theory [6, 59, 64, 63, 147, 14], but also in data bases (ontologies) [100], automated reasoning [123, 125], type checking [99], model checking [111], and structured theorem proving [3, 110]. In institution theory interpolation is considered in a form that generalizes several aspects of its common formulation. First, the signature inclusions that appear implicitly in the formulation of interpolation are abstracted to arbitrary signature morphisms. Then the common formulation of interpolation corresponds to the situation when the institution comes with the signature morphisms restricted to inclusions only. However the generalisation of interpolation to arbitrary signature morphism allows in the concrete situations for consideration of signature morphisms that may rename or even collapse syntactic entities. While such extended form of interpolation may be unusual in conventional logic, it is used in specification theory. A second generalisation of the concept of interpolation replaces individual sentences by finite sets of sentences. While in logics that have conjunction (such as classical propositional, first order logics, etc.) this does not mean anything, it is very meaningful in logics lacking

conjunctions, such as equational or Horn clause logics. In the latter ones interpolation may fail artificially due to unrealistic single sentence style formulation. These get us to the following definition of interpolation introduced in [34].

Definition 1.12 (Interpolation)**.** *[34] A commuting square of signature morphisms like below*

$$\begin{array}{ccc} \Sigma_0 & \xrightarrow{\varphi_1} & \Sigma_1 \\ {\scriptstyle \varphi_2}\downarrow & & \downarrow{\scriptstyle \theta^1} \\ \Sigma_2 & \xrightarrow[\theta_2]{} & \Sigma' \end{array}$$

is a Craig interpolation square *when for any finite sets* $E_1 \subseteq \mathrm{Sen}(\Sigma_1)$ *and* $E_2 \subseteq \mathrm{Sen}(\Sigma_2)$ *such that* $\mathrm{Sen}(\theta_1)(E_1) \models_{\Sigma'} \mathrm{Sen}(\theta_2)(E_2)$ *there exists a finite set* $E_0 \subseteq \mathrm{Sen}(\Sigma_0)$ *such that* $E_1 \models_{\Sigma_1} \mathrm{Sen}(\varphi_1)(E_0)$ *and* $\mathrm{Sen}(\varphi_2)(E_0) \models_{\Sigma_2} E_2$.

Such commuting squares are meant to emulate the intersection-union of signatures from the common formulation of interpolation, with Σ_0 in the role of $\Sigma_1 \cap \Sigma_2$ and Σ' in the role of $\Sigma_1 \cup \Sigma_2$. However in the common formulation of interpolation it is important that $\Sigma_1 \cup \Sigma_2$ is the lowest signature above Σ_1 and Σ_2. In the generalised formulation of interpolation this property appears as a category theoretic condition, that the commuting square is a *pushout* in the category Sign of the signatures of the institution. But when considering interpolation as a property of the institution as a whole, it is in general not meaningful to look for interpolation in all pushout squares. This leads to another generalisation layer in the formulation of interpolation, which restricts abstractly the range of φ_1 and φ_2 to designated subclasses of signature morphisms. Thus, given $\mathcal{L}$ and $\mathcal{R}$ subclasses of signature morphisms, the institution *has* $(\mathcal{L},\mathcal{R})$*-interpolation* when each pushout of a span (φ_1,φ_2) of signature morphisms with $\varphi_1 \in \mathcal{L}$ and $\varphi_2 \in \mathcal{R}$ is an interpolation square.

Institution theoretic interpolation has been established at the general level in relation to several different causes. One cause can be an axiomatizability property of the institution, like in [34]. A a typical example here is many sorted equational logic which has $(\mathcal{L},\mathcal{R})$-interpolation with $\mathcal{L}$ being the injective signature morphisms and $\mathcal{R}$ being all signature morphisms. Another cause can be the Robinson consistency property, like in [91]. An instance of this is many sorted first order logic which has $(\mathcal{L},\mathcal{R})$-interpolation when either $\mathcal{L}$ of $\mathcal{R}$ consists of signatures morphisms that are injective on the sorts. And yet another cause can be the existence of an adequate translation to an institution that has well established interpolation properties, like in [46].

The Craig interpolation property can be strengthened by adding to the 'primary' premises E_1 a set Γ_2 (of Σ_2-sentences) as 'secondary' premises. Craig-Robinson interpolation plays an important role in specification language theory, see [6, 59, 65]. The name 'Craig-Robinson' interpolation has been used for instances of this property in [139, 147, 65] and 'strong Craig interpolation' has been used in [59].

Definition 1.13 (Craig-Robinson interpolation)**.** *[38] In any institution we say that a commuting square of signature morphisms*

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

is a Craig-Robinson Interpolation square *(abbreviated* CRI *square) when for each set* E_1 *of* Σ_1*-sentences and each sets* E_2 *and* Γ_2 *of* Σ_2*-sentences, if* $\theta_1(E_1) \cup \theta_2(\Gamma_2) \models_{\Sigma'} \theta_2(E_2)$*, then there exists a set* E *of* Σ*-sentences such that* $E_1 \models_{\Sigma_1} \varphi_1(E)$ *and* $\Gamma_2 \cup \varphi_2(E) \models_{\Sigma_2} E_2$.

Also the $(\mathcal{L}, \mathcal{R})$-interpolation concept discussed above can be extended in a straightforward way from Craig interpolation to Craig-Robinson interpolation.

By taking Γ_2 to be the empty set $\emptyset$ we can see that any CRI square is also a CI square. The opposite implication does not hold in general. The following gives a sufficient condition when CI and CRI are equivalent interpolation concepts.

Proposition 1.14. *[38] In any compact institution that has implications, a commuting square of signature morphisms is a CRI square if and only if it is a CI square.*

1.1.7 *Definability*

One of the most important aspects of definability theory is the relationship between implicit and explicit definability. The following definition of the two kinds of definability at the level of abstract institutions has been introduced in [129].

Definition 1.15 (Definability)**.** *[129] Let* $\varphi : \Sigma \to \Sigma'$ *be a signature morphism and* E' *be a* Σ'*-theory. Then* φ

- *is* defined implicitly *by* E' *if the reduct functor*

$$\mathrm{Mod}(\Sigma', E') \longrightarrow \mathrm{Mod}(\Sigma') \xrightarrow{\mathrm{Mod}(\varphi)} \mathrm{Mod}(\Sigma)$$

  is injective, and

- *is* defined (finitely) explicitly *by* E' *if for each pushout square*

$$\begin{array}{ccc} \Sigma & \xrightarrow{\varphi} & \Sigma' \\ {\scriptstyle\theta}\downarrow & & \downarrow{\scriptstyle\theta'} \\ \Sigma_1 & \xrightarrow[\varphi_1]{} & \Sigma'_1 \end{array}$$

  and each sentence $\rho \in \mathrm{Sen}(\Sigma'_1)$*, there exists a (finite) set of sentences* $E_\rho \subseteq \mathrm{Sen}(\Sigma_1)$ *such that*

$$E' \models_{\Sigma'} (\forall \theta')(\rho \Leftrightarrow \varphi_1(E_\rho)).$$

A signature morphism φ has the (finite) definability property *if and only if a theory defines* φ *(finitely) explicitly whenever it defines* φ *implicitly.*

While in many institutions, that explicit implies implicit definability is immediate, at the level of abstract institutions this is non-trivial. For this to hold, in [129] it is shown that it is sufficient to impose only a rather mild restriction on signature morphisms, which in actual (many-sorted) situations requires only surjectivity of the sorts mapping.

The core of the institution-independent approach to definability consists of the study of the other much more difficult and meaningful implication, that implicit implies explicit definability. One result [129, 38] develops a generic definability theorem based upon the Craig-Robinson interpolation property, thus generalizing the well-known causality relationship between interpolation and definability in conventional model theory [21]. Another result [129, 38] has a complementary range of applications with respect to the previous one and is based upon general Birkhoff axiomatizability properties. The paper [129] presents a series of concrete instances of this general result in fragments of classical first order model theory and in partial algebra.

1.1.8 *Proof systems for institutional logic*

The main concept developed in [37] (originally introduced in [121]) refines the *entailment systems* of [112] to the situation when several different proofs between given sets of sentences are considered.

Definition 1.16 (Proof system)**.** *[37] A* proof system $(\mathrm{Sign}, \mathrm{Sen}, \mathrm{Pf})$ *consists of*

- *a category of 'signatures'* Sign*,*
- *a 'sentence functor'* $\mathrm{Sen}: \mathrm{Sign} \to \mathbf{Set}$*, and*
- *a 'proof functor'* $\mathrm{Pf}: \mathrm{Sign} \to \mathbf{CAT}$ *(giving for each signature* Σ *the category of the* Σ*-proofs)*

such that

1. $\mathrm{Sen}; \mathcal{P}; (-)^{\mathrm{op}}$ *is a sub-functor of* Pf*, and*
2. *the inclusion* $\mathcal{P}(\mathrm{Sen}(\Sigma))^{\mathrm{op}} \hookrightarrow \mathrm{Pf}(\Sigma)$ *is broad and preserves finite products of disjoint sets (of sentences) for each signature* Σ*, where* $\mathcal{P}: \mathbf{Set} \to \mathbf{CAT}$ *is the (*$\mathbf{CAT}$*-valued) power-set functor.*

The main results about proof systems developed in [37] are as follows:

1. The introduction of a formal concept of systems of proof rules for institutions and by showing that systems of proof rules generate freely proof systems. This adjunction result is based on an algebra of proofs. We argue that actual proof systems for institutions are freely generated by corresponding systems of proof rules.
2. The classical meta-rule of 'Generalization' is refined to a property of the proof system rather than considering it as a proof rule. We show that (universal) quantification can be added freely to any proof system such that its sentence part has a syntax for quantifiers, and argue that this is the way one obtains the actual proof systems with 'Generalization' as a meta-rule.
3. We show how these two universal properties lead to
   - the compactness of the proof system freely generated by a system of finitary proof rules,
   - the automatic transfer of compactness from proof systems to proof systems with universal quantification,
   - the automatic transfer of soundness from institutions with proof rules to institutions with proofs, and

- under the assumption of semantic quantification, further transfer of soundness to institutions with proof having proof-theoretic quantifiers.

These concepts and results have been used in several institution theoretic works on completeness such as [24, 89], etc.

A corresponding chapter of [38] contains a study of Boolean connectives for proof systems.

1.2 LOGIC COMBINATION

The entry on *Combining Logics* in the *Stanford Encyclopedia of Philosophy* [19] stresses the role of Computer Science applications as a main driving force for research in obtaining new logic systems from old, integrating features and preserving properties to a reasonable extent. In this section we present several works on logic combination in institution theory by the author. They can all be regarded as a generic and comprehensive (in the sense of addressing both the syntactic and the semantic levels) form of *hierarchical* logic combination, when the essential features of a logic are built on top of another logic.

1.2.1 *Possible worlds semantics in abstract institutions*

The development of semantics for modal logic on top of abstract institutions started with [62] and continued with the refinement to hybrid features in [108, 51]. The PhD thesis [106] was also developed in this area under the supervision of the author. Here we will present these developments in their refined hybrid logic form, called *hybridization of institutions*, its development being triggered by the concrete problem in (rigorous) software engineering, that of the formal specification of *reconfigurable* software systems.

The hybridisation method enriches a base (arbitrary) institution I with hybrid logic features and the corresponding Kripke semantics. The result is still an institution, $\mathcal{H}I$, called the *hybridisation of* I. This construction has several parameters, related to quantification and constraints on the Kripke semantics.

At the syntactic level the base signatures are enriched with nominals and polyadic modalities. Therefore, the category of *I-hybrid signatures*, denoted by $\mathrm{Sign}^{\mathcal{H}I}$, is defined as the direct (cartesian) product of categories of the original category of signatures Sign^{I} and that of signatures of *REL*, the sub-institution of (the institution of) first order logic, without non-constant operation symbols, Sign^{REL}.

The second step in the method is to enrich the base sentences accordingly. The sentences of the base institution I and the nominals in Nom are taken as atoms and composed with the boolean connectives, the modalities given in Λ, satisfaction operators indexed by nominals, and quantifiers by signature morphisms (from a designated quantification space $\mathcal{D}^{\mathcal{H}I}$).

Given a $\mathcal{H}I$-signature morphism φ, the translation of sentences $\mathrm{Sen}^{\mathcal{H}I}(\varphi)$ is defined structurally: e.g., $\mathrm{Sen}^{\mathcal{H}I}(\varphi)(i) = \varphi_{\mathrm{Nom}}(i)$, $\mathrm{Sen}^{\mathcal{H}I}(\varphi)(@_i\rho) = @_{\varphi_{\mathrm{Nom}}(i)}\mathrm{Sen}^{\mathcal{H}I}(\rho)$,
$\mathrm{Sen}^{\mathcal{H}I}(\varphi)([\lambda](\rho_1,\ldots,\rho_n)) = [\varphi_{\mathrm{MS}}(\lambda)](\mathrm{Sen}^{\mathcal{H}I}(\rho_1),\ldots,\mathrm{Sen}^{\mathcal{H}I}(\rho_n))$, etc.

Turning to semantics, models of $\mathcal{H}I$ can be regarded as (Λ-)Kripke structures whose worlds are I-models. Formally, they are pairs (M,W) where W is a (Nom,Λ)-model in *REL* and M is a function which assigns to

each state a model in $|\mathrm{Mod}^I(\Sigma)|$. In each world (M,W), W_n provides interpretation for nominal n, whereas relation W_λ interpretes modality λ. The reduct definition is lifted from the base institution: the reduct of a Δ'-model (M',W') along a signature morphism $\varphi:\ \Delta \to \Delta'$ is the Δ-model (M,W) such that W is the $(\varphi_{\mathrm{Nom}},\varphi_{\mathrm{MS}})$-reduct of W' (i.e, $|W| = |W'|$, $W_n = W'_{\varphi_{\mathrm{Nom}}(n)}$, for each nominal n, and $W_\lambda = W'_{\varphi_{\mathrm{MS}}(\lambda)}$ for each modality in Λ). In order to capture all kinds of constraints that may be around (the works on hybridization of institutions keep producing logics displaying a wide range of constraints, less and less conventional), in [51] the author has introduced a fully abstract formalization of constraints on Kripke structures simply as a sub-functor $\mathrm{Mod}^C \subseteq \mathrm{Mod}^{\mathcal{H}I}$, however subject to a technical property of reflecting model amalgamation.

Finally, the satisfaction relation for the hybridised institution resorts to the one in the base institution for sentences in I, i.e.,

Definition 1.17 (The Satisfaction Relation)**.** *Given a constrained model functor* $\mathrm{Mod}^C \subseteq \mathrm{Mod}^{\mathcal{H}I}$*, for any* $(M,W) \in |\mathrm{Mod}^C(\Sigma,\mathrm{Nom},\Lambda)|$ *and for any* $w \in |W|$ *we define:*

- $(M,W) \models^w \rho$ *iff* $M_w \models^I \rho$*; when* $\rho \in \mathrm{Sen}^I(\Sigma)$*,*
- $(M,W) \models^w i$ *iff* $W_i = w$*; when* $i \in \mathrm{Nom}$*,*
- $(M,W) \models^w \rho \vee \rho'$ *iff* $(M,W) \models^w \rho$ *or* $(M,W) \models^w \rho'$*,*
- $(M,W) \models^w \rho \wedge \rho'$ *iff* $(M,W) \models^w \rho$ *and* $(M,W) \models^w \rho'$*,*
- $(M,W) \models^w \rho \Rightarrow \rho'$ *iff* $(M,W) \models^w \rho$ *implies that* $(M,W) \models^w \rho'$*,*
- $(M,W) \models^w \neg\rho$ *iff* $(M,W) \not\models^w \rho$*,*
- $(M,W) \models^w [\lambda](\xi_1,\ldots,\xi_n)$ *iff for any* $(w,w_1,\ldots,w_n) \in W_\lambda$ *we have that* $(M,W) \models^{w_i} \rho_i$ *for some* $1 \leq i \leq n$*.*
- $(M,W) \models^w \langle\lambda\rangle(\xi_1,\ldots,\xi_n)$ *iff there exists* $(w,w_1,\ldots,w_n) \in W_\lambda$ *such that and* $(M,W) \models^{w_i} \xi_i$ *for any* $1 \leq i \leq n$*.*
- $(M,W) \models^w @_j\rho$ *iff* $(M,W) \models^{W_j} \rho$*,*
- $(M,W) \models^w (\forall\chi)\rho$ *iff* $(M',W') \models^w \rho$ *for any* (M',W') *such that* $\mathrm{Mod}^C(\chi)(M',W') = (M,W)$*,*
- $(M,W) \models^w (\exists\chi)\rho$ *iff* $(M',W') \models^w \rho$ *for some* (M',W') *such that* $\mathrm{Mod}^C(\chi)(M',W') = (M,W)$*, and*

We write $(M,W) \models \rho$ *iff* $(M,W) \models^w \rho$ *for any* $w \in |W|$*.*

The following result stated initially in [108] for a simplified context and then extended and proved in [51] shows that the result of this construction yields an institution.

Theorem 1.18. *[51] Assume* $\mathcal{D}^I$ *is adequate for* Mod^I*. Let* $\Delta = (\Sigma,\mathrm{Nom},\Lambda)$ *and* $\Delta' = (\Sigma',\mathrm{Nom}',\Lambda')$ *be two* $\mathcal{H}I$*-signatures and* $\varphi:\ \Delta \to \Delta'$ *a morphism of signatures. Given a constrained model functor* $\mathrm{Mod}^C \subseteq \mathrm{Mod}^{\mathcal{H}I}$*, for any* $\rho \in \mathrm{Sen}^{\mathcal{H}I}(\Delta)$*,* $(M',W') \in |\mathrm{Mod}^C(\Delta')|$*, and* $w \in |W|$

$$\mathrm{Mod}^C(\varphi)(M',W')(= \mathrm{Mod}^{\mathcal{H}I}(\varphi)(M',W')) \models^w \rho \text{ if and only if } (M',W') \models^w \mathrm{Sen}^{\mathcal{H}I}(\varphi)(\rho). \tag{2}$$

Corollary 1.19 (The Satisfaction Condition)**.** *[51]* $(\mathrm{Sign}^{\mathcal{H}I},\mathrm{Sen}^{\mathcal{H}I},\mathrm{Mod}^C,\models)$ *is an institution.*

1.2.2 *Institutional semantics for many valued logics*

The paper [49] builds on the idea that the essence of many-valued logic (mvl) is actually independent of the concrete syntactic context in which it is usually presented, that in fact it is independent of *any* syntactic context. The technical side of this abstract mvl development may be briefly and informally described as follows. Given an abstract category of signatures and an abstract sentence functor (that gives the sets of sentences corresponding to the signatures) we build both a syntax and a model theory as well as a satisfaction relation between them.

The given syntax is considered as atomic syntax and the full syntax is built by iterative applications of connectives ($\wedge$,$\vee$,$\Rightarrow$, etc.) and quantifiers. Quantifiers are also treated rather abstractly by the concept of *quantification space* of [40]; this covers concrete quantification situations much beyond first order.

Definition 1.20. *[130] A* logic syntax *is a pair* $(\mathrm{Sign}, \mathrm{Sen})$ *such that*

1. Sign *is a category whose objects are called 'signatures' and whose arrows are called 'signature morphisms', and*
2. *a functor* $\mathrm{Sen}: \mathrm{Sign} \to \mathbf{Set}$ *called 'sentence functor'; the objects of* $\mathrm{Sen}(\Sigma)$ *are called '*Σ*-sentences'.*

Given a residuated lattice L and a tuple $(\mathrm{Sign}, \mathsf{FSen}, \mathsf{CSen}, \mathsf{CSen}_0, \mathcal{D})$ such that

1. $(\mathrm{Sign}, \mathsf{FSen})$ is a logic syntax (called the *fuzzy atomic syntax*),
2. $(\mathrm{Sign}, \mathsf{CSen})$ is a logic syntax (called the *crisp atomic syntax*),
3. the fuzzy and the crisp atomic syntaxes are disjoint, i.e. for each signature Σ,
$$\mathsf{FSen}(\Sigma) \cap \mathsf{CSen}(\Sigma) = \emptyset,$$
4. $\mathsf{CSen}_0 \subseteq \mathsf{CSen}$ is a sub-functor (called the *crisp truth functor*), and
5. $\mathcal{D}$ is a quantification space for Sign

Definition 1.21 ($I(L)$ sentences)**.** *[49] Let* Sen^* *be the least mapping* $\mathrm{Sign} \to \mathbf{Set}$ *such that for each signature* Σ

- $\mathsf{FSen}(\Sigma) \cup \mathsf{CSen}(\Sigma) \subseteq \mathrm{Sen}^*(\Sigma)$,
- $\top, \bot \in \mathrm{Sen}^*(\Sigma)$,
- $\rho_1 \star \rho_2 \in \mathrm{Sen}^*(\Sigma)$ *for* $\star \in \{\wedge, \vee, \Rightarrow, \otimes\}$ *and for all* $\rho_1, \rho_2 \in \mathrm{Sen}^*(\Sigma)$*, and*
- $(\forall \chi)\rho$, $(\exists \chi)\rho \in \mathrm{Sen}^*(\Sigma)$ *for all* $\rho \in \mathrm{Sen}^*(\Sigma')$ *and each* $(\chi: \Sigma \to \Sigma') \in \mathcal{D}$,

and for each signature morphism $\varphi: \Sigma \to \Sigma_1$

- $\mathrm{Sen}^*(\varphi)(\rho) = \mathsf{FSen}(\varphi)(\rho)$ *for each* $\rho \in \mathsf{FSen}(\Sigma)$,
- $\mathrm{Sen}^*(\varphi)(\top) = \top$, $\mathrm{Sen}^*(\varphi)(\bot) = \bot$,
- $\mathrm{Sen}^*(\varphi)(\rho) = \mathsf{CSen}(\varphi)(\rho)$ *for each* $\rho \in \mathsf{CSen}(\Sigma)$,
- $\mathrm{Sen}^*(\varphi)(\rho_1 \star \rho_2) = \mathrm{Sen}^*(\varphi)(\rho_1) \star \mathrm{Sen}^*(\varphi)(\rho_2)$ *for* $\star \in \{\wedge, \vee, \Rightarrow, \otimes\}$ *and for all* $\rho_1, \rho_2 \in \mathrm{Sen}^*(\Sigma)$*, and*
- $\mathrm{Sen}^*(\varphi)((\forall \chi)\rho) = (\forall \chi(\varphi))\mathrm{Sen}^*(\varphi[\chi])(\rho)$ *for each* $\rho \in \mathrm{Sen}^*(\Sigma')$ *and similarly for* $(\exists \chi)\rho$.

Let $\mathrm{Sen}^{I(L)} = \mathrm{Sen}^* \times L$ *be the mapping* $\mathrm{Sign} \to \mathbf{Set}$ *defined by*

- $\mathrm{Sen}^{I(L)}(\Sigma) = \mathrm{Sen}^*(\Sigma) \times L$ *for each signature* Σ*, and*
- $\mathrm{Sen}^{I(L)}(\varphi)(\rho,x) = (\mathrm{Sen}^*(\varphi)(\rho),x)$ *for each signature morphism* $\varphi : \Sigma \to \Sigma_1$*, each* Σ*-quasi-sentence* ρ *and each* $x \in L$*.*

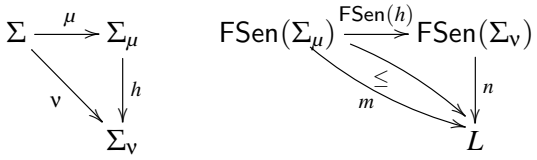
The model theory is defined generically by a comma category construction plus an interpretation of a corresponding atomic syntax into a fixed space of truth values considered here as a fixed complete residuated lattice L.

For each signature, the corresponding satisfaction relation between its models and its sentences is defined by recursion on the structure of the sentences in the usual Tarski style.

Definition 1.22 ($I(L)$ models)**.** *[49] For any signature* Σ*, a* Σ-model *is a pair* (μ,m) *that consists of*

- *a signature morphism* $\mu : \Sigma \to \Sigma_\mu$*, and*
- *a function* $m : \mathsf{FSen}(\Sigma_\mu) \to L$*.*

A Σ-homomorphism $h : (\mu,m) \to (\nu,n)$ *between two* Σ*-models is a signature morphism* $h : \Sigma_\mu \to \Sigma_\nu$ *such that* $\mu;h = \nu$ *and* $m \leq \mathsf{FSen}(h);n$*.*



The category of the Σ*-homomorphisms (composition inherited from* Sign*) is denoted* $\mathrm{Mod}^{I(L)}(\Sigma)$*.*

For any signature morphism $\varphi : \Sigma \to \Sigma'$ *the* φ-reduct $\mathrm{Mod}^{I(L)}(\varphi)(\mu',m')$ *of a* Σ'*-model* (μ',m') *is* $(\varphi;\mu',m')$*. The* φ*-reduct* $\mathrm{Mod}^{I(L)}(\varphi)(h')$ *of a* Σ'*-homomorphism* $h' : (\mu',m') \to (\nu',n')$ *is just* $h' : (\varphi;\mu',m') \to (\varphi;\nu',n')$*.*

In the concrete situations the component μ from the definition of $I(L)$ models is usually a signature extension with a set of constants representing the carrier (or underlying) set of the respective model. The component m interprets the atomic syntax corresponding to the extended signature into the space L of the truth values. Note that the crisp atomic syntax does not play any role in the semantics of $I(L)$.

Definition 1.23 ($I(L)$ satisfaction)**.** *[49] For each signature* Σ *we define a 'satisfaction degree' function* $(\_ \models \_) : |\mathrm{Mod}^{I(L)}(\Sigma)| \times \mathrm{Sen}^*(\Sigma) \to L$ *by*

- *for each* $\rho \in \mathsf{FSen}(\Sigma)$*,* $((\mu,m) \models \rho) = m(\mathsf{FSen}(\mu)(\rho))$*,*
- *for each* $\rho \in \mathsf{CSen}(\Sigma)$*,* $((\mu,m) \models \rho) = \begin{cases} \top & \text{when } \mathsf{CSen}(\mu)(\rho) \in \mathsf{CSen}_0(\Sigma_\mu), \\ \bot & \text{otherwise.} \end{cases}$
- $((\mu,m) \models \top) = \top$ *and* $((\mu,m) \models \bot) = \bot$*,*
- $((\mu,m) \models \rho_1 \star \rho_2) = ((\mu,m) \models \rho_1) \star ((\mu,m) \models \rho_2)$ *for* $\star \in \{\wedge,\vee,\Rightarrow,\otimes\}$*,*
- $((\mu,m) \models (\forall\chi)\rho) = \bigwedge\{(\mu',m) \models \rho \mid \mu' : \Sigma' \to \Sigma_\mu, \mu = \chi;\mu'\}$*, and*

– $((\mu,m) \models (\exists\chi)\rho) = \bigvee\{(\mu',m) \models \rho \mid \mu' :\ \Sigma' \to \Sigma_\mu, \mu = \chi;\mu'\}$.

Then for any Σ*-model* (μ,m) *and any sentence* $\rho \in \mathrm{Sen}^{I(L)}(\Sigma)$,

$$(\mu,m) \models_\Sigma^{I(L)} (\rho,k) \ \textit{ if and only if } \ k \leq ((\mu,m) \models \rho).$$

In [49] it is shown that this construction yields a(n ordinary binary) institution in which the sentences are pairs consisting of an ordinary sentence and a truth value. This is an immediate consequence of the many-valued satisfaction condition below:

Proposition 1.24. *[49] For any signature morphism* $\varphi :\ \Sigma \to \Sigma_1$*, any* Σ_1*-model* (μ,m)*, and any* $\rho \in \mathrm{Sen}^*(\Sigma)$

$$((\varphi;\mu,m) \models \rho) = ((\mu,m) \models \mathrm{Sen}^*(\varphi)(\rho)).$$

The generic abstract mvl thus developed (denoted $I(L)$) covers various important concrete mvl systems. In [49] its is shown that the semantics of traditional first-order mvl (such as in [92], denoted *MVL*) may be conservatively embedded into $I(L)$, which means that in this case the semantic consequence relation provided (generically) by $I(L)$ coincides with that of *MVL*. This applies also to various restrictions or extensions of *MVL*, i.e. propositional fragment, second order extension, etc. A similar result is shown for a rather natural fuzzy extension of the multi-algebras framework [95, 149, 150, 102].

Moreover our generic abstract mvl development may be used as a semantic oriented framework for defining in an uniform way *new* concrete many-valued logical systems over various different logic syntaxes. This also means an uniform semantic oriented method to export mvl to other logical contexts.

1.2.3 *Hidden preordered algebra*

In heterogenous specification frameworks it is crucial that any two logical formalisms involved have a 'least upper bound', which should appear as a 'super-logic' to both of them. Thus in the case of a system like CafeOBJ one needs to study such a combination between hidden algebra (*HA*), i.e. the logic underlying the behavioural specification paradigm, and preordered algebra (*POA*). This, denoted *HPOA* has been defined in model theoretic terms in [54, 56] (although only the latter reference constitutes the definitive solution to this combination problem). That *HPOA* is more than just putting *POA* and *HA* together is clear from the concept of behavioural transition which is syntactically supported in the CafeOBJ language as by the keyword `btrans`.

The paper [41] gives the definitive solution to the hierarchical combination between *POA* and *HA* and also provides a coinduction-like proof method for behavioural transitions which extends the well known coinduction for proving behavioural equivalences. The novel contributions that emerged from these works include:

1. A novel concept of congruence for preordered algebras.
   Definition 1.25 (Preordered algebra congruences)**.** *[41] A POA*-congruence *(preordered algebra congruence) on a preordered algebra* $(M,\leq)$ *for a signature* (S,F) *is a pair* $(\sim,\sqsubseteq)$ *such that*

– $\sim$ *is an* (S,F)*-congruence on the* (S,F)*-algebra* M*,*

– $\sqsubseteq$ *is a (n S-sorted) preorder on* M *which contains* $\leq$*, i.e.* $\leq \subseteq \sqsubseteq$*, and which is compatible with the operations, and*

– $a' \sim a, a \sqsubseteq b, b \sim b'$ *implies* $a' \sqsubseteq b'$ *for all elements* a, a', b, b' *of* M*.*

2. The development of the institution of hidden preoredered algebra (*HPOA*) based on the a theorem of the existence of the largest hidden preordered congruence for any algebra:
Definition 1.26 (Hidden *POA* congruence)**.** *[41] A* hidden *POA*-congruence *on a HPOA-algebra* $(A,\leq)$ *is a POA* $(H \cup V, BF)$*-congruence* $(\equiv, \sqsubseteq)$ *on* $(A,\leq)$ *such that on the visible sorts* $\equiv$ *is the identity and* $\sqsubseteq$ *is* $\leq$*.*
Definition 1.27 (Behavioural *POA* congruence)**.** *[41] The* behavioural *POA*-congruence *on* $(A,\leq)$ *is the which is the largest hidden POA-congruence on* $(A,\leq)$*.*
Theorem 1.28. *[41] Behavioural POA-congruence exists for any preordered* (H,V,F,BF)*-algebra.*

3. The extraction of a coinduction principle for behavioural transitions as an extension of the well known coinduction principle for behavioural equations [75]:
Corollary 1.29 (Coinduction for behavioural transitions)**.** *[41] The coinduction proof method for HPOA consists of the following steps.*
   a) Define an equivalence relation R and a preorder relation P for each hidden sort such that

   $$(s\ P\ s') \text{ and } (s\ R\ s1) \text{ and } (s'\ R\ s'1) \text{ imply } (s1\ P\ s'1)$$

   and

   $$(s \dashrightarrow s') \text{ implies } (s\ P\ s')$$

   for all `s`, `s'`, `s1`, `s'1`,
   b) Prove that both R and P are preserved by the behavioural operations,
   c) i. If we want to prove that $t \sim t'$ *then we show that* `t R t'`*, and*
   ii. If we want to prove that $t \sim\!> t'$ *then we show that* `t P t'`*.*

1.3 LOGIC BY TRANSLATION

Logic by translation, which means borrowing of logical properties across suitable translations between logical systems has emerged as an important method in mathematical logic, much associated with the universal trend in logic. Institution theory has developed its own logic by translation technology that owes to a solid concept of homomorphism between institutions, in the literature called 'comorphism' [83]. Institution comorphisms are mappings which preserve the mathematical structure of institutions, but from a logic and model theoretic perspective their significance is that of either embeddings or encodings between institutions. An important feature of comorphisms is that they relate between institutions both at the syntactic and at the semantic level, and both the syntactic and the semantic components of comorphisms are coherent to each other.

Definition 1.30 (Comorphisms)**.** *An* institution comorphism $(\Phi,\alpha,\beta):\ I \to I'$ *consists of*

1. *a functor* $\Phi:\ \mathrm{Sign} \to \mathrm{Sign}'$,
2. *a natural transformation* $\alpha:\ \mathrm{Sen} \Rightarrow \Phi;\mathrm{Sen}'$, *and*
3. *a natural transformation* $\beta:\ \Phi^{\mathrm{op}};\mathrm{Mod}' \Rightarrow \mathrm{Mod}$

such that the following satisfaction condition *holds*

$$M' \models'_{\Phi(\Sigma)} \alpha_\Sigma(e) \ \ \textit{iff}\ \ \beta_\Sigma(M') \models_\Sigma e$$

for each signature $\Sigma \in |\mathrm{Sign}|$, *for each* $\Phi(\Sigma)$*-model* M', *and each* Σ*-sentence* e.

Encodings between logics are represented as comorphism $I \to I'^{\mathrm{th}}$ where I'^{th} denotes the canonically defined institution of I'-theories, that has as signatures the pairs (Σ,E) where Σ is an I'-signature and E a set of Σ-sentences. The comorphism $I \to I'^{\mathrm{th}}$ are sometimes called *theoroidal* comorphisms.

The institution theoretic approach to logic by translation has achieved general recognition through the paper [120] that won the contest 'What is a logic translation?' at the *2nd World Congress on Universal Logic*, Xi'an, China, 2007.

In this section we present several borrowing results across institution comorphisms developed by the author.

1.3.1 *Borrowing interpolation*

The institution theoretic method for establishing interpolation developed in [46] can be described as follows: given an institution comorphism $I \to I'$, if I' has a certain interpolation property and the institution comorphism 'behaves well' with respect to interpolation, then I can be established to have a corresponding interpolation property. The properties required for the institution comorphisms have been originally introduced in [35]:

Definition 1.31 (Left interpolation property for comorphisms)**.** *[35] For a fixed class* $\mathcal{S} \subseteq \mathrm{Sign}$ *of signature morphisms, we say that an institution comorphism* $(\Phi,\alpha,\beta):\ I \to I'$ *has the* Craig $\mathcal{S}$-left Interpolation property *when for each* $(\varphi_1:\ \Sigma \to \Sigma_1) \in \mathcal{S}$, *for each set* E_1 *of* Σ_1*-sentences and each set* E_2 *of* $\Phi(\Sigma_1)$*-sentences such that* $\alpha_{\Sigma_1}(E_1) \models' \Phi(\varphi_1)(E_2)$, *there exists a set of* Σ*-sentences* E *such that* $E_1 \models \varphi_1(E)$ *and* $\alpha_\Sigma(E) \models' E_2$.

$$\begin{array}{ccc} \mathrm{Sen}(\Sigma) & \xrightarrow{\mathrm{Sen}(\varphi_1)} & \mathrm{Sen}(\Sigma_1) \\ \downarrow{\scriptstyle \alpha_\Sigma} & & \downarrow{\scriptstyle \alpha_{\Sigma_1}} \\ \mathrm{Sen}'(\Phi(\Sigma)) & \xrightarrow[\mathrm{Sen}'(\Phi(\varphi_1))]{} & \mathrm{Sen}'(\Phi(\Sigma_1)) \end{array}$$

The following is the reflection in the mirror of the left property.

Definition 1.32 (Right interpolation property for comorphisms)**.** *[35] For a fixed class* $\mathcal{S} \subseteq \mathrm{Sign}$ *of signature morphisms, we say that an institution comorphism* $(\Phi,\alpha,\beta):\ I \to I'$ *has the* Craig $\mathcal{S}$-right Interpolation

property *when for each* $(\varphi_2 : \Sigma \to \Sigma_2) \in \mathcal{S}$*, for each set* E_1 *of* $\Phi(\Sigma)$*-sentences and each set* E_2 *of* Σ_2*-sentences such that* $(\Phi(\varphi_2)(E_1) \models' \alpha_{\Sigma_2}(E_2)$*, there exists a set of* Σ*-sentences* E *such that* $E_1 \models \alpha_\Sigma(E)$ *and* $\varphi(E) \models' E_2$.

$$\begin{array}{ccc} \mathrm{Sen}(\Sigma) & \xrightarrow{\alpha_\Sigma} & \mathrm{Sen}'(\Phi(\Sigma)) \\ {\scriptstyle \mathrm{Sen}(\varphi_2)}\downarrow & & \downarrow{\scriptstyle \mathrm{Sen}'(\Phi(\varphi_2))} \\ \mathrm{Sen}(\Sigma_2) & \xrightarrow[\alpha_{\Sigma_2}]{} & \mathrm{Sen}'(\Phi(\Sigma_2)) \end{array}$$

The main borrowing interpolation theorem is:

Theorem 1.33 (Borrowing interpolation)**.** *[46] Let* $(\Phi, \alpha, \beta) : I \to I'$ *be a conservative institution comorphism such that* Φ *maps pushouts to quasi-pushouts, and let* $\mathcal{L}, \mathcal{R} \subseteq \mathrm{Sign}$ *be classes of signature morphisms such that* I' *has the Craig* $(\Phi(\mathcal{L}), \Phi(\mathcal{R}))$*-interpolation. Then* I *has Craig* $(\mathcal{L}, \mathcal{R})$*-interpolation.*

This generic result can be applied to various situations, requiring different styles of applying it. One situation is when I and I' have essentially the same expressive power, such as (first order) partial algebra and first order logic. Although partial algebra is more refined than first order logic, it can be encoded into first order logic in a rather strong way which makes possible a quite straightforward transfer of first order logic interpolation to partial algebra. The second situation we study is when I has significantly less expressive power than I', which means that their interpolation properties are rather different and moreover are obtained in different ways. This second situation is well illustrated by the logic of universal sentences versus first order logic, which is an embedding rather than an encoding. In this case the application of the main borrowing theorem requires more subtlety. Finally, a third situation is when I has more expressive power than I' but an encoding of I into I' by means of a comorphism exists. This is well illustrated by higher order logic in the role of I and first order logic in the role of I'.

The general borrowing interpolation result of [46], together with its above mentioned associated styles or patterns for applying it, constitute a new method within the rather large spectrum of methods to obtain interpolation. Taking into account the great difficulty of the problem, the existing methods are never enough. The strength of the borrowing method of [46] has at least several aspects to be mentioned. New interpolation results have been obtained in [46] for preordered algebras and for higher order logic in a rather smooth way. For many of the applications the alternatives seem to be much more difficult. For example, in partial or preordered algebra, based on some similarity with classical many-sorted first order logic, we may expect certain interpolation properties. We either have to replicate the whole complicated conceptual infrastructure of the classical framework to these new frameworks (which is very complex and even problematic task) or else to lift such infrastructure to a higher abstraction level as in [91, 34] (which is also a difficult task but in a different way). The method of [46] saves such efforts. Moreover in other situations (such as higher order logic) perhaps none of the above mentioned alternatives would work. The borrowing method of [46] has been applied not only for obtaining new results but also for a better understanding of already know results in partial algebra and even in fragments of first order logic. All these are related to a deeper aspect, that of offering a rather clear, although unavoidably partial, insight into the interdependent nature of interpolation properties in various logical systems.

1.3.2 *Borrowing definability*

A way to establish definability properties alternative to that by reliance upon interpolation or axiomatizability is by borrowing across an institution comorphisms. The following abstract necessary condition has be identified in [129]:

Definition 1.34. *[129] We say that an I-signature morphism $\varphi : \Sigma_1 \to \Sigma_2$ is* (Φ,α,β)-precise *whenever the function* $\mathrm{Mod}'(\Phi(\Sigma_2)) \to \mathrm{Mod}'(\Phi(\Sigma_1)) \times \mathrm{Mod}(\Sigma_2)$ *mapping each M'_2 to $\langle M'_2{\restriction_{\Phi(\varphi)}}, \beta_{\Sigma_2}(M'_2)\rangle$ is injective. We say that the comorphism (Φ,α,β) is* precise *when each I-signature morphism is (Φ,α,β)-precise.*

The following result from [129] establishes implicit definability by borrowing:

Theorem 1.35 (Borrowing implicit definability)**.** *[129] Let $(\Phi,\alpha,\beta) : I \to I'$ be an institution comorphism. For any (Φ,α,β)-precise signature morphism φ and theory E', $\Phi(\varphi)$ is defined implicitly by $\alpha(E')$ if φ is defined implicitly by an E'.*

The following result from [129] establishes explicit definability by borrowing:

Theorem 1.36 (Borrowing explicit definability)**.** *[129] Let $(\Phi,\alpha,\beta) : I \to I'$ be an institution comorphism. If*

1. *$(\Phi,\alpha,\beta) : I \to I'$ is conservative,*
2. *Φ preserves pushouts, and*
3. *α is surjective modulo the semantic equivalence $\models\!\!\models$,*

then any I-signature morphism φ is defined (finitely) explicitly by a theory E' if $\Phi(\varphi)$ is defined (finitely) explicitly by $\alpha(E')$.

The borrowing of implicit and explicit definability can be put together in order to obtain the borrowing of Beth definability in abstract institutions:

Corollary 1.37 (Borrowing of Beth definability)**.** *[129] Under the assumptions of Thm. 1.36, any (Φ,α,β)-precise signature morphism φ has the definability property if $\Phi(\varphi)$ has the definability property.*

In [129] this is illustrated with definability results in partial algebra borrowed from classical first order logic.

1.3.3 *Borrowing saturated models*

The general institution theoretic result on the existence of saturated models of [61] relies upon existential quantifiers and conjunctions. The existence of saturated models can be extended to sub-institutions with less expressive power of sentences by the following general result of [61].

Proposition 1.38. *Let $(\Phi,\alpha,\beta) : I \to I'$ be an institution comorphism and $\mathcal{D} \subseteq \mathrm{Sign}$, $\mathcal{D}' \subseteq \mathrm{Sign}'$ be classes of signature morphisms such that*

1. (Φ, α, β) *is conservative and has weak model amalgamation, and*
2. Φ *preserves inductive co-limits and* $\Phi(\mathcal{D}) \subseteq \mathcal{D}'$.

Then I *has* $\mathcal{D}$*-saturated models whenever* I' *has* $\mathcal{D}'$*-saturated models.*

In [38] this result has been instantiated to obtain the existence of saturated models in fragments of first order logic, such as Horn clause logic and equational logic.

1.3.4 *Encoding partial algebras as total algebras*

Partial functions play an important role in computing science; this is well know. In particular, the specification power of operations that are partially defined makes partial algebra as one of the important formalisms employed by modern formal methods, a prominent example being the recent algebraic specification language CASL [5]. However, the current mathematical culture, including school mathematics that is responsible for the basic patterns of our mathematical thinking, is strongly biased towards total functions. Reasons are manifold. First of all, algebraic reasoning with total functions is much simpler. Then, the semantics of partial functions is significantly more sophisticated than that of ordinary total algebra. These above two aspects are of course interdependent. Consequently, mechanical algebraic reasoning with total functions is supported by a rather impressive variety of tools and execution engines, most of them based upon the so-called term rewriting method, while partial algebra formalism lacks such kind of computational infrastructure.

The above mentioned reasons have led to various efforts to translate, or encode, partial algebra into logics with total functions with the benefit of doing things in the translation and exporting the results back to the original partial algebra framework. Several translations partial algebra into logics with total functions have been proposed in the literature, the most prominent one appearing in [118] encodes partial operations as total operations but employs also relations for expressing the domains of definition of the partial operations. Another important one [118, 129] encodes the partial operations as relation symbols.

The work [39] proposes a novel translation that, unlike the previous ones, has a pure algebraic nature since it does employ only operation symbols, no relation symbols. Thus any partial algebraic signature gets encoded as a set of conditional equations for a total algebraic signature. This translation is defined as theoroidal comorphism from partial algebra to equational logic with total functions as follows.

Definition 1.39. *[39] Each PA signature* (S, TF, PF) *gets mapped to an MSA theory* $((S \cup \{\mathbf{b}\}, TF \oplus PF), \Gamma_{(S,TF,PF)})$ *where*

- $(TF \oplus PF)_{w \to s} = TF_{w \to s} \cup PF_{w \to s}$ *when* $s \neq \mathbf{b}$,
- $(TF \oplus PF)_{ss \to \mathbf{b}} = \{\Subset_s\}$ *for each* $s \in S$, *and*
- $(TF \oplus PF)_{\to \mathbf{b}} = \{\mathsf{true}\}$.

and $\Gamma_{(S,TF,PF)}$ *contains the following conditional equations:*

1. $(\forall X)(X \Subset X = \mathsf{true}) \Rightarrow (\sigma(X) \Subset \sigma(X) = \mathsf{true})$ *for any* total *operation symbols* σ *and* X *a string of variables matching the arity of* σ.[1]

[1] If $X = \{x_1, \ldots, x_n\}$ then $X \Subset X$ denotes the finite conjunction $(x_1 \Subset x_1) \wedge \cdots \wedge (x_n \Subset x_n)$.

2. $(\forall X,Y)(X \mathbin{\widehat{\in}} Y = \mathsf{true}) \Rightarrow (X \mathbin{\widehat{\in}} X = \mathsf{true})$.
3. $(\forall X,Y)(X \mathbin{\widehat{\in}} Y = \mathsf{true}) \Rightarrow (X = Y)$.
4. $(\forall X)(\sigma(X) \mathbin{\widehat{\in}} \sigma(X) = \mathsf{true}) \Rightarrow (X \mathbin{\widehat{\in}} X = \mathsf{true})$ *for any* total *or* partial *operation symbols.*

Definition 1.40. *[39] Given a PA signature (S,TF,PF), the sentence translation $\alpha_{(S,TF,PF)}$ maps each (S,TF,PF)-sentence to a $(S \cup \{\mathbf{b}\}, TF \oplus PF)$-sentence as follows:*

- $\alpha_{(S,TF,PF)}(t \stackrel{e}{=} t') = (t \mathbin{\widehat{\in}} t' = \mathsf{true})$ *for any terms t and t' of the same sort.*
- $\alpha_{(S,TF,PF)}$ *commutes with all Boolean connectives, i.e.* $\alpha_{(S,TF,PF)}(\rho_1 \wedge \rho_2) = \alpha_{(S,TF,PF)}(\rho_1) \wedge \alpha_{(S,TF,PF)}(\rho_2)$, *etc.*
- $\alpha_{(S,TF,PF)}((\forall X)\rho) = (\forall X)((X \mathbin{\widehat{\in}} X) \Rightarrow \alpha_{(S,TF \cup X,PF)}(\rho))$.

The following gives the semantic part of the translation.

Definition 1.41. *A functor $\beta_{(S,TF,PF)} : MSA(S \cup \{\mathbf{b}\}, TF \oplus PF, \Gamma_{(S,TF,PF)}) \to PA(S,TF,PF)$ (often denoted simply by β when there is no danger of confusion) is defined as follows:*

- *For each $(S \cup \{\mathbf{b}\}, TF \oplus PF)$-algebra A satisfying $\Gamma_{(S,TF,PF)}$,*
  - $\beta(A)_s = \{a \in A_s \mid A_{\widehat{\in}_s}(a,a) = A_{\mathsf{true}}\}$ *for each sort $s \in S$,*
  - $\beta(A)_\sigma(a) = A_\sigma(a)$ *when $A_{\widehat{\in}_s}(A_\sigma(a),A_\sigma(a)) = A_{\mathsf{true}}$ for any operation symbol $\sigma \in TF_{w \to s} \cup PF_{w \to s}$.*
  - $\beta(A)_\sigma(a)$ *is undefined when $A_{\widehat{\in}_s}(A_\sigma(a),A_\sigma(a)) \neq A_{\mathsf{true}}$ for any operation symbol $\sigma \in TF_{w \to s} \cup PF_{w \to s}$.*

  Note that if $\beta(A)_\sigma(a)$ is defined then $\beta(A)_\sigma(a) \in \beta(A)$.
- *For each $(S \cup \{\mathbf{b}\}, TF \oplus PF)$-algebra homomorphism $h : A \to B$ we define the homomorphism of partial algebras $\beta(h) : \beta(A) \to \beta(B)$ defined by $\beta(h)(a) = h(a)$ for each $a \in \beta(A)$.*

Theorem 1.42 (Satisfaction condition)**.** *[39] For each PA signature (S,TF,PF), for each $(S \cup \{\mathbf{b}\}, TF \oplus PF)$-algebra A satisfying Γ, and for each (S,TF,PF)-sentence ρ*

$$A \models \alpha(\rho) \text{ if and only if } \beta(A) \models \rho.$$

In [39] several properties of this encoding have been developed, including that the model translations admit free constructions such that their universal homomorphisms are identities. Important consequence of this result include initial semantics for the Horn fragment of partial algebra (result already known but here obtained by borrowing from many sorted algebra) and the reflection of the semantic consequence across the translation. The latter leads to an important proof theoretic consequence: a sound and complete calculus for the Horn fragment of partial algebra can be expressed as ordinary equational calculus with total operations symbols. This provides a simple and efficient way to execute Horn partial algebraic specifications by ordinary term rewriting.

1.3.5 *Encoding hybridized institutions into first-order logic*

Motivated by a methodology for the formal specification and verification of reconfigurable systems, the work [60] which represents the core part of the PhD thesis [106] supervised by the author of this habilitation thesis, extends the traditional translation of modal logic to *FOL* [145] (for the hybrid variant [12]) to encodings of abstract hybridized institutions into *FOL*. This may also be regarded as 'hybridizations' of encodings into *FOL*. The idea to 'hybridize comorphisms' has been sketched within a much a preliminary form in [108], in [60] we have extended this in several directions: constrained models, theoroidal comorphisms (rather than plain comorphisms), and quantified sentences. This main elements of this rather elaborate encoding are given in what follows.

Notation 1.3.1. *[60] For any FOL-signature (S,F,P) we denote by $([S],[F],[P])$ the following FOL-signature:*

- $[S] = S \cup \{\mathrm{ST}\}$, *where* ST *is a designated sort not in S,*
- $[F]_{\underline{ar}\to s} = \begin{cases} F_{\underline{ar}'\to s} & \text{for any } s \in S, \underline{ar}' \in S^* \text{ such that } \underline{ar} = (\mathrm{ST})\underline{ar}' \\ \emptyset & \text{for the other cases;} \end{cases}$
- $[P]_{\underline{ar}} = \begin{cases} P_{\underline{ar}'} & \text{for any } \underline{ar}' \in S^* \setminus S \text{ such that } \underline{ar} = (\mathrm{ST})\underline{ar}'; \\ \emptyset, & \text{for the other cases.} \end{cases}$

For any morphism of FOL signatures $\varphi : (S,F,P) \to (S',F',P')$ we let $[\varphi] : ([S],[F],[P]) \to ([S'],[F'],[P'])$ morphism of FOL signatures defined as follows:

- $[\varphi]^{\mathrm{st}}(\mathrm{ST}) = \mathrm{ST}$,
- $[\varphi]^{\mathrm{st}}(s) = \varphi^{\mathrm{st}}(s)$ *for any* $s \in S$,
- $[\varphi]^{\mathrm{op}}_{(\mathrm{ST})\underline{ar}'\to s}(\sigma) = \varphi^{\mathrm{op}}_{\underline{ar}'\to s}(\sigma)$ *for any* $\sigma \in F_{\underline{ar}'\to s}$, *and*
- $[\varphi]^{\mathrm{rl}}_{(\mathrm{ST})\underline{ar}'}(\pi) = \varphi^{\mathrm{rl}}_{\underline{ar}'}(\pi)$ *for any* $\pi \in P_{\underline{ar}'}$.

Definition 1.43. *[60] For any FOL-signature (S,F,P) and any new constant x of sort* ST *we define the following translation*

$$[\_]^x_{(S,F,P)} : \mathrm{Sen}^{FOL}(S,F,P) \to \mathrm{Sen}^{FOL}([S],[F]+x,[P])$$

defined by

- $[t = t']^x = ([t]^x = [t']^x)$ *where* $[\sigma(t_1,\ldots,t_n)]^x = \sigma(x,[t_1]^x,\ldots,[t_n]^x)$;
- $[\pi(t)]^x = \pi(x,[t]^x)$;
- $[\rho_1 \star \rho_2]^x = [\rho_1]^x \star [\rho_2]^x$, *for* $\star \in \{\vee,\wedge,\Rightarrow\}$;
- $[\neg\rho]^x = \neg[\rho]^x$;
- $[(\forall Y)\rho]^x = (\forall Y)([\rho]^x)_Y$ *where* $([\rho]^x)_Y$ *is the result of replacing in* $[\rho]^x$ *all occurrences of* $y(z)$ *by* y *for each* y *in* Y.

Definition 1.44. *[60] Let (S,F,P) be any FOL-signature.*

- *For any $s \in S$ let us denote by D_s a new designated relation symbol with arity $(\mathrm{ST})s$;*
- *For any $\sigma \in F_{s_1 \dots s_n \to s}$, by D_σ we denote the Horn sentence*

$$(\forall y)(\forall x_1,\dots,x_n) \bigwedge_{1\le i\le n} D_{s_i}(y,x_i) \Rightarrow D_s(y,\sigma(y,x_1,\dots,x_n))$$

- $D_F = \{D_\sigma \mid \sigma \in F_{\underline{ar}\to s}, \underline{ar} \in S^*, s \in S\}$.

Definition 1.45. *[60] For any FOL-signature (S,F,P) and any $([S],[F],[P])$-model M' such that $M' \models D_F$, for any $w \in M'_{\mathrm{ST}}$ the (S,F,P)-model $M'|_w$ is defined as follows:*

- *for each $s \in S$, $(M'|_w)_s = \{m \in M'_s \mid (w,m) \in M'_{D_s}\}$;*
- *for each σ in F, $(M'|_w)_\sigma(m) = M'_\sigma(w,m)$;*
- *for each π in P, $m \in (M'|_w)_\pi$ iff $(w,m) \in M'_\pi$.*

Notation 1.3.2. *[60] For any (S,F,P)-sentence ρ, by $V(\rho)$ we denote the set of all sentences $(\forall x,y)D_s(x,y)$ for s any sort of a variable in a quantification that occurs in ρ. For any set E of sentences $V(E)$ denotes $\cup\{V(\rho) \mid \rho \in E\}$.*

Let $(\mathrm{Sign}^{\mathcal{H}I}, \mathrm{Sen}^{\mathcal{H}I}, \mathrm{Mod}^C, \models)$ be a hybridization of an institution I such that for all $\chi \in \mathcal{D}^{\mathcal{H}I}$:

- χ_{Nom} are finite extensions, and
- χ_{MS} are identities.

Given any comorphism $(\Phi,\alpha,\beta) : I \to FOL^{\mathrm{th}}$ such that for each $\varphi : \Sigma \to \Sigma'$ in $\mathcal{D}^I$ we have that

- the underlying *FOL* signature morphism of $\Phi(\varphi)$ is in $\mathcal{D}^{FOL}$; and
- the difference between the theories $\Phi(\Sigma')$ and $\Phi(\Sigma)$ consists of a finite set Γ_φ of sentences,

in the paper [60] we define a comorphism $(\Phi'^C,\alpha',\beta'^C) : (\mathrm{Sign}^{\mathcal{H}I}, \mathrm{Sen}^{\mathcal{H}I}, \mathrm{Mod}^C, \models) \to FOL^{\mathrm{th}}$ in two steps:

1. We define a functor $\Phi' : \mathrm{Sign}^{\mathcal{H}I} \to \mathrm{Sign}^{FOL^{\mathrm{th}}}$ and natural transformations $\alpha' : \mathrm{Sen}^{\mathcal{H}I} \Rightarrow \Phi'; \mathrm{Sen}^{FOL^{\mathrm{th}}}$ and $\beta' : \Phi'^{\mathrm{op}}; \mathrm{Mod}^{FOL^{\mathrm{th}}} \Rightarrow \mathrm{Mod}^{\mathcal{H}I}$.
2. We extend the definitions of Φ' and β' to Φ'^C and β'^C respectively and prove the Satisfaction Condition for $(\Phi'^C,\alpha',\beta'^C)$.

Definition 1.46 (Translation of the signatures)**.** *For any $\mathcal{H}I$ signature $(\Sigma,\mathrm{Nom},\Lambda)$, let $\Phi'(\Sigma,\mathrm{Nom},\Lambda) = ([S_\Sigma],[F_\Sigma]+\overline{\mathrm{Nom}},(D_s)_{s\in S_\Sigma}+[P_\Sigma]+\overline{\Lambda},\overline{\Gamma_\Sigma}\cup D_{F_\Sigma})$ where*

- *$\Phi(\Sigma) = ((S_\Sigma,F_\Sigma,P_\Sigma),\Gamma_\Sigma)$, where $(S_\Sigma,F_\Sigma,P_\Sigma)$ is a FOL-signature and Γ_Σ is a set of $(S_\Sigma,F_\Sigma,P_\Sigma)$-sentences;*
- $(\overline{\mathrm{Nom}})_{\underline{ar}\to s} = \begin{cases} \mathrm{Nom} & \text{when } \underline{ar} = \emptyset, s = \mathrm{ST}, \\ \emptyset & \text{for the other cases}; \end{cases}$

- $(\overline{\Lambda})_{\underline{\mathrm{ar}}} = \begin{cases} \Lambda_n & \textit{when } \underline{\mathrm{ar}} = (\mathrm{ST})^n, n \in \omega \\ \emptyset & \textit{for the other cases;} \end{cases}$
- $\overline{\Gamma_\Sigma} = \{\forall x\, [\gamma]^x \mid \gamma \in \Gamma_\Sigma\} \cup V(\Gamma_\Sigma)$.

Definition 1.47 (Translation of the sentences)**.** $\alpha'_{(\Sigma,\mathrm{Nom},\Lambda)}(\rho) = (\forall x)\alpha'^x_{(\Sigma,\mathrm{Nom},\Lambda)}(\rho)$*, where* $\alpha'^x_{(\Sigma,\mathrm{Nom},\Lambda)} : \mathrm{Sen}^{\mathcal{H}I}(\Sigma,\mathrm{Nom},\Lambda) \to ([S_\Sigma],[F_\Sigma]+\overline{\mathrm{Nom}}+x,(D_s)_{s\in S}+[P_\Sigma]+\overline{\Lambda})$ *with* x *being a constant of sort* ST*, is defined by*

- $\alpha'^x(i) = (i = x)$, $i \in \mathrm{Nom}$*;*
- *for each* $\rho \in \mathrm{Sen}^I(\Sigma)$, $\alpha'^x(\rho) = [\alpha_\Sigma(\rho)]^x$
- $\alpha'^x(\rho_1 \star \rho_2) = \alpha'^x(\rho_1) \star \alpha'^x(\rho_2)$, $\star \in \{\vee, \wedge, \implies\}$*;*
- $\alpha'^x(\neg\rho) = \neg\alpha'^x(\rho)$*;*
- $\alpha'^x(@_i\rho) = \alpha'^i(\rho)$*;*
- $\alpha'^x([\lambda](\rho_1,\ldots,\rho_n)) = \forall y_1,\ldots,y_n\,(\lambda(x,y_1,\ldots,y_n) \implies \bigvee_{1\le i\le n}\alpha'^{y_i}(\rho_i))$*;*
- $\alpha'^x(\langle\lambda\rangle(\rho_1,\ldots,\rho_n)) = \exists y_1,\ldots,y_n\,(\lambda(x,y_1,\ldots,y_n) \wedge \bigwedge_{1\le i\le n}\alpha'^{y_i}(\rho_i))$*;*
- $\alpha'^x_{(\Sigma,\mathrm{Nom},\Lambda)}((\forall i)\rho) = (\forall i)\alpha'^x_{(\Sigma,\mathrm{Nom}+i,\Lambda)}(\rho)$ *for* $\rho \in \mathrm{Sen}^{\mathcal{H}I}(\Sigma,\mathrm{Nom}+i,\Lambda)$*;*
- $\alpha'^x_\Delta((\forall\chi)\rho) = (\forall y)\big(\overline{\Gamma_{\chi_{\mathrm{Sig}}}} \cup \{D_y\} \implies \alpha'^x_{\Delta'}(\rho)\big)_y$ *for* $\rho \in \mathrm{Sen}^{\mathcal{H}I}(\Delta')$
  (where $\chi = (\chi_{\mathrm{Sig}}, 1_{\mathrm{Nom}}, 1_\Lambda) : \Delta = (\Sigma,\mathrm{Nom},\Lambda) \to \Delta' = (\Sigma',\mathrm{Nom},\Lambda)$ *and* $\Phi(\chi_{\mathrm{Sig}})$ *extends the signature of* $\Phi(\Sigma)$ *with the variable* y *and the theory* $\Phi(\Sigma)$ *with the finite set of sentences* $\Gamma_{\chi_{\mathrm{Sig}}}$*).*

Definition 1.48 (Translation of the models)**.** *For any* $\mathcal{H}I$ *signature* $(\Sigma,\mathrm{Nom},\Lambda)$ *and any* $\Phi'(\Sigma,\mathrm{Nom},\Lambda)$*-model* M' *we define* $\beta'_{(\Sigma,\mathrm{Nom},\Lambda)}(M') = (M,W)$ *where*

- W *is the reduct* $M'\restriction_{(\{\mathrm{ST}\},\overline{\mathrm{Nom}},\overline{\Lambda})}$*, i.e.* $|W| = M'_{\mathrm{ST}}$*,* $W_i = M'_i$ *for each* $i \in \mathrm{Nom}$*, and* $W_\lambda = M'_\lambda$ *for each* λ *in* Λ*, and*
- $M : |W| \to |\mathrm{Mod}^I(\Sigma)|$ *is defined for each* $w \in |W|$ *by* $M_w = \beta_\Sigma(M'|_w)$ *where* $M'|_w$ *denotes here the abbreviation* $(M'\restriction_{([S_\Sigma],[F_\Sigma],[P_\Sigma])})|_w$*.*

The following theorem is the key result of [60].

Theorem 1.49. *Assume a functor* C *matching* Φ' *such that*

1. *For any* $\mathcal{H}I$*-signature* $\Delta = (\Sigma,\mathrm{Nom},\Lambda)$ *and for any* Σ*-sentence* ξ *we have*
$$\Phi'^C(\Delta) \models V(\alpha_\Sigma(\xi)). \tag{3}$$
2. *Each signature morphism* $(\chi : \Delta \to \Delta') \in \mathcal{D}^{\mathcal{H}I}$ *with* $\chi_{\mathrm{Nom}} = 1_{\mathrm{Nom}}$
   - *is adequate for* β'^C*; and*
   - *satisfies*
$$C(\Delta') \models\!\models C(\Delta) \cup \{(\forall z_1,z_2)y(z_1) = y(z_2) \mid y \in Y\}. \tag{4}$$
   (where the signature of $\Phi(\chi_{\mathrm{Sig}})$ *adds the finite block of variables* Y *to the signature of* $\Phi(\Sigma)$*)*

Then, for any $\Delta = (\Sigma, \mathrm{Nom}, \Lambda) \in |\mathrm{Sign}^{\mathcal{H}I}|$, $\rho \in \mathrm{Sen}^{\mathcal{H}I}(\Delta)$, $M' \in |\mathrm{Mod}^{FOL^{\mathrm{th}}}(\Phi'^{C}(\Delta))|$ *and* $w \in M'_{\mathrm{ST}}$,

$$\beta'^{C}_{\Delta}(M') \models^{w}_{\Delta} \rho \text{ if and only if } M'^{w} \models_{\Phi'(\Delta)+x} \alpha'^{x}_{\Delta}(\rho), \tag{5}$$

where M'^{w} *denotes the expansion of* M' *to* $\Phi'(\Delta)+x$ *defined by* $M'^{w}_{x} = w$.

The most important consequence of this theorem is that the defined encoding is a comorphism indeed.

Corollary 1.50 (Satisfaction condition for $(\Phi'^{C}, \alpha', \beta'^{C})$)**.** *If in addition to the conditions of Thm. 1.49 above we also have that*

$$\beta'_{\Delta}(M') \in |\mathrm{Mod}^{C}(\Delta)| \text{ for each } \mathcal{H}I\text{-signature } \Delta \text{ and each } M' \in |\mathrm{Mod}^{FOL^{\mathrm{th}}}(\Phi'^{C}(\Delta))|$$

then $(\Phi'^{C}, \alpha', \beta'^{C})$ *is comorphism* $(\mathrm{Sign}^{\mathcal{H}I}, \mathrm{Sen}^{\mathcal{H}I}, \mathrm{Mod}^{C}, \models) \to FOL^{\mathrm{th}}$, *i.e. for any* $\Delta \in |\mathrm{Sign}^{\mathcal{H}I}|$, $\rho \in \mathrm{Sen}^{\mathcal{H}I}(\Delta)$ *and* $M' \in |\mathrm{Mod}^{FOL^{\mathrm{th}}}((\Phi'^{C})(\Delta))|$,

$$\beta'^{C}_{\Delta}(M') \models_{\Delta} \rho \text{ if and only if } M' \models_{\Phi'^{C}(\Delta)} \alpha'_{\Delta}(\rho).$$

The paper [60] also gives a general method to lift the conservativity property from the base comorphism $(\Phi, \alpha, \beta) : I \to FOL^{\mathrm{th}}$ to the comorphism $(\Phi'^{C}, \alpha', \beta') : \mathcal{H}I^{C} \to FOL^{\mathrm{th}}$. This result is of special importance since it allows to transfer proof tasks from the source to the target logic: first translate across the comorphism, then perform them in the target logic, and finally return back the results to the source logic.

The general results of [60] are applied in the same paper to a case study of a dynamically reconfigurable specification. The underlying logic of the case study is a particular hybridization of partial algebra. The case study includes also a formal verification with the corresponding proofs being performed by using SPASS [151] automatic first order logic prover through the Hets system [116].

1.4 MODEL THEORY FOR UNCONVENTIONAL AND NON-CLASSICAL LOGICS

One of the main aims of institutional model theory was to support the development of model theories for unconventional and non-classical logics, many of them being used in computer science. While few of these had been previously only partially developed, most of them did not have a model theory before. The reason is that a model theory for an unconventional logic is often a very difficult and complex task. However when approaching this problem from a top-down perspective, by using the abstract conceptual infrastructure provided by institution theory, the task of developing model theories for particular unconventional logic gets significantly simpler.

The institution theoretic model theory literature, including also [38], usually contains illustrations of the applicability of the abstract general results by instances to a range of unconventional logical systems that have a significant presence in computer science, such as partial algebra, preordered algebra, higher order logic, modal logics, etc. Here we will present several model theoretic developments that are heavily supported by institutional model theory in non-classical logics such as many-valued and modal logics. Sometimes these are also considered in a rather generic form in which their non-essential parts are abstract.

1.4.1 *Initial semantics in many valued logic*

The work [43] develops and studies a concept of quasi-variety for many-valued logic (*MVL*); a consequence being the derivation of an initial semantics result for many-valued logic. From the side of the theory of quasi-varieties this represents a study on the scope an limits of the concept of quasi-variety, from the many-valued logic side it represents a contribution to its model theory [92, 22, 69] which is only in the early stages of development.

A first series of results in [43] concerns the extension of the concept of quasi-variety of models from classical logic to *MVL*, prove that each quasi-variety of *MVL* models has a reachable initial model, and prove a reciprocal of the above mentioned result. The domain of this reciprocal is however restricted to the classes of models of *MVL* 'theories' that correspond to fuzzy sets of sentences:

Theorem 1.51. *[43] Let (C,P) be an MVL signature, L be a residuated lattice, and $\Gamma \subseteq L \times \mathrm{Sen}(C,P)$. Then $\mathrm{Mod}_{(C,P)}(\Gamma)$ is a quasi-variety if and only if for any $C' \supseteq C$, for any $A \subseteq L \times \mathsf{At}(C',P)$ (with $\mathsf{At}(C',P)$ denoting the set of the (C',P) atoms) and for any $E \subseteq C' \times C'$ the class of models $\mathrm{Mod}_{(C',P)}(\Gamma \cup A, E)$ has a reachable initial model.*

The paper [43] also introduces a concept of *MVL* 'Horn sentence', that involves also the residual connector characteristic to *MVL*:

Definition 1.52 (Horn sentence)**.** *[43] Any (C,P)-sentence of the form $(\forall X)H \Rightarrow \rho$ is called a* Horn sentence *when ρ is an $(C \cup X,P)$-atom and H is a quantifier-free $(C \cup X,P)$-sentence formed from atoms and the connectives $\wedge$, $\vee$, and $\otimes$ (the residual monoidal operation). Let $\mathsf{Horn}(C,P)$ denote the set of the Horn (C,P)-sentence.*

The following result shows that models of 'Horn theories' form a quasi-variety. Consequently we obtain that Horn theories admit initial models.

Theorem 1.53. *[43] For any $\Gamma \subseteq L \times \mathsf{Horn}(C,P)$, the class of models $\mathrm{Mod}_{(C,P)}(\Gamma)$ is a quasi-variety. Consequently $\mathrm{Mod}_{(C,P)}(\Gamma)$ has a reachable initial model.*

From the perspective of fuzzy logic programming [148, 69] the 'Horn theories' of [43] correspond to (fuzzy) logic programs and the existence of initial model result corresponds to the 'least Herbrand model' construction there. So, this may be regarded as an alternative way to obtain the (same) denotational semantics for fuzzy logic programming, a way which we regard as more structural and which bridges the gap towards the formal (algebraic) specification culture (see [33] for a general algebraic oriented approach to logic programming).

1.4.2 *Model theory for abstract many valued logics*

In [49] it is shown that the generic institution $I(L)$ (see Sect. 1.2.2) proposed as a general semantic framework for many-valued logics enjoys rather naturally a couple of properties that according to the

tradition of institution-independent model theory (e.g. [38]) are of crucial importance for the development of an in-depth model theory. Given that $I(L)$ may provide a generic model theory for various concrete many-valued logical systems, the importance of these properties transfers to the level of these concrete many-valued logical systems.

A first result shows that $I(L)$ enjoys a form of model amalgamation (see [38]) that is fundamental in the development of many model theoretic results at the level of abstract institutions.

Proposition 1.54. *[49] The institution $I(L)$ is semi-exact.*

A second result shows that $I(L)$ is equipped rather naturally with a system of diagrams (see also Sect. 1.1.1).

Proposition 1.55. *[49] $I(L)$ has diagrams such that for any $I(L)$ Σ-model (μ, m), its diagram $(\Sigma_{(\mu,m)}, E_{(\mu,m)})$ is defined by*

- $\Sigma_{(\mu,m)} = \Sigma_\mu$, *and*
- $E_{(\mu,m)} = \{(\rho, m(\rho)) \mid \rho \in \mathsf{FSen}(\Sigma_\mu)\}$.

1.4.3 *Ultraproducts for possible worlds semantics*

The work [62] extends the institution theoretic method of ultraproducts developed in [31] (see Sect. 1.1.3) to modalities and Kripke semantics defined over abstract institutions (see Sect. 1.2.1). A first preliminary result in this direction is the lifting of the existence of $\mathcal{F}$-products from the base institution to the Krike models [62, 38].

The method ultraproducts for possible worlds semantics requires the following refinement of the concept of preservation of sentences by filtered factors/products (Dfn. 1.7) to Kripke semantics.

Definition 1.56. *[62] Let $\mathcal{F}$ be a class of filters. For a signature Σ, a sentence ρ is*

- modally preserved by $\mathcal{F}$-factors *when for each $i \in I_{W_F}$, $(M_F, W_F) \models^i \rho$ implies "there exists $J \in F$ and $k \in \mu_J^{-1}(i)$ such that $(M_j, W_j) \models^{k_j} \rho$ for each $j \in J$", and*
- modally preserved by $\mathcal{F}$-products *when for each $i \in I_{W_F}$, "there exists $J \in F$ and $k \in \mu_J^{-1}(i)$ such that $(M_j, W_j) \models^{k_j} \rho$ for each $j \in J$" implies $(M_F, W_F) \models^i \rho$,*

for each filter $F \in \mathcal{F}$ over a set I, for each family $(M_j, W_j)_{j \in I}$ of Σ-Kripke models, and for each F-product $\{\mu_J : (M_J, W_J) \to (M_F, W_F) \mid J \in F\}$.

The following extends the ultraproducts fundamental Thm. 1.8 to possible worlds semantics.

Theorem 1.57 (Modal fundamental theorem)**.** *[62]*

1. *Each sentence of the base institution which is preserved by $\mathcal{F}$ products (in the base institution) is also modally preserved by $\mathcal{F}$-products (of Kripke models).*
2. *Each sentence of the base institution which is preserved by $\mathcal{F}$-factors (in the base institution) is also modally preserved by $\mathcal{F}$-factors (of Kripke models).*

3. *The sentences modally preserved by $\mathcal{F}$-products (of Kripke models) are closed under possibility $\Diamond$.*
4. *The sentences modally preserved by $\mathcal{F}$-factors (of Kripke models) are closed under possibility $\Diamond$.*

Moreover if $\mathcal{F}$ is closed under reductions,

5. *The sentences modally preserved by $\mathcal{F}$-products (of Kripke models) are closed under existential χ-quantification, when χ preserves $\mathcal{F}$-products of models in the base institution (i.e., $\mathsf{Mod}(\chi)$ preserves $\mathcal{F}$-products).*
6. *The sentences modally preserved by $\mathcal{F}$-factors (of Kripke models) are closed under existential χ-quantification, when χ lifts $\mathcal{F}$-products of Kripke models (i.e., $\mathsf{K}\text{-}\mathsf{Mod}(\chi)$ lifts $\mathcal{F}$-products).*
7. *The sentences modally preserved by $\mathcal{F}$-factors (of Kripke models) and the sentences modally preserved by $\mathcal{F}$-products (of Kripke models) are both closed under finite conjunctions.*
8. *The sentences modally preserved by $\mathcal{F}$-products (of Kripke models) are closed under infinite conjunctions.*
9. *If a sentence is modally preserved by $\mathcal{F}$-factors (of Kripke models) then its negation is modally preserved by $\mathcal{F}$-products (of Kripke models).*

And finally, if we further assume that $\mathcal{F}$ contains only ultrafilters,

10. *If a sentence is modally preserved by $\mathcal{F}$-products (of Kripke models) then its negation is modally preserved by $\mathcal{F}$-factors (of Kripke models).*
11. *The sentences modally preserved by both $\mathcal{F}$-products and factors (of Kripke models) are closed under negation.*

An immediate important consequence of this result, that has been derived in [62] is a general model compactness result for Kripke semantics over abstract institutions.

1.4.4 *Quasi-varieties and initial semantics in hybridized institutions*

The work [51] develops the quasi-variety method for establishing initial semantics in hybridized institutions (see Sect. 1.2.1). The aim of this work is to provide a generic initial semantics for specifications of dynamically reconfigurable systems, applicable to a wide range of concrete logical situations. This is achieved in three steps:

1. development of inclusion systems (of [59]) for Kripke semantics; this involves the rather sophisticated Grothendieck construction on inclusion systems of [42];
2. development of direct products of models for Kripke semantics, this being a special case of the F-products of Kripke models developed in [62];
3. development of preservation results by direct products and by sub-models determined by inclusion systems.

1.4.5 *Stratified institutions*

The work in [2] constitutes an institutional general unified study of a parameterization of the satisfaction relation (between models and sentences) by introducing the concept of 'stratified' institution. These are institutions for which the satisfaction relation is 'stratified' (or parameterized in other words) by 'states of models'. These 'states of models' may be explicit valuations of variables (like in first order logic), or implicit possible worlds (like in propositional modal logic), or combination of both (like in first order modal logic), or behavioral context (like in hidden algebra [55, 75, 82, 94]), or something else.

Definition 1.58 (Stratified institution)**.** *[2] A* stratified institution *consists of:*

- *a category* Sign *of signatures,*
- *a sentence functor* $\mathrm{Sen}: \mathrm{Sign} \to \mathbf{Set}$*,*
- *a model functor* $\mathrm{Mod}: \mathrm{Sign}^{\mathrm{op}} \to \mathbf{CAT}$*,*
- *a "stratification"* $[\![\_]\!]$ *which consists of a functor* $[\![\_]\!]_\Sigma: \mathrm{Mod}(\Sigma) \to \mathbf{Set}$ *for each signature* $\Sigma \in |\mathrm{Sign}|$ *(*states of models*), and a natural transformation* $[\![\_]\!]_\varphi: [\![\_]\!]_{\Sigma'} \Rightarrow [\![\_]\!]_\Sigma \circ \mathrm{Mod}(\varphi)$ *for each signature morphism* $\varphi: \Sigma \to \Sigma'$ *such that* $[\![M']\!]_\varphi$ *is surjective for each* $M' \in |\mathrm{Mod}(\Sigma')|$*, and*
- *a satisfaction relation between models and sentences which is parameterized by model states,* $M \models^{\eta}_{\Sigma} \rho$ *where* $\eta \in [\![M]\!]_\Sigma$ *such that the two following properties are equivalent:*
  1. $\mathrm{Mod}(\varphi)(M) \models^{[\![M]\!]_\varphi(\eta)}_{\Sigma} \rho$
  2. $M \models^{\eta}_{\Sigma'} \mathrm{Sen}(\varphi)(\rho)$

Then, we can define for every $\Sigma \in |\mathrm{Sig}|$*, the satisfaction relation* $\models_\Sigma \subseteq |\mathrm{Mod}(\Sigma)| \times \mathrm{Sen}(\Sigma)$ *as follows:*

$$M \models_\Sigma \rho \text{ if and only if } M \models^{\eta}_{\Sigma} \rho \text{ forall } \eta \in [\![M]\!]_\Sigma$$

It is possible to 'extract' canonically an institution out of a 'stratified' institution as follows:

Proposition 1.59. *[2] For each signature morphism* $\varphi: \Sigma \to \Sigma'$*, each* Σ'*-model* M' *and each* Σ*-sentence* ρ*, we have:* $M' \models_{\Sigma'} \mathrm{Sen}(\varphi)(\rho)$ *if and only if* $\mathrm{Mod}(\varphi)(M') \models_\Sigma \rho$*.*

At this level [2] also contains a development of a general Tarski style study of connectives which is an abstract unified approach to the usual Boolean connectives, to quantifiers, and to modal connectives, and we show that this determines canonically a stratified institution (and hence an institution). This way to explicitly structure the satisfaction relation opens the possibility to an institution-independent framework in which various modal and non-modal logics can be treated uniformly. This is illustrated in [2] by developing a general concept of elementary (model) homomorphism and by proving a general version of Tarski Elementary Chain Theorem [21, 143].

2

ALGEBRAIC SPECIFICATION

The field of formal specification and verification of software and hardware systems is without alternative in safety-critical or security areas where one cannot take the risk of failure. This includes several success stories such as the verification of the Pentium IV arithmetic, the verification of the Traffic Collision Avoidance System TCAS, various security protocols, etc. In many cases, only the use of logic-based techniques has been able to reveal serious bugs in software and hardware systems; in other cases, spectacular and costly failures such as the loss of the Mars Climate Orbiter could have been avoided by formal techniques.

The achievements of the author in this area are concerned with the trend called *algebraic specification* [137], which has a long history in computer science going back to the sixties and which is considered by many as being most solidly founded. While originally algebraic specification is based upon considering (many sorted) general algebra as an underlying logic for specification formalisms, since about three decades the perspective of this area has widened up to a multitude of logical systems and more recently even to heterogeneous environments based upon systems of logical systems. All these logical diversity has called for a meta-level abstract approach, given by the institution theory of Goguen and Burstall [73].

In this chapter we will present authored results in the areas of behavioural specification, structuring of specifications, heterogeneous specification, and formal verification through logic programming and structural induction. All these theoretical developments have been largely motivated by the work on the modern algebraic specification language CafeOBJ, whose main features will be also presented here.

2.1 BEHAVIOURAL SPECIFICATION

Modern algebraic specification theory and practice has extended the traditional many-sorted algebra-based specification to several new paradigms. One of the most promising is behavioural specification, which originates from the work of Horst Reichel [131, 132] and can be found in the literature under names such as hidden algebra [75, 76], observational logic [9, 94], coherent hidden algebra [55] and hidden logic [133]. Behavioural specification characterises how objects (and systems) *behave*, not how they are implemented. This new form of abstraction can be very powerful for the specification and verification of software systems since it naturally embeds other useful paradigms such as concurrency, object-orientation, constraints, nondeterminism, etc. (see [76] for details). In the tradition of algebraic specification, the behavioural abstraction is achieved by using specification with hidden sorts and a behavioural concept of satisfaction based on the idea of indistinguishability of states that are observationally the same, which also generalizes process algebra and transition systems (see [76]).

An important effort has been undertaken to develop languages and systems supporting the behavioural extension of conventional or less conventional algebraic specification techniques; these include CafeOBJ [54,

56], CIRC [135] and BOBJ [133]. In other situations, behavioural specification, although not directly realized at the level of the language definition, is employed as a mere methodological device [10].

In this section we present two important authored achievements in the area of behavioural specification, namely an extension of the original paradigm that puts it at another level with respect to applications, and a solidly founded method for object composition within behavioural specifications.

2.1.1 *Coherent hidden algebra*

This has been introduced in [55, 56] and we abbreviate it by *HA*. It is both a simplification and extension of the classical hidden algebra of [75, 76] in several directions, most notably by allowing operations with multiple hidden sorts in the arity, and differs only slightly from other modern formalizations of hidden algebra in the literature [94, 133]. *HA* also is significantly more general than coalgebra with final semantics [98] since it integrates smoothly data types and it allows behavioural operations with multiple hidden sorts. Here we present a slightly upgraded variant (the changes concerning mostly notational conventions) that is used in the most recent works.

Definition 2.1 (Signatures)**.** *A HA signature is a tuple* (H,V,F,BF) *where*

- $(H \cup V, F)$ *is an MSA signature with* $H \cap V = \emptyset$*; the sorts in* V *are called* visible sorts *and the sorts in* H *are called hidden sorts; and*
- $(H \cup V, BF)$ *is a sub-signature of* $(H \cup V, F)$ *such that* $BF_{w \to s} = \emptyset$ *when* $w \in V^*$*; the operations of* BF *are called* behavioural operations.

Definition 2.2 (Hidden algebras)**.** *Given a signature* (H,V,F,BF)*, an* (H,V,F,BF)-algebra *is just an MSA* $(H \cup V, F)$*-algebra.*

Definition 2.3 (Hidden congruence)**.** *Given a* (H,V,F,BF)*-algebra* A*, a* hidden (H,V,F,BF)*-congruence* $\sim$ *on* A *is just an* $(H \cup V, BF)$*-congruence which is identity on the visible sorts.*

Definition 2.4 (Behavioural equivalence)**.** *The largest hidden* (H,V,F,BF)*-congruence* $\sim_A$ *on a* (H,V,F,BF)*-algebra* A *is called the* behavioural equivalence on A.

A proof of the following crucial result can be found in several variants in several places in the literature; in the form represented by our particular hidden algebra formalization it can be found in [41]. This result generalizes the final semantics employed by the early hidden algebra frameworks [75] or by the coalgebraic approaches [98] to the situation of behavioural operations with multiple hidden sorts in the arity and of loose interpretation of the visible part of the signature.

Theorem 2.5. *Behavioural equivalence exists for any* (H,V,F,BF)*-algebra.*

Definition 2.6 (*HA* sentences)**.** *Given a HA signature* (H,V,F,BF)*, the* (H,V,F,BF)*-sentences are built like the MSA* $(H \cup V, F)$*-sentences from the two kinds of atoms, behavioural equations* $t \sim t'$ *and strict equations* $t = t'$ *by iteration of quantifications and Boolean connectives* ($\wedge$, $\vee$, $\neg$, $\Rightarrow$, *etc.)*

Definition 2.7 (*HA* satisfaction)**.** *The satisfaction relation between* (H,V,F,BF)*-algebras and* (H,V,F,BF)*-sentences is defined like in MSA, in the Tarski style by recursion on the structure of the sentences, with the addition that an algebra A satisfies a behavioural equation* $t \sim t'$ *if and only if* $A_t \sim_A A_{t'}$*.*

Definition 2.8 (Quasi-morphisms of signatures)**.** *A* quasi-morphism of *HA* signatures $\varphi : (H,V,F,BF) \to (S',V',F',BF')$ *is just an MSA morphism of signatures* $\varphi : (H \cup V,F) \to (H' \cup V',F')$ *such that*

- $\varphi(H) \subseteq H'$ *and* $\varphi(V) \subseteq V'$*, and*
- *the restriction of* φ *to* $(H \cup V,BF)$ *is an MSA signature morphism* $(H \cup V,BF) \to (H' \cup V',BF')$*.*

Fact 2.9. *The HA signature quasi-morphisms are closed under composition.*

Definition 2.10 (*HA* signature morphisms)**.** *A quasi-morphism* $\varphi : (H,V,F,BF) \to (H',V',F',BF')$ *is a* signature morphism *if and only if the following 'encapsulation' condition holds:*

for any $\sigma' \in BF'_{w \to s}$*, if* $w \cap \varphi(H) \neq \emptyset$ *(i.e. w contains an 'old' hidden sort) then there exists* σ *in BF such that* $\sigma' = \varphi(\sigma)$*.*

Notation 2.1.1. *Given a MSA signature* (S,F) *and a sort* $z \in S$ *we denote*

$$F_{[z]} = \{(\sigma,w,s) \mid w \in S^*, s \in S, \sigma \in F_{w \to s}, z \in w\}.$$

Note that for any MSA signature (S,F) *and sort* $z \in S$*, any morphism of MSA signatures* $\varphi : (S,F) \to (S',F')$ *induces a map* $\varphi_{[z]} : F_{[z]} \to F'_{[\varphi(z)]}$ *defined by* $\varphi_{[z]}(\sigma,w,s) = (\varphi(\sigma),\varphi(w),\varphi(s))$*.*

Fact 2.11. *Let* $\varphi : (H,V,F,BF) \to (H',V',F',BF')$ *be any HA quasi-morphism of signatures. Then*

1. *If* φ *is injective on hidden sorts then* φ *is a morphism of signatures if and only if for any* $h \in H$ *the map* $\varphi_{[h]} : BF_{[h]} \to BF'_{[\varphi(h)]}$ *is surjective.*
2. *If* φ *is injective then* φ *is a morphism of signatures if and only if for any* $h \in H$ *the map* $\varphi_{[h]} : BF_{[h]} \to BF'_{[\varphi(h)]}$ *is bijective.*

Fact 2.12. *The HA signature morphisms are closed under composition.*

The additional 'encapsulation' condition of Dfn. 2.10 has the flavour of class encapsulation from object orientation and guarantees that the behavioural equivalence on the 'old' (hidden) sorts is not changed.

The following is a very important consequence of this and has been proved in several different variants in several places in the literature (perhaps for the first time in [71] but within a significantly more restricted hidden algebra context).

Corollary 2.13 (*HA* satisfaction condition)**.** *For any HA signature morphism* $\varphi : \Sigma \to \Sigma'$*, any* Σ*-sentence* ρ *and any* Σ'*-algebra* A' *we have that*

$$A' \models \varphi(\rho) \text{ if and only if } A'|_\varphi \models \rho.$$

Hence, HA is institution.

Both [71, 75] comment that the derivation of the encapsulation condition on the signatures from the meta-principle of invariance of truth under change of notation (the Satisfaction Condition of institutions) seem to confirm the naturalness of both principles.

Coinduction

Thm. 2.5 provides the foundation for the rather notorious coinduction proof method. This is a rather efficient proof method, which however has a heuristic component, and which has been emphsised as one of the important formal verifications mehtodologies in CafeOBJ (see [54]).

Suppose that one wants to prove that two states, represented as terms s and s', are behaviourally equivalent. Then it is enough to perform the following steps.

1. Define an equivalence relation R (called a *coinduction relation*) for each hidden sort,
2. Prove that R is a hidden congruence, and
3. Prove that *s R s'*.

The heuristic component of the coinduction method is represented by the choice of the relation R. Often R happens to be the behavioural equivalence, however the coinduction method does not require this. The choice of R is thus left to the user which has to rely upon his insight into the problem. Some methods have been invented in order to assist and ease the process of finding such coinduction relations, such as the so-called 'circular coinduction' of [133].

2.1.2 *Hierarchical object composition in behavioural specification*

In [36] we formally define the novel concept of *behavioural object* within hidden algebra, which is the logic of CafeOBJ behavioural specification. Informally, a behavioural object is just a special kind of behavioural specification which formally specifies the space of the states of the objects together with actions ('methods') changing the state of the object, and with observations ('attributes') to (ordinary) data types. This is the basis for a precise definition of several types of composition operators on behavioural objects, such as parallel composition (without synchronisation), dynamic composition (in which component objects gets created and deleted dynamically), and composition with synchronisation generalising both the former operators. Informally, these composition operators are based on specifications of projections from the state space of the compound object to the state spaces of the components. The definitions in [36] give mathematical foundations for the corresponding methodological definitions of object composition in [54, 96, 57]. The composition operators support hierarchical composition processes in the sense that the result of a composition is still a behavioural object which can be therefore used for another composition.

This framework permits a clear formulation of semantical properties of the composition operators, such as associativity and commutativity, and final semantics (i.e. the existence of final composition models). We show that the basic parallel composition operator is associative and commutative modulo a meaningful equivalence relation between behavioural objects. Informally, two behavioural objects are *equivalent* when there is an isomorphism between the implementations modulo the same state space, the same actions, and the same behavioural equivalence between the states. For the general composition with synchronisation operator in [36] a compositionality result is proved for the behavioural equivalence relation, result which constitute the foundation for automation of the verification process at the level of a compound object (see [54, 96, 57]), and the existence of final semantics.

Below we review some of the concepts and results of this object composition method. The definitions and results, although essentially developed in [36], are presented here in a upgraded form that covers the structured case also, and which has been introduced in some unpublished lecture notes used for teaching a master level course at Școala Normală Superioară București between 2002-2011.

Definition 2.14 (Behavioural object)**.** *[36] A* behavioural object *B is pair consisting of a behavioural specification SP_B and a distinguished hidden sort h_B in $Sig[\mathrm{SP}_B] = (H_B, V_B, , F_B, BF_B)$ such that each behavioural operation in BF_B is* monadic*, i.e. it has only one hidden sort in its arity. We may denote $Sig[\mathrm{SP}_B]$ by $Sig[B]$. We establish the following terminological conventions:*

- *The hidden sort h_B denotes the (space of the)* states *of B.*
- *The visible sorted behavioural operations on h_B are called B-*observations*.*
- *The h_B-sorted behavioural operations on h_B are called B-*actions*.*

*For any behavioural object $B = (\mathrm{SP}_B, h_B)$, a B-*algebra *is just an algebra of the specification SP_B. The class of B-algebras is denoted by $\mathsf{Alg}(B)$.*

Definition 2.15 (Parallel composition)**.** *[36] A behavioural object B is a* parallel composition *of behavioural objects $B1$ and $B2$ when*

$$\mathrm{SP}_B = \mathrm{SP}_{B1} + \mathrm{SP}_{B2} + (Sig[B], E)$$

where $Sig[B]$ adds to $Sig[B1] \cup Sig[B2]$ (we assume that $h_{B1} \neq h_{B2}$) the following:

1. *h_B as hidden sort,*
2. *'projections' $\{\pi_1 : h_B \to h_{B1},\ \pi_2 : h_B \to h_{B2}\}$ as behavioural operations,*
3. *the set of B-actions: $\{\sigma_i : h_B w \to h_B \mid \sigma \in (BF_{Bi})_{h_{Bi} w \to h_{Bi}},\ i \in \{1,2\}\}$,*
4. *a set of B-observations, obs_B, and*
5. *a set of constants of sort h_B, const_B,*

and E is the union of the following sets of (universally quantified) equations:

1. *$\{(\forall \{x\} \cup W)\ \pi_i(\sigma_i(x,W)) = \sigma(\pi_i(x), W) \mid \sigma$ Bi-action, $i \in \{1,2\}\}$,*
2. *$\{(\forall \{x\} \cup W)\ \pi_j(\sigma_i(x,W)) = \pi_j(x) \mid \sigma$ Bi-action, $\{i,j\} = \{1,2\}\}$,*
3. *$\{(\forall \{x\} \cup W)\sigma(x,W) = c_\sigma[\pi_i(x)] \mid \sigma \in \mathsf{obs}_B\}$ where $i \in \{1,2\}$ and $c_\sigma[z]$ is any visible behavioural $(Sig[Bi] + W)$-context (with h_{Bi} being the sort of z), and*
4. *$\{\pi_i(c) = c_i \mid c \in \mathsf{const}_B,\ i \in \{1,2\}\}$ where c_i is any constant of sort h_{Bi}.*

Notation 2.1.2. *Let us denote by $B1 \| B2$ the class of behavioural objects B which are parallel compositions of behavioural objects $B1$ and $B2$.*

Parallel composition of behavioural objects enjoys the following important compositionality property for the behavioural equivalence.

Proposition 2.16. *[36] For any behavioural objects $B1$ and $B2$, for each parallel composition $B \in B1 \| B2$, we have that*

$$a \sim_A a' \text{ if and only if } A_{\pi_1}(a) \sim_{A_1} A_{\pi_1}(a') \text{ and } A_{\pi_2}(a) \sim_{A_2} A_{\pi_2}(a')$$

for each B-algebra A, elements $a, a' \in A_{h_B}$, and where A_i is the reduct of A to Bi for each $i \in \{1,2\}$.

Definition 2.17 (Equivalent B-algebras)**.** *[36] For any behavioural object B, two B-algebras A and A' are* equivalent*, denoted $A \equiv A'$, when*

- *$A_{h_B} = A'_{h_B}$ and $\sim_A = \sim_{A'}$ on the sort h_B, and*
- *$A_\sigma = A'_\sigma$ for each B-action σ.*

Note that the equality $A_\sigma = A'_\sigma$ above implies implicitly that $A_v = A'_v$ for each visible sort v in the arity of any B-action σ.

Definition 2.18 (Equivalent behavioural objects)**.** *[36] Two behavioural objects B and B' are* equivalent*, denoted $B \equiv B'$, when there exists a pair of functions $\Phi : Mod[B] \to Mod[B']$ and $\Psi : Mod[B'] \to Mod[B]$ which are inverse to each other modulo algebra equivalence, i.e. $A \equiv \Psi(\Phi(A))$ for each $A \in Mod[B]$ and $A' \equiv \Phi(\Psi(A'))$ for each $A' \in Mod[B']$.*

Note that isomorphic objects are equivalent. The following shows that our parallel composition without synchronisation is unique modulo equivalence of objects.

Proposition 2.19. *[36] Let $B1$ and $B2$ be behavioural objects and let $B, B' \in B1 \| B2$.*

1. *If there are no constants of the sorts h_B and $h_{B'}$ then B and B' have isomorphic classes of algebras.*
2. *B and B' are equivalent objects, i.e. $B \equiv B'$.*

In the following we show that given any models for each of the component objects, such that they are consistent on their common parts, they can be amalgamated 'canonically' to a model of the compound object. For this we need a concept of homomorphism for hidden algebras.

Definition 2.20 (Homomorphism of hidden algebras)**.** *A homomorphism $h : A \to B$ between two (H, V, F, BF)-algebras is just a homomorphism of $(H \cup V, F)$-algebra $A \to B$ which preserves the behavioural equivalence relation, i.e. if $a \sim_A a'$ then $h(a) \sim_{A'} h(a')$.*

The following gives the final semantics for parallel composition without synchronisation.

Theorem 2.21. *[36] Let $B \in B1 \| B2$ and let A_i be algebras of Bi for $i \in \{1,2\}$ such that $A_1 \restriction_{Sig[B1] \cap Sig[B2]} = A_2 \restriction_{Sig[B1] \cap Sig[B2]}$. Then there exists a B-algebra A expanding A_1 and A_2 such that for any other B-algebra A' expanding A_1 and A_2 there exists an unique B-algebra homomorphism $A' \to A$ expanding 1_{A_1} and 1_{A_2}. (In this case A is called the* final *B-algebra expanding A_1 and A_2.)*

The parallel composition of behavioural objects is trivially commutative because the order in the parallel composition is immaterial.

Fact 2.22. *For any behavioural objects B1 and B2, we have that* $B1\|B2 = B2\|B1$.

The associativity of parallel composition is non-trivial, and even its formulation requires some subtelty. On the one hand let us first take any $B_{12} \in B_1\|B_2$ and then any $B_{(12)3} \in B_{12}\|B_3$. On the other hand let us first take any $B_{23} \in B_2\|B_3$ and then take any $B_{1(23)} \in B_1\|B_{23}$. So what is the relationship between $B_{(12)3}$ and $B_{1(23)}$?

However the nice thing is that $B_{(12)3}$ and $B_{1(23)}$ can be proved equivalent in the sense of Dfn. 2.18.

Definition 2.23. *Under the above notations, we say that* $B_{(12)3}$ *and* $B_{1(23)}$ *are* compatible on constants *when there is a bijective correspondence* $\overline{(\ )}$ *between* $\mathsf{const}_{B_{(12)3}}$ *and* $\mathsf{const}_{B_{1(23)}}$ *such that for each* $c \in \mathsf{const}_{B_{(12)3}}$ *there are constants* c_1, c_2, c_3, c_{12} *and* c_{23} *such that*

- $\mathrm{SP}_{B_{(12)3}} \models \{\pi_{12}(c) = c_{12},\ \pi_1(c_{12}) = c_1,\ \pi_2(c_{12}) = c_2,\ \pi_3(c) = c_3\}$, *and*
- $\mathrm{SP}_{B_{1(23)}} \models \{\pi'_1(\overline{c}) = c_1,\ \pi'_{23}(\overline{c}) = c_{23},\ \pi'_2(c_{23}) = c_2,\ \pi'_3(c_{23}) = c_3\}$.

Note that when $\mathsf{const}_{B_{(12)3}} = \mathsf{const}_{B_{1(23)}} = \emptyset$, the objects $B_{(12)3}$ and $B_{1(23)}$ are trivially compatible on constants.

Theorem 2.24. *[36] If* $B_{(12)3}$ *and* $B_{1(23)}$ *are compatible on constants then* $B_{(12)3} \equiv B_{1(23)}$.

The mathematical definition below extends the definition of parallel composition (Dfn. 2.15) to the synchronized situations.

Definition 2.25 (Synchronized composition)**.** *[36] A behavioural object B is a* synchronized composition *of behavioural objects B1 and B2 when*

$$\mathrm{SP}_B = \mathrm{SP}_{B1} + \mathrm{SP}_{B2} + VIS + (Sig[B], E)$$

where VIS is a specification with only visible sorts and $Sig[B]$ *adds to* $Sig[B1] \cup Sig[B2] \cup Sig[VIS]$ *(we assume that* $h_{B1} \neq h_{B2}$*) the following:*

1. h_B *as hidden sort,*
2. *'projections'* $\{\pi_1 :\ h_B w_1 \to h_{B1},\ \pi_2 :\ h_B w_2 \to h_{B2}\}$ *for some strings* w_1 *and* w_2 *of visible sorts, as behavioural operations,*
3. *the set of B-actions that includes* $\{\sigma_i :\ h_B w \to h_B \mid \sigma \in (BF_{Bi})_{h_{Bi}w \to h_{Bi}},\ i \in \{1,2\}\}$,
4. *a set of B-observations,* obs_B, *and*
5. *a set of constants of sort* h_B, const_B,

and E is the union of the following sets of (universally quantified conditional) equations:

1. *For each B-action* σ *and* $i \in \{1,2\}$,
   $\{(\forall \{x\} \cup W \cup W_i)\ C_i^{\sigma,k}[x,W,W_i] \Rightarrow (\pi_i(\sigma(x,W),W_i) = \tau_i^{\sigma,k}[x,W,W_i]) \mid k \in \{1,\ldots,n_i\}\}$, *where*
   - *each* $\tau_i^{\sigma,k}[x,W,W_i]$ *is a* h_{Bi}*-sorted term of behaviourally coherent Bi-operations applied either to* $\pi_i(x,W_i)$ *or to a Bi-constant, and*

– $C_i^{\sigma,k}[x,W,W_i]$ *is a quantifier-free formula formed (by iterations of negations, conjunctions, and disjunctions), from equations* $t = t'$ *(over the signature extended with* x, W, W_i*), and where* t *and* t' *either*

  – *do not have operations with hidden sorts, or*

  – *they are terms of the form* $c[\pi_j(x,W_j)]$ *where* $c[z]$ *is a behaviourally coherent context for* Bj *with* $W_j \subseteq W \cup W_i$,

  such that for each $A \in Mod[\mathrm{SP}_B]$

  – $A \models (\forall \{x\} \cup W \cup W_i)\ \bigvee \{C_i^{\sigma,k}[x,W,W_i] \mid k \in \{1,\ldots,n_i\}\}$

  – $A \models (\forall \{x\} \cup W \cup W_i)\ C_i^{\sigma,k}[x,W,W_i] \wedge C_i^{\sigma,k'}[x,W,W_i] \Leftrightarrow \mathtt{false}$ *for* $k \neq k' \in \{1,\ldots,n_i\}$.

2. $\{(\forall \{x\} \cup W)\sigma(x,W) = c_\sigma[\pi_i(x,W_i)] \mid \sigma \in \mathsf{obs}_B\}$ *where* $i \in \{1,2\}$ *and* $c_\sigma[z]$ *is any visible behavioural* $(Sig[Bi] + W)$*-context (with* h_{Bi} *being the sort of* z*) and* $W_i \subseteq W$*, and*

3. $\{\pi_i(c,W_i) = c_{W_i} \mid c \in \mathsf{const}_B,\ i \in \{1,2\}\}$ *where* c_{W_i} *is any* Bi*-term of sort* h_{Bi} *formed from constants and variables* W_i.

Notation 2.1.3. *Let us denote by* $B1 \otimes B2$ *the class of behavioural objects* B *which are synchronised compositions of behavioural objects* $B1$ *and* $B2$.

Note that Dfn. 2.25 generalizes Dfn. 2.15 in two different ways. The parallel composition of Dfn. 2.15 is indeed a particular case of synchronized composition as defined in Dfn. 2.25 by taking each projection π_i without parameters, each $n_i = 1$, each $C_i^{\sigma,k} = \mathtt{true}$, and $\tau_i^{\sigma_i,1}[x,W] = \sigma[\pi_i(x),W]$ and $\tau_i^{\sigma_j,1}[x,W] = \pi_i(x)$ for $i \neq j$.

The following extends the compositionality property of behavioural equivalence from the parallel case (Prop. 2.16) to the synchronized case.

Theorem 2.26. *[36] For any behavioural objects* $B1$ *and* $B2$*, for each composition with synchronisation* $B \in B1 \otimes B2$*, we have that*

$$a \sim_A a' \text{ if and only if } (\forall W_i) A_{\pi_i}(a,W_i) \sim_{A_i} A_{\pi_i}(a',W_i) \text{ for } i \in \{1,2\}$$

for each B*-algebra* A*, elements* $a,a' \in A_{h_B}$*, and where* A_i *is the* Bi*-reduct of* A *for each* $i \in \{1,2\}$.

The following results extends the final algebra semantics of parallel composition result (Thm. 2.21) to synchronized compositions.

Theorem 2.27. *[36] Let* $B \in B1 \otimes B2$*. For any* Bi*-algebras* A_i *for* $i \in \{1,2\}$ *and* $A_V \in Mod[VIS]$ *such that*

$$A_1 \restriction_{Sig[B1] \cap Sig[B2]} = A_2 \restriction_{Sig[B1] \cap Sig[B2]} \text{ and } A_i \restriction_{Sig[Bi] \cap Sig[VIS]} = A_V \restriction_{Sig[Bi] \cap Sig[VIS]} \text{ for } i \in \{1,2\},$$

there exists a B*-algebra* A *expanding* A_1*,* A_2*, and* A_V *such that for any other similar* B*-algebra* A' *there exists an unique homomorphism of* B*-algebras* $A' \to A$ *expanding* 1_{A_1}*,* 1_{A_2}*, and* 1_{A_V}.

While the significance of Thm. 2.27 is that it guarantees the consistency of the composition and also gives an indication how implementation of the component specifications can be composed as a compound implementation, Thm. 2.26 shows that the formal verification of behavioural properties at the level of the compound object may be automatic. Moreover the rigidness of the methodology proposed allows for a straightforward realization of this methodology at the level of the specification language constructs, which may result as an important tool support for this object composition methodology.

2.2 MODULARITY/STRUCTURING

A promising approach to developing large and complex systems (which may be software, hardware, or both) is to start from a description of the system as an interconnection of some specification modules. This permits the verification of many properties to be carried out at the level of design, rather than code, and thus should improve reliability. With suitable mechanical support, it might also improve the efficiency of the development process. In addition, it promotes reuse, because some modules may be taken directly from a library, or else may be modifications of library modules. For this reason, many modern programming and specification languages support some form of modularisation, and most results about modules have appeared in the context of formal software engineering, particularly specification languages.

The earliest work on software modules is by Parnas [126, 127, 128]. Program modules differ from earlier program structuring mechanisms such as subroutines, procedures and blocks, in that they may include a number of procedure and data definitions, may be parameterised, may import other modules, and may hide certain elements. A major motivation for modules in this sense is to facilitate the modification of software, by localising the representation of data and the operations that depend upon it; this is called *information hiding*. Such modules support software reuse because they can be specified, verified, and compiled separately. Note that this notion of module is essentially *syntactic*: it concerns texts that describe systems. The earliest work on specification modules is by Goguen and Burstall, for their specification language Clear [17, 18], the semantics of which is based on institutions. This approach to modules has been applied to various logic-based languages, particularly OBJ [85] (an equational based on order sorted algebra), Eqlog [78] (which combines the functional and logic paradigms), FOOPS [80, 86] (which combines the functional and object paradigms), and FOOPlog [80] (which combines all three paradigms); it could also be applied to any pure logic-based programming languages, such as (pure) Lisp and (pure) Prolog. In [70], it is even extended to imperative programming. Modern specification languages such as CASL [5] and CafeOBJ [54] owe their structuring mechanisms more to the model oriented upgrade of the Clear approach proposed in the seminal paper [136] than to the Clear approach itself.

In this section we review some authored contributions to the foundations of structuring specifications.

2.2.1 *Module algebra*

The *module algebra* of Bergstra, Heering and Klint [6] attempts to capture the horizontal structure of modules with equations among certain basic operations on modules, including sum, renaming, and information hiding. These equations, together with constructors for signatures and sentences, give a many sorted equational presentation, about which some interesting results can be proved, including a normal form theorem. Unfortunately, this work has first order logic built into its choice of the constructors for signatures and sentences. However, Bergstra *et al.* abstract some interesting general principles from this special case. The work [53] (which was published in a special journal issue dedicated to Jan Bergstra) develops module algebra over arbitrary institutions rather than first order logic by extending the model oriented module algebra proposed in [136] (see also [137]) with new specification building operators dealing with non-protecting importation modes. The non-protecting importation modes constitute a feature of module systems of OBJ [85] and CafeOBJ [54] but they do not exist in CASL [5]. The paper [53] argues that specification with non-protecting importation modes has strictly more expressive power than that without. Below we review the institution independent semantics for structured specifications of [53] that extends that defined in [136] (which has been rather intensively used in the foundational works about structuring of specifications) with new building operators. In addition in [53] we also propose that for any specification SP we may calculate the *set of its axioms* $Ax[\mathrm{SP}] \subseteq \mathrm{Sen}(Sig[\mathrm{SP}])$.

Because the union operator below is defined total in [53] rather than partial like in [136, 137] we have to resort to the concept of inclusion system. Inclusion systems were introduced in [59] as a categorical device supporting an abstract general study of structuring of specification and programming modules that is independent of any underlying logic. There they were defined in a stronger version (corresponding to the epic inclusion systems with unions in our paper); here we use their weaker variant introduced by [27] under the name of 'weak inclusion systems'. Inclusion systems provide the underlying mathematical structure for module imports (which constitute the most fundamental structuring construct) in specification theory, and consequently have been used in a series of general module algebra studies such as [59, 84, 38]. Moreover they have also been used for developing axiomatizability [134, 32, 38] and definability [1] results within the framework of institution-independent model theory [38].

Inclusion systems capture categorically the concept of set-theoretic inclusion in a way reminiscent of how the rather notorious concept of factorization system [13] captures categorically the set-theoretic injections; however in many applications the former are more convenient than the latter. In fact, the applications to specification module algebra can be done only with inclusion systems, since factorization systems lack the uniqueness feature of inclusion systems.

Definition 2.28 (Inclusion systems)**.** $\langle I, \mathcal{E} \rangle$ *is a* inclusion system *for a category* $\mathbb{C}$ *if* I *and* $\mathcal{E}$ *are two sub-categories with* $|I| = |\mathcal{E}| = |\mathbb{C}|$ *such that*

1. I *is a partial order (with the ordering relation denoted by* $\subseteq$*), and*
2. *every arrow* f *in* $\mathbb{C}$ *can be factored uniquely as* $f = e_f; i_f$ *with* $e_f \in \mathcal{E}$ *and* $i_f \in I$*.*

The arrows of I are called abstract inclusions*, and the arrows of $\mathcal{E}$ are called* abstract surjections*. The domain of the inclusion i_f in the factorization of f is called the* image of f *and is denoted as* $\mathrm{Im}(f)$ *or* $f(A)$ *when A is a domain of f.*

Definition 2.29 (Epic inclusion systems)**.** *An inclusion system is* epic *when all abstract surjections are epis.*

The concept below is critical for the semantics of (software) module imports and is one of the important features that distinguishes inclusion systems from factorization systems in the sense that the latter can not support such concept in a proper way.

Definition 2.30 (Unions)**.** *An inclusion system $\langle I, \mathcal{E}\rangle$* has unions *when I has finite least upper bounds (denoted $\cup$).*

Let us fix two classes of signature morphisms $\mathcal{T}, \mathcal{D} \subseteq \mathrm{Sign}$, considered as parameters for the structuring process.

Each finite presentation (Σ, E) is a specification such that

$$\begin{aligned} \mathit{Sig}[(\Sigma,E)] &= \Sigma, \\ \mathit{Ax}[(\Sigma,E)] &= E, \\ \mathit{Mod}[(\Sigma,E)] &= \mathrm{Mod}(\Sigma,E). \end{aligned}$$

For any specifications SP_1 and SP_2 we can take their *union* $\mathrm{SP}_1 \cup \mathrm{SP}_2$ with

$$\begin{aligned} \mathit{Sig}[\mathrm{SP}_1 \cup \mathrm{SP}_2] &= \mathit{Sig}[\mathrm{SP}_1] \cup \mathit{Sig}[\mathrm{SP}_2], \\ \mathit{Ax}[\mathrm{SP}_1 \cup \mathrm{SP}_2] &= \mathit{Ax}[\mathrm{SP}_1] \cup \mathit{Ax}[\mathrm{SP}_2], \\ |\mathit{Mod}[\mathrm{SP}_1 \cup \mathrm{SP}_2]| &= \{M \in \mathrm{Mod}(\mathit{Sig}[\mathrm{SP}_1 \cup \mathrm{SP}_2]) \mid M\restriction_{\mathit{Sig}[\mathrm{SP}_i]} \in \mathit{Mod}[\mathrm{SP}_i] \text{ for each } i \in \{1,2\}\}. \end{aligned}$$

For any specification SP and signature morphism $(\varphi : \ \mathit{Sig}[\mathrm{SP}] \to \Sigma') \in \mathcal{T}$ we can take its *translation along* φ denoted by $\mathrm{SP} \star \varphi$ and such that

$$\begin{aligned} \mathit{Sig}[\mathrm{SP} \star \varphi] &= \Sigma', \\ \mathit{Ax}[\mathrm{SP} \star \varphi] &= \varphi(\mathit{Ax}[\mathrm{SP}]), \\ |\mathit{Mod}[\mathrm{SP} \star \varphi]| &= \{M' \in \mathrm{Mod}(\Sigma') \mid M'\restriction_{\varphi} \in \mathit{Mod}[\mathrm{SP}]\}. \end{aligned}$$

When φ is inclusion we may denote $\mathrm{SP} \star \varphi$ by $\mathrm{SP} \star \Sigma'$.

For any specification SP' and signature morphism $(\varphi : \ \Sigma \to \mathit{Sig}[\mathrm{SP}']) \in \mathcal{D}$ we can take its *derivation along* φ denoted by $\varphi \mid \mathrm{SP}'$ such that

$$\begin{aligned} \mathit{Sig}[\varphi \mid \mathrm{SP}'] &= \Sigma, \\ \mathit{Ax}[\varphi \mid \mathrm{SP}'] &= \varphi^{-1}(\mathit{Ax}[\mathrm{SP}']^{**}), \\ |\mathit{Mod}[\varphi \mid \mathrm{SP}']| &= \{M'\restriction_{\varphi} \ \mid \ M' \in \mathit{Mod}[\mathrm{SP}']\}. \end{aligned}$$

When φ is inclusion we may denote $\varphi \mid \mathrm{SP}'$ by $\Sigma \mid \mathrm{SP}'$.

Given a class $\mathcal{H}$ of model homomorphisms, we consider the *$\mathcal{H}$-extension* of a specification SP, denoted $\mathcal{H}(\mathrm{SP})$, such that

$$\begin{aligned}
Sig[\mathcal{H}(\mathrm{SP})] &= Sig[\mathrm{SP}],\\
Ax[\mathcal{H}(\mathrm{SP})] &= Ax[\mathrm{SP}], \text{ and}\\
|Mod[\mathcal{H}(\mathrm{SP})]| &= \{M' \in \mathrm{Mod}(Sig[\mathrm{SP}]) \mid M' \models Ax[\mathrm{SP}] \text{ and}\\
&\qquad \text{there exists } (h:\ M \to M') \in \mathcal{H} \text{ with } M \in Mod[\mathrm{SP}]\}
\end{aligned}$$

Given a class $\mathcal{H}$ of model homomorphisms, for any specifications SP_1 and SP_2 and signature morphism $\varphi:\ Sig[\mathrm{SP}_1] \to Sig[\mathrm{SP}_2]$, we consider the *$\mathcal{H}$-free restriction of* SP_2 *modulo* φ *and* SP_1, denoted $\mathrm{SP}_2\ !_{\mathcal{H}}\ (\varphi, \mathrm{SP}_1)$, such that

$$\begin{aligned}
Sig[\mathrm{SP}_2\ !_{\mathcal{H}}\ (\varphi, \mathrm{SP}_1)] &= Sig[\mathrm{SP}_2],\\
Ax[\mathrm{SP}_2\ !_{\mathcal{H}}\ (\varphi, \mathrm{SP}_1)] &= Ax[\mathrm{SP}_2], \text{ and}\\
|Mod[\mathrm{SP}_2\ !_{\mathcal{H}}\ (\varphi, \mathrm{SP}_1)]| &= \{M_2 \in Mod[\mathrm{SP}_2] \mid \text{there exists } M_1 \in Mod[\mathrm{SP}_1] \text{ and}\\
&\qquad \text{an universal arrow } (\eta:\ M_1 \to M_2{\restriction_\varphi}) \in \mathcal{H}\\
&\qquad \text{to the reduct functor } Mod[\mathrm{SP}_2] \to \mathrm{Mod}(Sig[\mathrm{SP}_1])\}.
\end{aligned}$$

When φ is an inclusion of signatures we may omit φ from the notations and denote $\mathrm{SP}_2\ !_{\mathcal{H}}\ (\varphi, \mathrm{SP}_1)$ simply by $\mathrm{SP}_2\ !_{\mathcal{H}}\ \mathrm{SP}_1$. When SP_1 is a presentation of the form $(\Sigma, \emptyset)$, with Σ signature, we may simply write it as Σ and denote the specification $\mathrm{SP}_2\ !_{\mathcal{H}}\ (\varphi, \mathrm{SP}_1)$ by $\mathrm{SP}_2\ !_{\mathcal{H}}\ \varphi$ or $\mathrm{SP}_2\ !_{\mathcal{H}}\ \Sigma$ when φ is inclusion. When $\mathcal{H}$ is the class of identities we omit it as the subscript of !, and the universal property of the models of $\mathrm{SP}_2\ !_{\mathcal{H}}\ (\varphi, \mathrm{SP}_1)$ is called *strongly persistently* φ*-free*.

Remark 2.31.

1. As mentioned above in some of the literature, e.g. [14, 137] etc., the union $\cup$ is usually partially defined, only for specifications over the same signature. The general union of two specifications is then obtained as the (partially defined) union of their translations to the union signature. Like in [59] our use of inclusion systems (concept introduced in [59]) allows for the direct definition of the union of *any* specifications, without any conditions.
2. Note that if $\mathcal{T}$ and $\mathcal{D}$, resp., are the class of the identities, then **TRANS** and **DERIV**, resp. are cancelled. The rather realistic idea to define **TRANS** and **DERIV** relative to sub-classes of signature morphisms seems to belong to [14]. Often in practice $\mathcal{D}$ is the class of signature inclusions while $\mathcal{T}$ is the class of all signature morphisms.
3. $\mathcal{H}$-**EXT** is a completely new operator introduced in [53] for capturing non-protecting importation modes.
4. Our operator $\mathcal{H}$-**FREE** constitutes a significant extension in [53] of the existing initial semantics operator that can be found in the literature (such as in [137]) which corresponds to the case when $\mathcal{H}$ is the class of the identities and SP_1 is empty. The extension to arbitrary $\mathcal{H}$ is motivated by the capture of initial semantics in relation with non-protecting importation modes.

Fact 2.32. *The following defines a preorder on specifications*

$$\mathrm{SP}_1 \models \mathrm{SP}_2 \text{ if and only if } Sig[\mathrm{SP}_1] = Sig[\mathrm{SP}_2] \text{ and } Mod[\mathrm{SP}_1] \subseteq Mod[\mathrm{SP}_2].$$

The Definitions 2.33 and 2.34 together with the Facts 2.35 and 2.36 below can be found in the literature, for example in [137].

Definition 2.33 (Equivalent specifications)**.** *Two specifications* SP_1 *and* SP_2 *are* equivalent*, denoted* $\mathrm{SP}_1 \models\!\!\models \mathrm{SP}_2$*, when* $\mathrm{SP}_1 \models \mathrm{SP}_2$ *and* $\mathrm{SP}_2 \models \mathrm{SP}_1$.

In general it is possible to have different specifications that are equivalent. When we are interested only in the semantics of specifications rather than in the way they are constructed, it does make sense to consider specifications modulo this equivalence relation.

Definition 2.34 (Specification morphisms)**.** *A* specification morphism $\varphi : \mathrm{SP}_1 \to \mathrm{SP}_2$ *between specifications* SP_1 *and* SP_2 *is a signature morphism* $\varphi : Sig[\mathrm{SP}_1] \to Sig[\mathrm{SP}_2]$ *such that* $\mathrm{SP}_2 \models \mathrm{SP}_1 \star \varphi$.

Fact 2.35. *A signature morphism* $\varphi : Sig[\mathrm{SP}_1] \to Sig[\mathrm{SP}_2]$ *is a specification morphism* $\mathrm{SP}_1 \to \mathrm{SP}_2$ *if and only if* $\varphi \mid \mathrm{SP}_2 \models \mathrm{SP}_1$.

Fact 2.36. *For any institution* I*, the specifications and their morphisms under the obvious composition form a category, denoted* $\mathbb{S}pec^I$.

Proposition 2.37. *For any specifications* SP, SP′, SP″,

$$\mathrm{SP} \cup \mathrm{SP}' \models\!\!\models \mathrm{SP}' \cup \mathrm{SP}. \tag{6}$$

$$\mathrm{SP} \cup \mathrm{SP} \models\!\!\models \mathrm{SP}. \tag{7}$$

$$(\mathrm{SP} \cup \mathrm{SP}') \cup \mathrm{SP}'' \models\!\!\models \mathrm{SP} \cup (\mathrm{SP}' \cup \mathrm{SP}''). \tag{8}$$

The following series of module algebra results represent new contributions of [53].

Proposition 2.38. *In any institution, for any pushout of signatures as below*

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

and for any specifications SP_1, SP_2 *such that* $\Sigma_k = Sig[\mathrm{SP}_k]$ *for* $k \in \{1,2\}$ *we have that*

$$(\varphi_k;\theta_k) \mid (\mathrm{SP}_1 \star \theta_1 \cup \mathrm{SP}_2 \star \theta_2) \models (\varphi_1 \mid \mathrm{SP}_1) \cup (\varphi_2 \mid \mathrm{SP}_2), \text{ for } k \in \{1,2\}. \tag{9}$$

If the institution has weak model amalgamation then

$$(\varphi_k;\theta_k) \mid (\mathrm{SP}_1 \star \theta_1 \cup \mathrm{SP}_2 \star \theta_2) \models\!\!\models (\varphi_1 \mid \mathrm{SP}_1) \cup (\varphi_2 \mid \mathrm{SP}_2), \text{ for } k \in \{1,2\}. \tag{10}$$

Corollary 2.39. *In any institution with unions and intersections of signatures, for any specifications* SP_1 *and* SP_2*, let* $\Sigma = Sig[\mathrm{SP}_1] \cap Sig[\mathrm{SP}_2]$*. Then*

$$\Sigma \mid (\mathrm{SP}_1 \cup \mathrm{SP}_2) \models (\Sigma \mid \mathrm{SP}_1) \cup (\Sigma \mid \mathrm{SP}_2). \tag{11}$$

Moreover if the institution has weak model amalgamation and each intersection-union square of signatures is pushout then

$$\Sigma \mid (\mathrm{SP}_1 \cup \mathrm{SP}_2) \models\!\models (\Sigma \mid \mathrm{SP}_1) \cup (\Sigma \mid \mathrm{SP}_2). \tag{12}$$

The distributivity rule (12) above has been stated as an exercise in [137] for the particular case of equational logic. Its property oriented variant has been a cornerstone in [6] (for the special case of many sorted first order logic) and in [59] (in the general institution-independent case), its proof has been significantly more difficult that the proof above of its model oriented variant and required an interpolation property for the underlying institution.

Fact 2.40. *If* $\mathcal{H}$ *contains all identities then for each flat specification* (Σ, E)

$$\mathcal{H}(\Sigma, E) \models\!\models (\Sigma, E). \tag{13}$$

Fact 2.41. *If* $\mathcal{H} = \mathcal{H}'; \mathcal{H}''$ *then for each specification* SP

$$\mathcal{H}'(\mathcal{H}''(\mathrm{SP})) \models\!\models \mathcal{H}(\mathrm{SP}). \tag{14}$$

Definition 2.42. *A class* $\mathcal{H}$ *of model homomorphisms is preserved by a signature morphism* φ *when* $h\restriction_\varphi \in \mathcal{H}$ *for each* $h \in \mathcal{H}$*.*

Proposition 2.43. *If each inclusion of signatures preserves* $\mathcal{H}$ *then for all specifications* SP_1 *and* SP_2 *we have*

$$\mathcal{H}(\mathrm{SP}_1 \cup \mathrm{SP}_2) \models \mathcal{H}(\mathrm{SP}_1) \cup \mathcal{H}(\mathrm{SP}_2) \tag{15}$$

Recall from [38] the following concept:

Definition 2.44 (Lifting of relations)**.** *Let* $\varphi : \Sigma_1 \to \Sigma_2$ *be a signature morphism and* $\mathcal{R} = \langle \mathcal{R}_1, \mathcal{R}_2 \rangle$ *with* $\mathcal{R}_1 \subseteq |\mathrm{Mod}(\Sigma_1)| \times |\mathrm{Mod}(\Sigma_1)|$ *and* $\mathcal{R}_2 \subseteq |\mathrm{Mod}(\Sigma_2)| \times |\mathrm{Mod}(\Sigma_2)|$ *be a pair of binary relations. We say that* φ lifts $\mathcal{R}$ *if and only if for each* $M_2 \in |\mathrm{Mod}(\Sigma_2)|$ *and* $N_1 \in |\mathrm{Mod}(\Sigma_1)|$*, if* $\langle M_2\restriction_\varphi, N_1 \rangle \in \mathcal{R}_1$*, there exists* $N_2 \in |\mathrm{Mod}(\Sigma_2)|$ *such that* $N_2\restriction_\varphi = N_1$ *and* $\langle M_2, N_2 \rangle \in \mathcal{R}_2$*.*

$$\begin{array}{c} M_2\restriction_\varphi \overset{\mathcal{R}_1}{\text{———}} N_1 = N_2\restriction_\varphi \\ M_2 \underset{\mathcal{R}_2}{\text{———}} (\exists) N_2 \end{array}$$

Proposition 2.45. *If* φ *lifts* $\overset{\mathcal{H}}{\longleftarrow}$ *then*

$$\mathcal{H}(\mathrm{SP}) \star \varphi \models \mathcal{H}(\mathrm{SP} \star \varphi). \tag{16}$$

If φ *lifts* $\xrightarrow{\mathcal{H}}$ *and* $\mathcal{H}$ *preserves the satisfaction of all sentences of the institution then*

$$\mathcal{H}(\varphi \mid \mathrm{SP}') \models \varphi \mid \mathcal{H}(\mathrm{SP}'). \tag{17}$$

If φ *preserves* $\mathcal{H}$ *then*

$$\mathcal{H}(\mathrm{SP} \star \varphi) \models \mathcal{H}(\mathrm{SP}) \star \varphi. \tag{18}$$

$$\varphi \mid \mathcal{H}(\mathrm{SP}') \models \mathcal{H}(\varphi \mid \mathrm{SP}'). \tag{19}$$

Corollary 2.46. *If each morphism in* $\mathcal{D}$ *(i.e. used for derivation) lifts isomorphisms and* $\mathcal{H};\mathrm{Iso} \subseteq \mathrm{Iso};\mathcal{H}$ *then the class of models* $Mod[\mathrm{SP}]$ *of each specification* SP *is closed under isomorphisms.*

Proposition 2.47. *Assume the institution is semi-exact. For any pushout of signatures as below*

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

and for any specifications SP_1, SP_2 *and* SP *such that* $\Sigma_k = Sig[\mathrm{SP}_k]$ *for* $k \in \{1,2\}$ *and* $\Sigma = Sig[\mathrm{SP}]$ *we have*

$$(\mathrm{SP}_1 \star \theta_1 \cup \mathrm{SP}_2 \star \theta_2) \,!_{\mathcal{H}}\, ((\varphi_k;\theta_k),\mathrm{SP}) \models (\mathrm{SP}_1 \,!_{\mathcal{H}}\, (\varphi_1,\mathrm{SP})) \star \theta_1 \cup (\mathrm{SP}_2 \,!_{\mathcal{H}}\, (\varphi_2,\mathrm{SP})) \star \theta_2 \quad \textit{if } (\varphi_1 \mid \mathrm{SP}_1) \between (\varphi_2 \mid \mathrm{SP}_2). \tag{20}$$

Corollary 2.48. *In any semi-exact institution in which the intersection-union squares of signatures are pushouts we have that for any specifications* SP_1 *and* SP_2 *and for* $\Sigma = Sig[\mathrm{SP}_1] \cap Sig[\mathrm{SP}_2]$

$$(\mathrm{SP}_1 \,!\, \Sigma) \cup (\mathrm{SP}_2 \,!\, \Sigma) \models (\mathrm{SP}_1 \cup \mathrm{SP}_2) \,!\, \Sigma \text{ and} \tag{21}$$

$$(\mathrm{SP}_1 \cup \mathrm{SP}_2) \,!\, \Sigma \between (\mathrm{SP}_1 \,!\, \Sigma) \cup (\mathrm{SP}_2 \,!\, \Sigma) \text{ if } (\Sigma \mid \mathrm{SP}_1) \between (\Sigma \mid \mathrm{SP}_2). \tag{22}$$

Proposition 2.49. *If the institution is semi-exact, then for any pushout of signatures*

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

and any specification SP' *such that* $Sig[\mathrm{SP}'] = \Sigma'$ *we have that*

$$(\theta_1 \mid \mathrm{SP}') \,!\, \varphi_1 \models \theta_1 \mid (\mathrm{SP}' \,!\, \theta_2). \tag{23}$$

2.2.2 *Axiomatic approach to structured specifications*

In the traditional institution theoretic approaches to structuring of specifications (e.g. [136, 59], etc.) one may consider an 'upper' institution whose signatures are either theories (in the theory oriented approach, e.g. [59]) or structured specifications (in the model oriented approach, e.g. [136]) that has the following couple of properties:

- there is a 'forgetful' functor Φ to the signatures of the base institution,
- both the 'upper' and the base institution share the same sentences modulo Φ, and
- the models of the 'upper' institution are (modulo Φ) a sub-class of the models of the base institution.

The main idea underlying the approach of [45] is to consider an abstract institution in the role of this 'upper' institution together with some properties relating it to the base institution. This is what in [45] we call the *upper level of institution independence*. Technically speaking, the whole situation can be condensed in a special form of an institution morphism (in the sense of [73]), and this is taken as the axiomatic basis for developing the theory of structured or modular specifications, without reference to theories or to specification building operators. The benefits of this approach are as follows:

1. From the point of view of the model oriented approach to structured specifications, the axiomatic approach of [45] achieves independence from the commitment to any specific set of specification building operators, in other words we achieve a general uniform theory of structured specifications that can be used for any particular set of specification building operators. This is very important when we consider the richness of possible specification building operators (the book [137] gives a hint about this) with new ones being proposed (in [53] for dealing with non-protecting importation modes). Moreover, one may want also to consider quotienting structured specifications under various module algebra rules (in the sense of [6, 59, 137]), a situation which is also captured naturally by the approach of [45]. This approach may cover also structuring contexts that are beyond conventional formal specification, such as the modular approach to the model expansion problems [144].

2. It unifies the theory and the model oriented approaches to modularization, many concepts or results that seemed to bear high similarity can be now seen precisely as being both instances of the same concept or result. A basic familiar example may be given by the lifting of co-limits from signatures to specifications that can be found in [73] for the theory oriented approach and in [137] for the model oriented approach. Moreover all the concepts or results developed here can be easily reflected down to either the theory or the model oriented approach.

3. The theory is developed in a top down manner, with the hypotheses introduced on a by-need basis with the benefit of understanding clearly the causality relationships between the various aspects of specification structuring and modularization.

The main concept of the approach introduced by [45] is the following:

Definition 2.50 (Structured institutions)**.** *[45] Given two institutions* $I = (\mathrm{Sign}, \mathrm{Sen}, \mathrm{Mod}, \models)$ *and* $I' = (\mathrm{Sign}', \mathrm{Sen}', \mathrm{Mod}', \models')$ *we say that* I' *is* structured over I through Φ *when*

- $\Phi : \mathrm{Sign}' \to \mathrm{Sign}$ *is a functor,*
- *for each* I'*-signature* Σ' *we have* $\mathrm{Sen}(\Phi(\Sigma')) = \mathrm{Sen}'(\Sigma')$ *and for each* I'*-signature morphism* φ *we have* $\mathrm{Sen}(\Phi(\varphi)) = \mathrm{Sen}'(\varphi)$*,*
- *for each* I'*-signature* Σ' *we have that* $\mathrm{Mod}'(\Sigma')$ *is a full subcategory of* $\mathrm{Mod}(\Phi(\Sigma'))$ *such that for each* I'*-signature morphism* $\varphi : \Sigma'_1 \to \Sigma'_2$ *the diagram below commutes*

$$\begin{array}{ccc} \mathrm{Mod}'(\Sigma'_1) & \xrightarrow{\subseteq} & \mathrm{Mod}(\Phi(\Sigma'_1)) \\ {\scriptstyle \mathrm{Mod}'(\varphi)}\uparrow & & \uparrow{\scriptstyle \mathrm{Mod}(\Phi(\varphi))} \\ \mathrm{Mod}'(\Sigma'_2) & \xrightarrow[\subseteq]{} & \mathrm{Mod}(\Phi(\Sigma'_2)) \end{array}$$

  and

- *for each* I'*-signature* Σ'*, each* Σ'*-model* M' *and each* Σ'*-sentence* ρ *we have that*

$$M' \models'_{\Sigma'} \rho \text{ if and only if } M' \models_{\Phi(\Sigma')} \rho.$$

Definition 2.51 (Lifting co-limits)**.** *[45] Given an institution* I' *structured over* I *through* Φ*, we say that* Φ lifts co-limits *when for each diagram* D *in* Sign' *each co-limit* μ *of* $D;\Phi$*, i.e. the image in* Sign *of* D *through* Φ*, can be lifted to a co-limit* μ' *of* D *such that* $\mu'\Phi = \mu$*.*

The following consequence of lifting co-limits is rather straightforward, and generalizes corresponding results in the literature (e.g. [137]).

Fact 2.52. *If* Φ *lifts co-limits in the sense of Dfn. 2.51 then it also preserves co-limits.*

The following generalizes the lifting of amalgamation from the base institution to that of the structured specifications found in the literature (e.g. [137]).

Definition 2.53 (Compositionality)**.** *[45] An institution* I' *structured over* I *through* Φ *is* compositional *when for each pushout in* Sign'

$$\begin{array}{ccc} \Sigma' & \xrightarrow{\varphi_1} & \Sigma'_1 \\ {\scriptstyle \varphi_2}\downarrow & & \downarrow{\scriptstyle \theta_1} \\ \Sigma'_2 & \xrightarrow[\theta_2]{} & \Omega' \end{array}$$

for any model $M' \in \mathrm{Mod}(\Phi(\Omega'))$*,* $M'\restriction_{\Phi(\theta_k)} \in \mathrm{Mod}'(\Sigma'_k)$*,* $k \in \{1,2\}$*, implies* $M' \in \mathrm{Mod}'(\Omega')$*.*

Proposition 2.54. *[45] Let* I' *be an institution structured over* I *through* Φ *such that*

1. Φ *preserves pushouts, and*
2. I' *structured over* I *through* Φ *is compositional.*

If I *has model amalgamation (resp. weak model amalgamation, semi-exactness) then* I' *has model amalgamation (resp. weak model amalgamation, semi-exactness).*

In the paper [45] we introduce a concept of normal form for abstract structured specifications that captures abstractly the normal forms from the model oriented approach to structured specifications (see [6, 20, 14]). We show that in the presence of normal forms it is possible to lift a series of important logical properties from the base institution to the upper institution of the abstract structured specifications. These properties well known for their relevance to specification, include compactness, and interpolation. Another important property studied is the closure of (the class of) models of an abstract structured specification under isomorphisms. Moreover, like in the work [14], we use normal forms for lifting a sound and complete proof system from the base institution to the institution of the abstract structured specifications (however this is done differently from [14]).

Definition 2.55 (Normal form)**.** *[45] Given an institution I' structured over I through Φ and a class $\mathcal{D}$ of I-signature morphisms, a pair (φ, E) where $(\varphi : \Phi(\Sigma') \to \Sigma) \in \mathcal{D}$ and $E \subseteq \mathrm{Sen}(\Sigma)$ is a* $\mathcal{D}$-normal form *for an I'-signature Σ' when $\mathrm{Mod}'(\Sigma') = \mathrm{Mod}(\Sigma, E)\restriction_{\varphi}$. When E is finite we say that the normal form is* finitary. *We say that I' admits (finitary) $\mathcal{D}$-normal forms when each I'-signature has at least a (finitary) $\mathcal{D}$-normal form.*

Proposition 2.56. *[45] Let $(\varphi : \Phi(\Sigma') \to \Sigma, E)$ be a $\mathcal{D}$-normal form for an I'-signature Σ'. Then for each set $\Gamma' \subseteq \mathrm{Sen}'(\Sigma')$ and $\rho \in \mathrm{Sen}'(\Sigma')$*

$$\Gamma' \models'_{\Sigma'} \rho \text{ if and only if } E \cup \varphi(\Gamma') \models_{\Sigma} \varphi(\rho).$$

Definition 2.57. *A signature morphism $\varphi : \Sigma \to \Sigma'$ in an institution* lifts isomorphisms *(of models) if and only if for any two isomorphic Σ-models $M \cong N$ and any φ-expansion M' of M there exists a φ-expansion N' of N such that $M' \cong N'$.*

Proposition 2.58. *[45] Let I' be an institution structured over I through Φ such that I' admits $\mathcal{D}$-normal forms for some class $\mathcal{D}$ of I-signature morphisms. If each morphism from $\mathcal{D}$ lifts isomorphisms then for each I'-signature Σ' we have that $\mathrm{Mod}'(\Sigma')$ is closed under (I-model) isomorphisms.*

Proposition 2.59. *[45] Let I' be an institution structured over I through Φ such that I' admits $\mathcal{D}$-normal forms for some class $\mathcal{D}$ of I-signature morphisms. If I is compact then I' is compact too.*

Theorem 2.60. *[45] Let I' be an institution structured over I through Φ and $\mathcal{L}'$ and $\mathcal{R}'$ classes of signature morphisms such that*

1. *Φ preserves pushouts,*
2. *the structuring of I' is compositional,*
3. *I' admits $\mathcal{D}$-normal forms for some class $\mathcal{D}$ of I-signature morphisms,*
4. *I has Craig-Robinson $(\mathcal{L}, \mathcal{R})$-interpolation, and*
5. *$\Phi(\mathcal{L}'); \mathcal{D} \subseteq \mathcal{L}$ and $\Phi(\mathcal{R}'); \mathcal{D} \subseteq \mathcal{R}$.*

Then I' has Craig-Robinson $(\mathcal{L}', \mathcal{R}')$-interpolation.

Definition 2.61. *[45] Let I' be an institution structured over I through Φ and let $\mathcal{D}$ be a designated class of I-signature morphisms. We let $\vdash'$ be the least entailment system for I' such that for each I'-signature Σ', for each of its $\mathcal{D}$-normal forms $(\varphi : \Phi(\Sigma') \to \Sigma, E)$, for each $E' \subseteq \mathrm{Sen}'(\Sigma')$ and each $\rho \in \mathrm{Sen}'(\Sigma')$*

$$E' \vdash'_{\Sigma'} \rho \text{ if } E \cup \varphi(E') \models_{\Sigma} \varphi(\rho).$$

The following is an immediate consequence of Prop. 2.56.

Corollary 2.62. *$(I', \vdash')$ is sound. Moreover, if I' admits $\mathcal{D}$-normal forms then $(I', \vdash')$ is complete too.*

Dfn. 2.61 together with Cor. 2.62 constitute the basis for a rather simple lifting of a sound and complete proof theory from the base institution I to the abstract structured specifications (the institution I'). This goes as follows. Assuming that I' admits $\mathcal{D}$-normal forms, if we are interested to prove that ρ is a property of an abstract structured specification Σ', i.e. that $\models'_{\Sigma'} \rho$, then we have to do the following:

1. Compute a $\mathcal{D}$-normal form $(\varphi : \Phi(\Sigma') \to \Sigma, E)$ for Σ'. For example, for the $(\mathcal{T}, \mathcal{D})$-structured specifications of Sect. 2.2.1, the literature (e.g. [38]) gives a simple algorithm for this, the result being a finitary $\mathcal{D}$-normal form.
2. Prove
   $$E \models_{\Sigma} \varphi(\rho)$$
   by using a sound and complete proof theory of the base institution.

Note that the computed normal form may be any since according to Prop. 2.56 any normal form has the same effect. The important thing here is to have at least one normal form. This procedure corresponds to (some of the) actual formal verification practices, for example implementations of the OBJ family of languages (e.g. CafeOBJ [54]) compute tacitly such normal forms as flattenings of actual loose semantics modules. When performing formal verifications, the users of these languages often invoke the `open` command which (among other things) makes available for the proof process the set E of sentences of the normal form. In such methodologies the reuse of proofs comes in forms of lemmas, which may be properties proved for component parts of the specification and which are being brought to the actual context via the 'translation' property of entailment systems.

The core verification methodology for structured specifications discussed here is simpler than that emerging from the fundamental work of [14], for example it does not require interpolation. The drawback here is that the correspondence between the modular structure of proofs and that of specifications is lost. The existence of simple proof systems via normal forms has been known in the literature and is explicitly stated in [137]. Note though that all these rely upon a common important requirement: the existence of normal forms, which is explicit in our approach and in [14] and implicit in [115].

Although one of the main points of the theory of abstractly structured specifications is the liberation from concrete structuring operators, in some situations it is useful to talk about structuring operators in an abstract context. This is especially relevant when studying algebraic rules of module compositions. In [52] we have introduced a couple of the most important generic structuring operators in the literature within the context of the abstractly structured specifications. The following analogy with monoids may be quite helpful. The

structured specifications in the traditional approach [136] would correspond to the free monoids, while the concept of abstractly structured specifications endowed with definitions for some (concrete) structuring operators corresponds to the class of *all* monoids.

Definition 2.63 (Unions)**.** *[52] Let an institution I' be structured over I through Φ such that I is inclusive (when exist the unions denoted by $\cup$). We say that I'* has unions *when for any I'-signatures Σ'_1 and Σ'_2 such that $\Phi(\Sigma'_1) \cup \Phi(\Sigma'_2)$ exists there exists a designated I'-signature denoted $\Sigma'_1 \cup \Sigma'_2$ such that*

- $\Phi(\Sigma'_1 \cup \Sigma'_2) = \Phi(\Sigma'_1) \cup \Phi(\Sigma'_2)$, *and*
- $|\mathrm{Mod}'(\Sigma'_1 \cup \Sigma'_2)| = \{M' \in |\mathrm{Mod}(\Phi(\Sigma'_1) \cup \Phi(\Sigma'_2))| \mid M'|_{\Phi(\Sigma'_k)} \in |\mathrm{Mod}'(\Sigma'_k)|, k \in \{1,2\}\}$.

Definition 2.64 (Translation and Derivation)**.** *[52] For any institution I' that is structured over I through Φ and any I-signature morphism $\varphi : \Sigma \to \Omega$, we say that*

- *I' has* φ-translations *when for any I'-signature Σ' such that $\Phi(\Sigma') = \Sigma$ there exists a designated I'-signature, denoted $\Sigma' <\!\!- \varphi$, such that*
  - $\Phi(\Sigma' <\!\!- \varphi) = \Omega$, *and*
  - $|\mathrm{Mod}'(\Sigma' <\!\!- \varphi)| = \{M' \in |\mathrm{Mod}(\Omega)| \mid M'|_\varphi \in |\mathrm{Mod}'(\Sigma')|\}$.
- *I' has* φ-derivations *when for any I'-signature Ω' such that $\Phi(\Omega') = \Omega$ there exists a designated I'-signature, denoted $\varphi \square \Omega'$, such that*
  - $\Phi(\varphi \square \Omega') = \Sigma$, *and*
  - $|\mathrm{Mod}'(\varphi \square \Omega')| = \{M'|_\varphi \mid M' \in |\mathrm{Mod}'(\Omega')|\}$.

For any class $\mathcal{D}$ of I-signature morphisms we say that I' has $\mathcal{D}$-translations/derivations *when it has φ-translations/derivations for each $\varphi \in \mathcal{D}$.*

2.2.3 *Parameterized specifications*

Parameterization is one of the most important modularization features of structuring of specifications since it allows generic modules that can be instantiated to several concrete situations. Pushout-style parameterization originate from work on Clear [17] and constitutes the basis of parameterized specification for the whole OBJ family of languages (i.e. OBJ3 [85], CafeOBJ [54], etc.) but also for ACT TWO [66] and other languages. In what follows we review recent upgrades of pushout-style parameterization concepts and foundational results developed in [53] and [45].

Definition 2.65 (Parameterized I'-signatures)**.** *[45] Let I' be an institution structured over an inclusive institution I through Φ. A* parameterized I'-signature*, denoted $\Sigma'(\iota)$, consists of an I'-signature morphism $\iota : P \to \Sigma'$ such that $\Phi(\iota)$ is inclusion $\Phi(P) \subseteq \Phi(\Sigma')$. Then P is called the* parameter *of the I'-signature and Σ' the* body *of the parameterized I'-signature.*

Definition 2.66 (Instantiation of parameters)**.** *[45] Let I' be an institution structured over an inclusive institution I through Φ such that*

1. Φ *lifts co-products,*
2. Φ *has a left adjoint* $\overline{\Phi}$ *such that the units of the adjunctions are identities, and*
3. *the inclusion system of* I*-signatures has unions and intersections.*

Given a parameterized I'*-signature* $\iota : P \to \Sigma'$ *and an* I'*-signature morphism* $v : P \to \Sigma'_1$ *such that* $\Phi(P)$ *and* $\Phi(\Sigma'_1)$ *are disjoint an* instance $\Sigma'(\iota \Leftarrow v)$ of $\Sigma'(\iota)$ through v *is defined as a pushout of* I'*-signature morphisms as follows:*

$$\begin{array}{ccc} P + (\Sigma' \Cap \Sigma'_1) & \xrightarrow{\iota + i} & \Sigma' \\ {\scriptstyle v+i_1}\downarrow & & \downarrow \\ \Sigma'_1 & \longrightarrow & \Sigma'(\iota \Leftarrow v) \end{array}$$

where

- $\Sigma' \Cap \Sigma'_1$ *denotes* $\overline{\Phi}(\Phi(\Sigma') \cap \Phi(\Sigma'_1))$,
- $P + (\Sigma' \Cap \Sigma'_1)$ *is a co-product of* P *and* $\Sigma' \Cap \Sigma'_1$ *obtained as a lifting of the disjoint union* $\Phi(P) \cup (\Phi(\Sigma') \cap \Phi(\Sigma'_1))$,
- $i : \Sigma' \Cap \Sigma'_1 \to \Sigma'$ *and* $i_1 : \Sigma' \Cap \Sigma'_1 \to \Sigma'_1$*, resp., denote the* I'*-signature morphisms* $\overline{\Phi}(\Phi(\Sigma') \cap \Phi(\Sigma'_1) \subseteq \Phi(\Sigma')); \varepsilon_{\Sigma'}$ *and* $\overline{\Phi}(\Phi(\Sigma') \cap \Phi(\Sigma'_1) \subseteq \Phi(\Sigma'_1)); \varepsilon_{\Sigma'_1}$*, resp., where* ε *denotes the co-unit of the adjunction between* I*-signatures and* I'*-signatures, and*
- $\iota + i$ *and* $v + i_1$*, resp., are the corresponding unique morphisms given by the co-product property of* $P + (\Sigma' \Cap \Sigma'_1)$.

In the following we provide another definition for parameter instantiations that under some technical conditions on the structured institution is equivalent to Dfn. 2.66 but that may be technically more convenient than Dfn. 2.66 in some situations (such as dealing with multiple parameters; see the last result of [53]).

Notation 2.2.1. Let I' be an institution structured over an inclusive institution I through Φ such that

1. Φ lifts pushouts,
2. Φ has a left adjoint $\overline{\Phi}$ such that the units of the adjunctions are identities, and
3. each intersection-union square of I-signatures is pushout.

For any I'-signatures Σ' and Σ'_1 by $\Sigma' \Cup \Sigma'_1$ we denote a lifting of the intersection-union square determined by $\Phi(\Sigma')$ and $\Phi(\Sigma'_1)$ to a pushout of I'-signature morphisms as shown by the following diagram:

$$\begin{array}{ccc} \Phi(\Sigma') \cap \Phi(\Sigma'_1) & \xrightarrow{\subseteq} & \Phi(\Sigma') \\ {\scriptstyle \subseteq}\downarrow & & \downarrow{\scriptstyle \subseteq} \\ \Phi(\Sigma'_1) & \xrightarrow[\subseteq]{} & \Phi(\Sigma') \cup \Phi(\Sigma'_1) \end{array} \qquad \begin{array}{ccc} \Sigma' \Cap \Sigma'_1 & \xrightarrow{i} & \Sigma' \\ {\scriptstyle i_1}\downarrow & & \downarrow{\scriptstyle i'} \\ \Sigma'_1 & \xrightarrow[i'_1]{} & \Sigma' \Cup \Sigma'_1 \end{array}$$

Note that $\Sigma' \Cup \Sigma'_1$ in general is not unique, but rather denotes a class of isomorphic I'-signatures. However we are going to be lax about this and when there is not danger of error $\Sigma' \Cup \Sigma'_1$ will mean whatever member of this class of I'-signatures.

Proposition 2.67. *Let I' be an institution structured over an inclusive institution I through Φ. In addition to the hypotheses of Dfn. 2.66 let us also assume that*

1. *Φ lifts pushouts and*
2. *each intersection-union square of I-signatures is pushout.*

Given a parameterized I'-signature $\iota : P \to \Sigma'$ and an I'-signature morphism $v : P \to \Sigma'_1$ such that $\Phi(P)$ and $\Phi(\Sigma'_1)$ are disjoint, then $\Sigma'(\iota \Leftarrow v)$ may be defined as a pushout of I'-signature morphisms as follows:

$$\begin{array}{ccc} P + \Sigma'_1 & \xrightarrow{(\iota;i')+i'_1} & \Sigma' \biguplus \Sigma'_1 \\ {\scriptstyle v+1_{\Sigma'_1}}\downarrow & & \downarrow{\scriptstyle v'} \\ \Sigma'_1 & \xrightarrow[\iota']{} & \Sigma'(\iota \Leftarrow v) \end{array}$$

where

- *$P + \Sigma'_1$ is a co-product that lifts the disjoint union $\Phi(P) \cup \Phi(\Sigma'_1)$, and*
- *$(\iota, i') + i'_1$ and $v + 1_{\Sigma'_1}$, resp., are the unique I'-signature morphism 'extending' $(\iota; i')$, i'_1 and v, $1_{\Sigma'_1}$, resp., according to the universal property of the co-product $P + \Sigma'_1$.*

Moreover, if in addition

3. *the inclusion system for the I-signatures is epic, and*
4. *each idempotent-by-extension I-signature morphism admits free extensions along any I-signature inclusion*

then we may choose the instance $\Sigma'(\iota \Leftarrow v)$ such that

$$\Phi(\Sigma'_1) \subseteq \Phi(\Sigma'(\iota \Leftarrow v)).$$

The study of multiple parameters and methods for their instantiations has been an important common theme in both [53] and [45].

Definition 2.68 (Multiple parameters)**.** *[53, 45] A* multiple parameterized specification *is a specification with several parameters such that for any parameters P_1 and P_2 we have that $Sig[P_1]$ and $Sig[P_2]$ are disjoint.*

Proposition 2.69. *[53, 45] For any multiple parameterized specification $\mathrm{SP}(P_1, \ldots, P_n)$ we have that $\mathrm{SP}(P_1 \cup \cdots \cup P_n)$ is a (single) parameterized specification.*

One way to instantiate multiply parameterized specifications is the simultaneous of the parallel one.

Corollary 2.70 (Simultaneous instantiation of parameters)**.** *[53, 45] Let us consider a multiple parameterized specification $\mathrm{SP}(P_1, P_2)$ with two parameters P_1 and P_2. Then for any specification morphisms $v_1 : P_1 \to \mathrm{SP}_1$ and $v_2 : P_2 \to \mathrm{SP}_2$ such that for all $i, j \in \{1, 2\}$ $Sig[P_i]$ and $Sig[\mathrm{SP}_j]$ are disjoint, we have that $P_1 \cup P_2$ and $\mathrm{SP}_1 \cup \mathrm{SP}_2$ are disjoint.*

Consequently, since the condition of Dfn. 2.66 is fulfilled, the instance of $\mathrm{SP}(P_1, P_2)$ by ν_1 and ν_2 *is defined as* $\mathrm{SP}(P_1 \cup P_2 \Leftarrow \nu_1 + \nu_2)$ *where* $\nu_1 + \nu_2$ *is the unique specification morphism that makes the diagram below commute*

$$\begin{array}{ccccc} P_1 & \xrightarrow{\subseteq} & P_1 \cup P_2 & \xleftarrow{\supseteq} & P_2 \\ {\scriptstyle \nu_1}\downarrow & & \downarrow{\scriptstyle \nu_1+\nu_2} & & \downarrow{\scriptstyle \nu_2} \\ \mathrm{SP}_1 & \xrightarrow[\subseteq]{} & \mathrm{SP}_1 \cup \mathrm{SP}_2 & \xleftarrow[\supseteq]{} & \mathrm{SP}_2 \end{array}$$

According to [53, 45] another way to instantiate multiply parameterized specifications is given by a sequential procedure as below. Given the data of Cor. 2.70 we instantiate the parameters one by one by treating them as single separate parameters (Dfn. 2.66). Because in this case it is technically more convenient, let us use the variant of parameter instantiation given by Prop. 2.67. The process of sequential instantiation of parameters can be visualised in the diagram below:

$$\begin{array}{ccccc} & & P_1 \cup \mathrm{SP}_1 & \xrightarrow{\nu_1 + 1_{Sig[\mathrm{SP}_1]}} & \mathrm{SP}_1 \\ & & \subseteq\downarrow{\scriptstyle i_1} & & \downarrow{\scriptstyle i_1'}\subseteq \\ P_2 & \xrightarrow{\subseteq} & \mathrm{SP} \cup \mathrm{SP}_1 & \xrightarrow{\nu_1'} & \mathrm{SP}(P_1 \Leftarrow \nu_1) \\ \subseteq\downarrow{\scriptstyle i_2} & & & & \downarrow{\scriptstyle i_2'}\subseteq \\ P_2 \cup \mathrm{SP}_2 & & \xrightarrow[i_3]{\subseteq} & & \mathrm{SP}(P_1 \Leftarrow \nu_1) \cup \mathrm{SP}_2 \\ \downarrow{\scriptstyle \nu_2 + 1_{Sig[\mathrm{SP}_2]}} & & & & \downarrow{\scriptstyle \nu_2'} \\ \mathrm{SP}_2 & & \xrightarrow[\subseteq]{i_3'} & & \mathrm{SP}(P_1 \Leftarrow \nu_1)(P_2 \Leftarrow \nu_2) \end{array} \tag{24}$$

The correctness of the second instantiation step relies upon the fact that P_2 is indeed a parameter for the result $\mathrm{SP}(P_1 \Leftarrow \nu_1)$ of the first instantiation step. This follows immediately from the result below.

Proposition 2.71. *[53, 45] In addition to the technical hypotheses underlying the sequential instantiation defined above let us also assume that*

- *there exists a signature* 0 *initial both in* Sign *and in the subcategory* I *of the abstract inclusions,*
- *the inclusion system is epic and distributive, and*
- *each idempotent-by-extension signature morphism admits free extensions.*

Then for the diagram of sequential instantiation (24) the signature morphism $(Sig[P_2] \subseteq Sig[\mathrm{SP} \cup \mathrm{SP}_1]); \nu_1'$ *is an inclusion.*

In general the results of the simultaneous and of the sequential institution of parameters need not produce the same result (up to isomorphism). The following result, originally from [53], gives sufficient conditions widely applicable in the concrete situations such that these two instantiation methods yield essentially the same result.

Theorem 2.72. *[53, 45] Let* $\mathrm{SP}(P_1, P_2)$ *be a multiple parameterized specification and* $\nu_i : P_i \to \mathrm{SP}_i$, $i \in \{1,2\}$, *two specification morphisms such that* $Sig[P_i]$ *and* $Sig[\mathrm{SP}_j]$ *are disjoint, for* $i, j \in \{1,2\}$. *If in*

addition to the hypotheses of Prop. 2.71 the base institution strongly admits free extensions for idempotent-by-extension signature morphisms, then there exists an instantiation $\mathrm{SP}(P_1 \Leftarrow v_1)$ *(of the first parameter) such that any further instantiation* $\mathrm{SP}(P_1 \Leftarrow v_1)(P_2 \Leftarrow v_2)$ *(of the second parameter) is isomorphic with the results* $\mathrm{SP}(P_1 + P_2 \Leftarrow v_1 + v_2)$ *of the simultaneous instantiation, making the diagram below commutative.*

$$\begin{array}{ccc} \mathrm{SP}(P_1 \cup P_2 \Leftarrow v_1 + v_2) & \xrightarrow{\cong} & \mathrm{SP}(P_1 \Leftarrow v_1)(P_2 \Leftarrow v_2) \\ & \nwarrow^{\subseteq} \quad \nearrow_{\subseteq} & \\ & \mathrm{SP}_1 \cup \mathrm{SP}_2 & \end{array}$$

2.2.4 *Structuring behavioural specifications*

Bbehavioural specification poses a particular challenge given by some of the specificities of its underlying logic that raise serious obstacles when attempting to apply established general theories on modularization. For example, one major source of problems is given by the fact that the union or aggregation of behavioural signatures is inherently partial rather than total and moreover, by noticing that this partiality is induced by two different factors. On the one hand, one may not aggregate signatures that share a sort name that is declared visible in one of the signatures and hidden in the other. On the other hand, any aggregation of signatures has to fulfil the encapsulation condition characteristic to behavioural signature morphisms, which cannot be guaranteed in all possible situations; this property is essential for the basic *satisfaction condition* of the underlying logic to hold, which in turn is absolutely necessary with respect to modularization. The requirement of both of these hypotheses (i.e. the preservation and the reflection of sorts' visibility, together with the encapsulation condition) has been extensively discussed in the literature (e.g. [76, 94]), not only from a technical perspective, but also in terms of their practical relevance. Hence, under the current general assumptions on behavioural specification, the union of signatures arises naturally as a partial operation.

In [52] we develop foundations for structuring of behavioural specifications in the light of the most recent developments in structuring specifications in general [53, 45]. This means reliance upon well established theoretical devices used in modularization studies such as institutions, pushouts, and inclusion systems. It also means that only some of the concepts and results already available at the general level may be applied directly, while much has to be reconsidered for the specific situation of behavioural specifications, thus leading to a series of new theoretical investigations.

The following series of results concern the existence of pushouts and pullbacks of *HA* signature morphisms.

Proposition 2.73. *[52] The forgetful functor from the category of quasi-morphisms of HA signatures to* Sign^{MSA} *lifts pushouts.*

Proposition 2.74. *[52] The forgetful functor from the category of HA signature morphisms to the category of the quasi-morphisms of HA signatures lifts pushouts.*

Proposition 2.75. *[52] The forgetful functor from the category of the quasi-morphisms of HA signatures to* Sign^{MSA} *lifts pullbacks.*

Proposition 2.76. *[52] Every pullback of HA-signature morphisms can be lifted from the category of quasi-morphisms of HA-signatures.*

Even though the question of establishing precisely the class of cospans (θ_1, θ_2) that admit pullbacks in Sign^{HA} remains open, it should be noted that the answer to this question is irrelevant for the purposes of our work, as the pullback property provided by Prop. 2.75 is sufficient for the subsequent developments on the structuring of behavioural specifications.

Amalgamation follows from Prop. 2.73 and 2.76and from the amalgamation in many sorted algebra.

Corollary 2.77. *[52] Each pushout square of HA signature morphisms is an amalgamation square.*

Inclusion systems of [59] have become a standard foundational concept in the theory structured specifications for modelling specification imports.

Proposition 2.78. *[52] The category of the quasi-morphisms of HA signatures admits an epic inclusion system that inherits the strong inclusion system of MSA signatures as follows.*

- *The abstract inclusions* $(H,V,F,BF) \subseteq (H',V',F',BF')$ *are such that*
  - $H \subseteq H'$, $V \subseteq V'$, *and*
  - *for each* $w \in (H \cup V)^*$ *and* $s \in H \cup V$, $F_{w\to s} \subseteq F'_{w\to s}$.
- *The abstract surjections* $\varphi : (H,V,F,BF) \to (H',V',F',BF')$ *are such that*
  - $\varphi(H) = H'$ *and* $\varphi(V) = V'$, *and*
  - *for each* $w' \in (H' \cup V')^*$ *and* $s' \in H' \cup V'$,

$$F'_{w'\to s'} = \bigcup \{\varphi(F_{w\to s}) \mid \varphi(w) = w', \varphi(s) = s'\} \text{ and}$$

$$BF'_{w'\to s'} = \bigcup \{\varphi(BF_{w\to s}) \mid \varphi(w) = w', \varphi(s) = s'\}.$$

Proposition 2.79. *[52] Let* φ *be any morphism of HA signatures. Let us factor* $\varphi = e_\varphi; i_\varphi$ *through the inclusion system for the quasi-morphisms defined in Prop. 2.78. Then both the abstract surjection* e_φ *and the abstract inclusion* i_φ *are morphisms of HA signatures.*

Corollary 2.80. *The category of HA signature morphisms admits an inclusion system that inherits the strong inclusion system for the MSA signatures.*

Notation 2.2.2. *The inclusions, unions, and intersections, resp., are denoted by* $\subseteq$, $\cup$, $\cap$, *resp., in the inclusion system for the quasi-morphisms of HA signatures and by* $\sqsubseteq$, $\sqcup$, $\sqcap$, *resp., in the inclusion system for the HA signature morphisms.*

The following series of results establishes the existence of intersections of *HA* signatures. These are crucial for parameterized specifications and their instantiation with soem form of sharing.

Notation 2.2.3. *For any MSA signature* (S,F) *and set* $S_0 \subseteq S$, *we denote by* $(S_0;F)$ *the largest sub-signature of* (S,F) *having* S_0 *as the set of sorts, i.e.* (S_0,F_0) *with* $(F_0)_{w\to s} = F_{w\to s}$ *for* $w \in S_0^*$ *and* $s \in S_0$.

For any HA signature (H,V,F,BF) *and sets* $H_0 \subseteq H$ *and* $V_0 \subseteq S$, *we denote by* $(H_0,V_0;F,BF)$ *the signature* (H_0,V_0,F_0,BF_0) *where* $(H_0 \cup V_0, F_0) = (H_0 \cup V_0; F)$ *and* $(H_0 \cup V_0, BF_0) = (H_0 \cup V_0; BF)$.

Proposition 2.81. *Let $(\Sigma_i)_{i\in I}$ be a non-empty set of HA signatures. Then:*

1. *There exists $\bigcap_{i\in I}\Sigma_i$.*
2. *There exists $\bigsqcap_{i\in I}\Sigma_i$. Moreover, $\bigsqcap_{i\in I}\Sigma_i = \bigcap_{n\in\omega}\Sigma^{(n)}$, where*
   - *$\Sigma^{(0)} = \bigcap_{i\in I}\Sigma_i$, and*
   - *for each $n>0$, if $\Sigma^{(n)} = \left(H^{(n)}, V, F^{(n)}, BF^{(n)}\right)$ then*

$$\Sigma^{(n+1)} = \left(H^{(n+1)}, V, F^{(n+1)}, BF^{(n+1)}\right) = \left(H^{(n+1)}, V; F^{(n)}, BF^{(n)}\right)$$

   where $H^{(n+1)} = H^{(n)} \setminus \left\{h \in H^{(n)} \mid BF^{(n)}_{[h]} \neq (BF_i)_{[h]} \text{ for some } i \in I\right\}$.
3. *$\bigsqcap_{i\in I}\Sigma_i \sqsubseteq \bigcap_{i\in I}\Sigma_i$.*

Proposition 2.82. *For any two HA signatures and $\Sigma_1 = (H_1, V_1, F_1, BF_1)$ and $\Sigma_2 = (H_2, V_2, F_2, BF_2)$ the following are equivalent:*

1. *$H_1 \cap V_2 = H_2 \cap V_1 = \emptyset$;*
2. *there exists $\Sigma_1 \cup \Sigma_2$ which inherits the union of the underlying MSA signatures;*
3. *there exists $\Sigma_1 \cup \Sigma_2$;*
4. *there exists Σ' such that $\Sigma_1, \Sigma_2 \subseteq \Sigma'$;*
5. *$\Sigma_1 \cap \Sigma_2$ (which exists according to Prop. 2.81) inherits the intersection of the underlying MSA signatures.*

Moreover, in these situations the corresponding intersection-union square of HA signatures is a pushout square of quasi-signature morphisms.

The algebraic properties of the union $\sqcup$ and intersection $\sqcap$ of *HA*-signatures play an important role in establishing algebraic properties of behavioural specification modules. The union $\sqcup$ on the class of *HA*-signatures is a partial rather than total operation, which gives rise to a partial algebra of signatures in the sense of [15]. In what follows we make use of two types of equalities specific to partial algebras: $t \overset{e}{=} t'$ for the *existence equality*, i.e. both t and t' are defined and their values are equal, and $t = t'$ for the *strong equality*, i.e. either both t and t' are undefined or $t \overset{e}{=} t'$.

Fact 2.83. *[52] For any HA-signatures Σ, Σ_1, Σ_2,*

$$\Sigma \sqcup \Sigma \overset{e}{=} \Sigma, \text{ and} \tag{25}$$

$$\Sigma_1 \sqcup \Sigma_2 = \Sigma_2 \sqcup \Sigma_1. \tag{26}$$

Proposition 2.84. *[52] For any HA-signatures Σ_1, Σ_2, Σ_3,*

$$\Sigma_1 \sqcup (\Sigma_2 \sqcup \Sigma_3) = (\Sigma_1 \sqcup \Sigma_2) \sqcup \Sigma_3. \tag{27}$$

Proposition 2.85. *[52] For any HA-signatures Σ, Σ_1, Σ_2,*

$$\Sigma \sqcup \Sigma_1 \sqcup \Sigma_2 \overset{e}{=} \Sigma \sqcup \Sigma_1 \sqcup \Sigma_2 \quad \textit{implies} \quad \Sigma \sqcap (\Sigma_1 \sqcup \Sigma_2) \overset{e}{=} (\Sigma \sqcap \Sigma_1) \sqcup (\Sigma \sqcap \Sigma_2).$$

In general, the distributivity rule of Prop. 2.85 does not hold unconditionally, not even in a weaker form such as

$$\Sigma \sqcap (\Sigma_1 \sqcup \Sigma_2) = (\Sigma \sqcap \Sigma_1) \sqcup (\Sigma \sqcap \Sigma_2).$$

The following is a simple counterexample. Let Σ consist of a visible sort v, Σ_1 of a visible sort s and Σ_2 of a hidden sort s. Then $\Sigma_1 \sqcup \Sigma_2$ does not exist, hence the left-hand side of the rule does not exist. On the other hand the right-hand side of the equation does exist and it is the empty signature.

The next series of results from [52] develops a module algebra for behavioural structured specifications, which is a partial rather than total algebra. The behavioural specifications are considered over an abstract institution that satisfies the compositionality properties of *HA* rather than over *HA* itself. The reason for this is to accomodate other behavioural specification formalism that are based upon other behavioural logics than *HA*, or as stated in [52] to accomodate even specification formalisms that are not necessarily behavioural but display the same kind of partiality of the union as *HA*. Therefore let us consider a (Sig, I)-structured institution I' that has unions in the sense of Dfn. 2.63. Let us call the signatures of I' *specifications*, and denote them by SP, SP$'$, etc.

Definition 2.86 (Module expression)**.** *[52] The set of* module expressions *is the least set such that*

- SP *is a module expression for each 'variable'* SP *denoting a specification,*
- $E_1 \sqcup E_2$ *is a module expression when* E_1 *and* E_2 *are module expressions,*
- E –> φ *and* $\varphi\Box E$ *are module expressions when* E *is a module expression and* φ *is an* I*-signature morphism.*

Definition 2.87. *[52] For any module expressions* E *and* E'*,*

- $E \equiv E'$ *when none of* E *and* E' *are defined or they are both defined and we have* $Sig(E) = Sig(E')$ *and* $\mathrm{Mod}'(E) = \mathrm{Mod}'(E')$*; and*
- $E \stackrel{e}{\equiv} E'$ *when* E *and* E' *are defined and* $E \equiv E'$*.*

Corollary 2.88. *[52] If the unions of* I *satisfy the idempotence (25), commutativity (26) and associativity (27) rules then for any specifications* SP, SP$'$, SP$''$*,*

$$\mathrm{SP} \sqcup \mathrm{SP} \stackrel{e}{\equiv} \mathrm{SP}. \tag{28}$$

$$\mathrm{SP} \sqcup \mathrm{SP}' \equiv \mathrm{SP}' \sqcup \mathrm{SP}. \tag{29}$$

$$(\mathrm{SP} \sqcup \mathrm{SP}') \sqcup \mathrm{SP}'' \equiv \mathrm{SP} \sqcup (\mathrm{SP}' \sqcup \mathrm{SP}''). \tag{30}$$

Proposition 2.89. *If* I' *has* $\mathcal{D}$*-translations for a class* $\mathcal{D}$ *of* I*-signature morphisms that is closed under composition to the left with inclusions, then for any specifications* SP_1, SP_2 *and any* $\varphi \in \mathcal{D}$*,*

$$\mathrm{SP}_1 \sqcup \mathrm{SP}_2 \stackrel{e}{\equiv} \mathrm{SP}_1 \sqcup \mathrm{SP}_2 \quad \textit{implies} \quad (\mathrm{SP}_1 \sqcup \mathrm{SP}_2) \text{–>} \varphi \equiv (\mathrm{SP}_1 \text{–>} \varphi) \sqcup (\mathrm{SP}_2 \text{–>} \varphi). \tag{31}$$

Proposition 2.90. *[52] If I' has $\mathcal{D}$-derivations for a class $\mathcal{D}$ of I-signature morphisms that is closed under composition then for any pushout of I-signatures that consists of morphisms from $\mathcal{D}$ as below*

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

and for any specifications SP_1, SP_2 *such that* $\Sigma_k = Sig(\mathrm{SP}_k)$ *for* $k \in \{1,2\}$, *it holds that*

$$(\varphi_k;\theta_k)\Box(\mathrm{SP}_1 \mathrel{->} \theta_1 \sqcup \mathrm{SP}_2 \mathrel{->} \theta_2) \stackrel{e}{\equiv} (\varphi_1\Box\mathrm{SP}_1)\sqcup(\varphi_2\Box\mathrm{SP}_2), \quad \textit{for } k \in \{1,2\}. \tag{32}$$

Corollary 2.91. *[52] If I' has translations and derivations for inclusions and in I each intersection-union square describes a pushout then for any specifications* SP_1 *and* SP_2,

$$\mathrm{SP}_1 \sqcup \mathrm{SP}_2 \stackrel{e}{\equiv} \mathrm{SP}_1 \sqcup \mathrm{SP}_2 \quad \textit{implies} \quad \Sigma\Box(\mathrm{SP}_1 \sqcup \mathrm{SP}_2) \stackrel{e}{\equiv} (\Sigma\Box\mathrm{SP}_1)\sqcup(\Sigma\Box\mathrm{SP}_2), \tag{33}$$

where $\Sigma = Sig(\mathrm{SP}_1) \sqcap Sig(\mathrm{SP}_2)$.

Cor. 2.88, Prop. 2.89, Prop. 2.90, and Cor. 2.91 can be easily instantiated to the case when $I = HA$ based on the results previously discussed. They can also be very easily applied to other frameworks with partial unions of signatures, such as those considered in [137]. Moreover, they generalise corresponding module-algebra properties that have been previously proved in the literature, in which the unions are assumed to be total.

For a specific set of specification structuring operators for equational logic, a corresponding variant of the distributivity rule (33) has been stated as an exercise in [137] and has been proved in an abstract institution-independent setting in [53]. Its property-oriented variant has been a cornerstone in [6] (for the special case of many sorted first order logic) and in [59] this gets a general institution-independent treatment and proof. The property-oriented variants of [6, 59] required not only significantly more difficult proofs but also significantly harder conditions, namely an interpolation property for the underlying institution. Since derivation gets here ($\Box$ in Dfn. 2.64) a model-oriented definition, the result of Cor. 2.91 shares with the corresponding result from [137] freedom from interpolation (which is quite important since interpolation in general is difficult to establish, and in the particular case of *HA* it has not been studied yet, at least up to our knowledge). However, a big difference between these related results is that in our framework the union of specifications is a partial operation, hence the conditional form of the rule (33) and a more sophisticated proof, which in the case of *HA* relies upon the series of results about the existence of unions of *HA*-signatures.

2.3 HETEROGENEOUS SPECIFICATION

The specification of large software systems requires heterogeneous multi-logic specifications, since complex problems have different aspects that are best specified in different logics. In addition to this, heterogeneous specifications have the benefit that different approaches being developed at different sites can be related,

which means a formal interoperability among languages and tools. In many cases, specialized languages and tools often have their strengths in particular aspects. Using heterogeneous specification, these strengths can be combined with comparably small effort. These ideas have been expressed in many different occasions by various scientists over the past ten or fifteen years, e.g. [119, 138], etc. Currently there are several specification environments directly realising the idea of heterogeneous specification, most prominent ones being Hets [119, 116] and CafeOBJ [54]. These environments are solidly and essentially based upon the concept of Grothendieck institution (introduced first by [30] within the context of the semantics of CafeOBJ and then by [117] with a different twist). Grothendieck institutions represent an institution theoretic replica of a category theoretic construction originally from algebraic geometry [87], which in essence 'flatten' a diagram of institutions to a single institution while retaining all data provided by the respective diagram of institutions. The concept Grothendieck institution has spread beyond that of formal specification, to other computing science areas, for example providing foundations for heterogeneous ontologies [72, 101].

In this section we review authored contributions to the foundations of heterogeneous specifications in the theory of Grothendieck institutions and of Grothendieck inclusion systems.

2.3.1 *Grothendieck institutions*

The theory of Grothendieck institutions introduced in [30] has been preceded by the work [28] on institution theoretic foundations of heterogeneous specifications, the former representing a definitive upgrade of the latter. Grothendieck institutions generalize the flattening Grothendieck construction from (indexed) categories to (indexed) institutions. Regarded from a fibration theoretic angle, Grothendieck institutions are just institutions for which their category of signatures is fibred. For example, the actual institutions with many-sorted signatures appear naturally as fibred institutions determined by the fibrations given by the functor mapping each signature to its set of sort symbols. In this sense, fibred institutions can be regarded as the reflection of many-sortedness at the level of abstract institutions. However for modeling heterogeneous multi-logic environments the flattening Grothendieck construction on a system of institutions related by institution morphisms (here called *indexed institution*) seems to be more adequate than the fibred institutions approach. A Grothendieck institution puts together a system of institutions into a single institution such that the individual identities of the component institutions and the relationships between them are fully retained.

Definition 2.92. *[30] Given a category I, a* fibred institution over the base I *is a tuple* $(\Pi : \mathrm{Sign} \to I, \mathrm{Mod}, \mathrm{Sen}, \models)$ *such that*

- $\Pi : \mathrm{Sign} \to I$ *is a fibred category, and*
- $(\mathrm{Sign}, \mathrm{Mod}, \mathrm{Sen}, \models)$ *is an institution.*

The fibred institution is split *when the fibration* Π *is split. A* cartesian institution morphism *is an institution morphism between fibred institutions for which the signature mapping functor is a cartesian functor between the corresponding fibred categories of signatures.*

Given a fibred institution $I = (\Pi : \mathrm{Sign} \to I, \mathrm{Mod}, \mathrm{Sen}, \models)$*, for each object* $i \in |I|$*, the* fibre of I at i *is the institution* $I^i = (\mathrm{Sign}^i, \mathrm{Mod}^i, \mathrm{Sen}^i, \models^i)$ *where*

- Sign^i *is the fibre of* Π *at* i*, and*
- Mod^i, Sen^i, *and* $\models^i$ *are the restrictions of* Mod, Sen, *and respectively* $\models$ *to* Sign^i.

Proposition 2.93. *[30] Given a fibred institution* $I = (\Pi : \mathrm{Sign} \to I, \mathrm{Mod}, \mathrm{Sen}, \models)$*, for each arrow* $u \in I(i,j)$*, any inverse image functor* $\Phi^u : \mathrm{Sign}^j \to \mathrm{Sign}^i$ *(with distinguished cartesian morphisms* $\varphi^u_{\Sigma'} : \Phi^u(\Sigma') \to \Sigma'$*) determines a canonical institution morphism* $(\Phi^u, \alpha^u, \beta^u) : I^j \to I^i$ *between the fibres of* I*, where for each signature* Σ' *in the fibre* Sign^j *at* j*,* $\alpha^u_{\Sigma'} = \mathrm{Sen}(\varphi^u_{\Sigma'})$ *and* $\beta^u_{\Sigma'} = \mathrm{Mod}(\varphi^u_{\Sigma'})$.

Definition 2.94. *[30] The* Grothendieck institution $\mathcal{J}^\sharp = (\mathrm{Sign}^\sharp, \mathrm{Sen}^\sharp, \mathrm{Mod}^\sharp, \models^\sharp)$ *of an indexed institution* $\mathcal{J} : I^{\mathrm{op}} \to \mathbb{I}ns$ *is defined as follows:*

1. *Let* $\mathrm{Sign} : I^{\mathrm{op}} \to \mathbf{CAT}$ *be the indexed institution mapping each index* i *to* Sign^i *and each index morphism* u *to* Φ^u*; then the category of the signatures of* $\mathcal{J}^\sharp$ *is the Grothendieck category* $\mathrm{Sign}^\sharp$*. Thus the signatures of* $\mathcal{J}^\sharp$ *consist of pairs* $\langle i, \Sigma \rangle$ *with* i *index and* $\Sigma \in |\mathrm{Sign}^i|$ *and signature morphisms* $\langle u, \varphi \rangle : \langle i, \Sigma \rangle \to \langle i', \Sigma' \rangle$ *consists of index morphisms* $u : i \to i'$ *and signature morphisms* $\varphi : \Sigma \to \Phi^u(\Sigma')$.
2. *The model functor* $\mathrm{Mod}^\sharp : (\mathrm{Sign}^\sharp)^{\mathrm{op}} \to \mathbf{CAT}$ *is given by*
   - $\mathrm{Mod}^\sharp(\langle i, \Sigma \rangle) = \mathrm{Mod}^i(\Sigma)$ *for each index* $i \in |I|$ *and signature* $\Sigma \in |\mathrm{Sign}^i|$*, and*
   - $\mathrm{Mod}^\sharp(\langle u, \varphi \rangle) = \beta^u_{\Sigma'}; \mathrm{Mod}^i(\varphi)$ *for each* $\langle u, \varphi \rangle : \langle i, \Sigma \rangle \to \langle i', \Sigma' \rangle$.
3. *The sentence functor* $\mathrm{Sen}^\sharp : \mathrm{Sign}^\sharp \to \mathbf{Set}$ *is given by*
   - $\mathrm{Sen}^\sharp(\langle i, \Sigma \rangle) = \mathrm{Sen}^i(\Sigma)$ *for each index* $i \in |I|$ *and signature* $\Sigma \in |\mathrm{Sign}^i|$*, and*
   - $\mathrm{Sen}^\sharp(\langle u, \varphi \rangle) = \mathrm{Sen}^i(\varphi); \alpha^u_{\Sigma'}$ *for each* $\langle u, \varphi \rangle : \langle i, \Sigma \rangle \to \langle i', \Sigma' \rangle$.
4. *The satisfaction relation is given by*

   $$M \models^\sharp_{\langle i, \Sigma \rangle} e \text{ if and only if } M \models^i_\Sigma e$$

   for each index $i \in |I|$*, signature* $\Sigma \in |\mathrm{Sign}^i|$*, model* $M \in |\mathrm{Mod}^\sharp(\langle i, \Sigma \rangle)|$*, and sentence* $e \in \mathrm{Sen}^\sharp(\langle i, \Sigma \rangle)$.

The following shows that the above construction gives an institution indeed.

Proposition 2.95. *[30]* $\mathcal{J}^\sharp$ *is an institution. Moreover, for each index* $i \in |I|$ *there exists a canonical institution morphism* $(\Phi^i, \alpha^i, \beta^i) : \mathcal{J}^i \to \mathcal{J}^\sharp$ *mapping any signature* $\Sigma \in |\mathrm{Sign}^i|$ *to* $\langle i, \Sigma \rangle \in |\mathrm{Sign}^\sharp|$ *and such that the components of* α^i *and* β^i *are identities.*

In [117] the author argues that Grothendieck institutions can be constructed using comorphisms instead of morphisms, and moreover that comorphism-based Grothendieck institutions may be more friendly towards some important model theoretic properties than the morphism-based ones.

Definition 2.96. *[117, 38] Given a category* I *of indices, an* indexed comorphism-based institution *, in short called* indexed co-institution*,* $\mathcal{J}$ *is a functor* $\mathcal{J} : I^{\mathrm{op}} \to co\mathbb{I}ns$*. (Recall that* $co\mathbb{I}ns$ *is the quasi-category having institutions as objects and institution comorphisms as arrows.) Its Grothendieck institution* $\mathcal{J}^\sharp$ *is defined as follows:*

1. *its category of signatures is* $((\mathrm{Sign}; (\_)^{\mathrm{op}})^\sharp)^{\mathrm{op}}$ *where* $\mathrm{Sign} : I^{\mathrm{op}} \to \mathbf{CAT}$ *is the* indexed *category of signatures of the indexed co-institution* $\mathcal{J}$*,* $(\_)^{\mathrm{op}} : \mathbf{CAT} \to \mathbf{CAT}$ *is the 'opposite' functor, and* $(\mathrm{Sign}; (\_)^{\mathrm{op}})^\sharp$ *is its Grothendieck category; this means that*

- *signatures are pairs* $\langle i, \Sigma \rangle$ *for* $i \in |I|$ *index and* $\Sigma \in |\mathrm{Sign}^i|$, *and*
- *signature morphisms are pairs* $\langle u, \varphi \rangle : \langle i, \Sigma \rangle \to \langle i', \Sigma' \rangle$ *where* $u \in I(i', i)$ *and* $\varphi \in \mathrm{Sign}^{i'}(\Phi^u(\Sigma), \Sigma')$,

2. *its model functor* $\mathrm{Mod}^\sharp : (\mathrm{Sign}; (\_)^{\mathrm{op}})^\sharp \to \mathbf{CAT}$ *is given by*
   - $\mathrm{Mod}^\sharp(\langle i, \Sigma \rangle) = \mathrm{Mod}^i(\Sigma)$ *for each index* $i \in |I|$ *and signature* $\Sigma \in |\mathrm{Sign}^i|$, *and*
   - $\mathrm{Mod}^\sharp(\langle u, \varphi \rangle) = \mathrm{Mod}^{i'}(\varphi); \beta^u_\Sigma$ *for each* $\langle u, \varphi \rangle : \langle i', \Sigma' \rangle \to \langle i, \Sigma \rangle$,
3. *its sentence functor* $\mathrm{Sen}^\sharp : ((\mathrm{Sign}; (\_)^{\mathrm{op}})^\sharp)^{\mathrm{op}} \to \mathbf{Set}$ *is given by*
   - $\mathrm{Sen}^\sharp(\langle i, \Sigma \rangle) = \mathrm{Sen}^i(\Sigma)$ *for each index* $i \in |I|$ *and signature* $\Sigma \in |\mathrm{Sign}^i|$, *and*
   - $\mathrm{Sen}^\sharp(\langle u, \varphi \rangle) = \alpha^u_\Sigma; \mathrm{Sen}^{i'}(\varphi)$ *for each* $\langle u, \varphi \rangle : \langle i', \Sigma' \rangle \to \langle i, \Sigma \rangle$,
4. $M \models^\sharp_{\langle i, \Sigma \rangle} e$ *if and only if* $M \models^i_\Sigma e$ *for each index* $i \in |I|$, *signature* $\Sigma \in |\mathrm{Sign}^i|$, *model* $M \in |\mathrm{Mod}^\sharp(\langle i, \Sigma \rangle)|$, *and sentence* $e \in \mathrm{Sen}^\sharp(\langle i, \Sigma \rangle)$.

where $\mathcal{I}^i = (\mathrm{Sign}^i, \mathrm{Mod}^i, \mathrm{Sen}^i, \models^i)$ *for each index* $i \in |I|$ *and* $\mathcal{I}^u = (\Phi^u, \alpha^u, \beta^u)$ *for* $u \in I$ *index morphism.*

Proposition 2.97. *[117, 38] The comorphism-based Grothendieck institution* $\mathcal{I}^\sharp$ *is indeed an institution, i.e., the satisfaction condition holds.*

With respect to the choice between the morphism or the comorphism styles of Grothendieck institutions, the following result gives a quite common situation in the concrete applications when these yield the same result. This result may also be interpreted as an invariance of the concept of Grothendieck institution with respect to the duality between the concepts of institution morphism and comorphism given by an adjunction between the signature categories that was discovered in [4].

Proposition 2.98. *[38] For each dual pair of an adjoint-indexed institution* $\mathcal{I}$ *and an adjoint-indexed coinstitution* $\overline{\mathcal{I}}$ *their Grothendieck institutions* $\mathcal{I}^\sharp$ *and* $\overline{\mathcal{I}}^\sharp$ *are isomorphic.*

An important example when Prop. 2.98 applies is that of CafeOBJ where the respective diagram of institutions is indeed an adjoint-indexed institution, which means that the CafeOBJ institution may be defined either as a morphism-based Grothendieck institution (like originally in [56]) or as a comorphism-based institution (like adopted by later works on the semantics of CafeOBJ, e.g. [42]).

In the paper [30] the author also answers the problem of a Grothendieck institutions as universal constructions. This answer represents an extension of the traditional Grothendieck construction on categories to institutions:

Theorem 2.99. *[30] The Grothendieck institution is the lax co-limit of the respective indexed institution in the 2-category of institutions.*

2.3.2 *Lifting local properties to global properties*

A most important theme in the theory of Grothendieck institutions has been that of the lifting of institution theoretic properties from the local level of the component institutions to the global level of the Grothendieck institution. With the exception of free constructions, all of them can be done more easily within comorphism-based contexts. All of the lifting properties presented in this section have been developed first in [28], and

later on upgraded in [30] for morphism-based Grothendieck institutions and in [117] for comorphism-based institutions. Here we present them in their comorphism-based version like in [38].

The problem of co-limits of signatures in the Grothendieck institution is reduced to the problem of co-limits in Grothendieck categories:

Definition 2.100. *[30, 117, 38] For any category J we say that an indexed co-institution $\mathcal{J} : I^{\mathrm{op}} \to co\mathbb{I}ns$* supports J-co-limits *when*

- *the index category I is J-complete, i.e., has J-limits,*
- *the indexed category of signatures* $\mathrm{Sign} : I^{\mathrm{op}} \to \mathbb{C}at$ *of $\mathcal{J}$ is* locally J-co-complete*, i.e.,* Sign^i *has all J-co-limits for each index $i \in |J|$, and*
- *for each index morphism u, the comorphism $\mathcal{J}^u$ preserves J-co-limits of signatures (meaning that the corresponding sentence translation functors Φ^u preserve pushouts).*

Theorem 2.101. *[30, 117, 38] The category of theories $\mathrm{Th}^{\mathcal{J}^\sharp}$ of a comorphism-based Grothendieck institution $\mathcal{J}^\sharp$ has J-co-limits if the indexed co-institution $\mathcal{J}$ supports J-co-limits.*

The following solves the problem of model amalgamation in Grothendieck institutions.

Definition 2.102. *[30, 117, 38] An indexed coinstitution $\mathcal{J} : I^{\mathrm{op}} \to co\mathbb{I}ns$ is* locally (semi-)exact *if and only if the institution $\mathcal{J}^i$ is (semi-)exact for each index $i \in I$.*

Proposition 2.103. *[30, 117, 38] Let $\mathcal{J} : I^{\mathrm{op}} \to co\mathbb{I}ns$ be a co-institution which supports pushouts. Then the semi-exactness of the Grothendieck institution $\mathcal{J}^\sharp$ implies the local semi-exactness of the indexed co-institution $\mathcal{J}$.*

Proposition 2.104. *[30, 117, 38] If the Grothendieck institution of an indexed co-institution $\mathcal{J}$ which supports pushouts is semi-exact, then each institution comorphism $\mathcal{J}^u = (\Phi^u, \alpha^u, \beta^u)$ is exact.*

Definition 2.105. *[30, 117, 38] An indexed co-institution $\mathcal{J} : I^{\mathrm{op}} \to co\mathbb{I}ns$ is* semi-exact *if and only if for each pullback*

$$\begin{array}{ccc} i & \xleftarrow{u1} & j1 \\ {\scriptstyle u2}\uparrow & & \uparrow{\scriptstyle v1} \\ j2 & \xleftarrow[v2]{} & k \end{array}$$

in I and each signature Σ in I^i, the square

$$\begin{array}{ccc} \mathrm{Mod}^i(\Sigma) & \xleftarrow{\beta^{u1}_{\Sigma}} & \mathrm{Mod}^{j1}(\Phi^{u1}(\Sigma)) \\ {\scriptstyle \beta^{u2}_{\Sigma}}\uparrow & & \uparrow{\scriptstyle \beta^{v1}_{\Phi^{u1}(\Sigma)}} \\ \mathrm{Mod}^{j2}(\Phi^{u2}(\Sigma)) & \xleftarrow[\beta^{v2}_{\Phi^{u2}(\Sigma)}]{} & \mathrm{Mod}^{k}(\Phi^{vi}(\Phi^{ui}(\Sigma))) \end{array}$$

is a pullback.

Proposition 2.106. *[30, 117, 38] If the Grothendieck institution $\mathcal{J}^\sharp$ of an indexed co-institution $\mathcal{J} : I^{\mathrm{op}} \to co\mathbb{I}ns$ which supports pushouts is semi-exact, then $\mathcal{J}$ is also semi-exact.*

Theorem 2.107. *[30, 117, 38] Let $\mathcal{J} : I^{\mathrm{op}} \to co\mathbb{I}ns$ be an indexed coinstitution which supports pushouts. Then the Grothendieck institution $\mathcal{J}^\sharp$ is semi-exact if and only if*

1. *the indexed coinstitution $\mathcal{J}$ is locally semi-exact,*
2. *the indexed coinstitution $\mathcal{J}$ is semi-exact, and*
3. *all institution comorphisms are exact.*

Interpolation is lifted from the component institutions to the Grothendieck institution as follows.

Definition 2.108. *[35] A commuting square of institution comorphisms*

$$\begin{array}{ccc} I & \xrightarrow{(\Phi_1,\alpha_1,\beta_1)} & I_1 \\ {\scriptstyle (\Phi_2,\alpha_2,\beta_2)}\downarrow & & \downarrow{\scriptstyle (\Phi'_1,\alpha'_1,\beta'_1)} \\ I_2 & \xrightarrow[(\Phi'_2,\alpha'_2,\beta'_2)]{} & I' \end{array}$$

is a Craig Interpolation square *if for each I-signature Σ, for each set E_1 of $\Phi_1(\Sigma)$-sentences and for each set E_2 of $\Phi_2(\Sigma)$-sentences, if $(\alpha'_1)_{\Phi_1(\Sigma)}(E_1) \models' (\alpha'_2)_{\Phi_2(\Sigma)}(E_2)$, then there exists a set E of Σ-sentences such that $E_1 \models^{I_1} (\alpha_1)_\Sigma(E)$ and $(\alpha_2)_\Sigma(E) \models^{I_2} E_2$.*

$$\begin{array}{ccc} \mathrm{Sen}(\Sigma) & \xrightarrow{(\alpha_1)_\Sigma} & \mathrm{Sen}^1(\Phi_1(\Sigma)) \\ {\scriptstyle (\alpha_2)_\Sigma}\downarrow & & \downarrow{\scriptstyle (\alpha'_1)_{\Phi_1(\Sigma)}} \\ \mathrm{Sen}^2(\Phi_2(\Sigma)) & \xrightarrow[(\alpha'_2)_{\Phi_2(\Sigma)}]{} & \mathrm{Sen}'(\Phi'_k(\Phi_k(\Sigma))) \end{array}$$

Theorem 2.109. *[35] Let $\mathcal{J} : I^{\mathrm{op}} \to co\mathbb{I}ns$ be an indexed coinstitution which supports pushouts such that*

- *there are fixed classes of index morphisms $\mathcal{L}, \mathcal{R} \subseteq I$ containing all identities, and*
- *for each index $i \in |I|$ there are fixed classes of signature morphisms $\mathcal{L}^i, \mathcal{R}^i \subseteq \mathrm{Sign}^i$ containing all identities,*

such that

- *$\mathcal{L}$ and $\mathcal{R}$ are stable under pullbacks,*
- *$\Phi^u(\mathcal{R}^i) \subseteq \mathcal{R}^j$ for each index morphism $u : j \to i$ in $\mathcal{L}$, and*
- *$\Phi^u(\mathcal{L}^i) \subseteq \mathcal{L}^j$ for each index morphism $u : j \to i$ in $\mathcal{R}$.*

Let $\mathcal{L}^\sharp$, and $\mathcal{R}^\sharp$, be the classes of signature morphisms $\langle u : j \to i, \varphi \rangle$ of the Grothendieck institution such that $u \in \mathcal{L}$, respectively $u \in \mathcal{R}$, and $\varphi \in \mathcal{L}^j$, respectively $\varphi \in \mathcal{R}^j$.

Then the Grothendieck institution $\mathcal{J}^\sharp$ has the Craig $(\mathcal{L}^\sharp, \mathcal{R}^\sharp)$-interpolation property if and only if

1. *for each index i the institution $\mathcal{J}^i$ has the $(\mathcal{L}^i, \mathcal{R}^i)$-interpolation property,*
2. *each pullback square of index morphisms*

$$\begin{array}{ccc} & \xleftarrow{\mathcal{L}} & \\ \mathcal{R}\uparrow & & \uparrow \\ & \xleftarrow{} & \end{array}$$

determines a Craig interpolation square of institution comorphisms,

3. *for each* $u: j \to i$ *in* $\mathcal{L}$ *the institution comorphism* $\mathcal{J}^u = (\Phi^u, \alpha^u, \beta^u)$ *has the Craig* $\mathcal{R}^i$*-right interpolation property, and*
4. *for each* $u: j \to i$ *in* $\mathcal{R}$ *the institution comorphism* $\mathcal{J}^u$ *has the Craig* $\mathcal{L}^i$*-left interpolation property.*

A rather spectacular application of the interpolation result of Thm. 2.109 in institutional model theory, that has not been necessarily envisaged when developing the (specification motivated) interpolation theory for Grothendieck institutions, is the following.

Theorem 2.110. *[38] Consider a conservative institution comorphism* $(\Phi, \alpha, \beta): I \to I'$ *and classes* $\mathcal{L}^i, \mathcal{R}^i$ *of signature morphisms in* I *such that*

1. I *and* I' *have pushouts of signatures and* Φ *preserves pushouts,*
2. *the institution comorphism* (Φ, α, β) *has Craig* $\mathcal{L}^i$*-left interpolation,*
3. I' *has implications and it is quasi-compact, and*
4. I' *has Craig* $(\Phi(\mathcal{L}^i), \Phi(\mathcal{R}^i))$*-interpolation.*

Then the institution I *has Craig-Robinson* $(\mathcal{L}^i, \mathcal{R}^i)$*-interpolation.*

The practical importance of this result is that it provides an efficient way to establish Craig-Robinson interpolation properties in logics without implications, such as equational logic, Horn clause logic, etc. A list of such concrete applications of Thm. 2.110 in fragments of first order logic is given in [38].

2.3.3 *Grothendieck inclusion systems*

In the case of inclusion systems in a heterogeneous framework established as a Grothendieck institution the following important practical problem arises:

> Assuming that each 'local' institution (of an indexed system of institutions) comes equipped with an inclusion system for its category of the signatures, do we have a canonical inclusion system for signatures of the corresponding 'global' Grothendieck institution?

In [42] we have provided a definitive and complete answer to this problem by considering also an inclusion system for the indexation, i.e. for the category of the indices. In fact, this problem is only about the signatures of institutions, which means that in this case we have a problem only about categories equipped with inclusion systems, the concept of institution not being needed in this work. The canonicity of our Grothendieck inclusion system construction (Thm. 2.116) is expressed as an universal property (Prop. 2.120). Moreover, this construction applies also to contexts that are very different from the main motivation of our work, namely that of the inclusion systems for the signatures of Grothendieck institutions underlying multi-logic heterogeneous specification. In [42] there are various examples of 'strong' inclusion systems in use in concrete specification frameworks, or from model theory, or from abstract deductive theories. Moreover the inclusion system for Kripke semantics developed in [51] is of the same kind. Although each of these arise in a particular context, they have a common flavour (hence they are known in the literature

as 'strong') which can be explained in the light of the work in [42] as being instances of a same general construction. The paper [42] also explores how some properties of inclusion systems which are important for the semantics of specification languages lift to Grothendieck inclusion systems.

The problem solved in [42] has received preliminary answers in [28] and [30], however the constructions proposed there can now be regarded as partial cases of the main construction in [42] (Thm. 2.116).

The definition below gives the concept of (homo)morphism of inclusion systems.

Definition 2.111 (Inclusive functors)**.** *A functor* $\mathcal{U} : \langle I, \mathcal{E}\rangle \to \langle I', \mathcal{E}'\rangle$ *(between the underlying categories of the inclusion systems) is* inclusive *when it preserves the inclusions, i.e.* $\mathcal{U}(I) \subseteq I'$.

Fact 2.112. *Inclusion systems and inclusive functors form a category denoted* $\mathbb{IS}$. *Moreover,* $\mathbb{IS}$ *can be endowed with a 2-categorical structure with the 2-cells being defined as natural transformations between inclusive functors such that all their components are inclusions.*

The equational definition of the concept of adjunction by the so-called 'triangular laws' (see [103]) permits the well known generalization of the concept of adjunction from **CAT** to abstract 2-categories. Let us call adjunctions defined in $\mathbb{IS}$ as $\mathbb{IS}$*-adjunctions*.

Fact 2.113. *An* $\mathbb{IS}$*-adjunction between inclusion systems consists of an adjunction between the underlying categories such that both the left and the right adjoints are inclusive functors and such that all components of the unit and the co-unit are inclusions.*

Definition 2.114. *[42] An* enriched indexed inclusion system *is a functor* $B : \langle I, \mathcal{E}\rangle \to \mathbb{IS}^{\mathrm{op}}$ *from the underlying category of an inclusion system 'of indices' to the opposite of the category of inclusions systems and inclusive functors.*

Definition 2.115. *[42] An enriched indexed inclusion system* $B : \langle I, \mathcal{E}\rangle \to \mathbb{IS}^{\mathrm{op}}$ *is* invertible *when for each index morphism u, the corresponding inclusive functor* B^u *has an* $\mathbb{IS}$*-left-adjoint denoted* $[-]^u$. *It is* $\mathcal{E}$-invertible *when the* $\mathbb{IS}$*-left-adjoint to* B^u *exists for* $u \in \mathcal{E}$ *(and not necessarily for all index morphisms u).*

The following establishes the existence of Grothendieck inclusion systems.

Theorem 2.116. *[42] For any* $\mathcal{E}$*-invertible enriched indexed inclusion system* $B : \langle I, \mathcal{E}\rangle \to \mathbb{IS}^{\mathrm{op}}$ *the Grothendieck category* $B^\sharp$ *of* B^{op}; $(\mathbb{IS} \to \mathbf{CAT})$ *(from the opposite of the underlying category of* $\langle I, \mathcal{E}\rangle$ *to* **CAT***) can be endowed with an inclusion system* $\langle I^\sharp, \mathcal{E}^\sharp\rangle$ *such that* $\langle u, \varphi\rangle : \langle j, \Sigma\rangle \to \langle j', \Sigma'\rangle$ *is*

- *an* abstract inclusion *iff both u and* φ *are abstract inclusions, and*
- *an* abstract surjection *iff u is an abstract surjection and* $\Sigma' = [\varphi(\Sigma)]^u$.

The following result develops sufficient conditions for unions for Grothendieck inclusion systems and requires invertible enriched indexed inclusion systems which is stronger than the $\mathcal{E}$-invertible condition of Thm. 2.116.

Proposition 2.117. *[42] For any invertible enriched indexed inclusion system* $B : \langle I, \mathcal{E}\rangle \to \mathbb{IS}^{\mathrm{op}}$*, the Grothendieck inclusion system* $\langle I^\sharp, \mathcal{E}^\sharp\rangle$ *has unions if*

– *the inclusion system of indices* $\langle I, \mathcal{E}\rangle$ *has unions, and*
– *for each index* j *the 'local' inclusion system* $B^j = \langle I^j, \mathcal{E}^j\rangle$ *has unions.*

The following result develops sufficient conditions for the Grothendieck inclusion systems to be epic.

Proposition 2.118. *[42] In addition to the conditions of Thm. 2.116 if the inclusion system of the indices* $\langle I, \mathcal{E}\rangle$ *is epic,* $B^j = \langle I^j, \mathcal{E}^j\rangle$ *is epic for each index* j*, and* B^u *are faithful for* $u \in \mathcal{E}$*, then the inclusion system* $\langle I^\sharp, \mathcal{E}^\sharp\rangle$ *defined in Thm. 2.116 is epic too.*

The following result characterizes the Grothendieck inclusion system $\langle I^\sharp, \mathcal{E}^\sharp\rangle$ defined by Thm. 2.116 as a lax colimit enriched by $\mathbb{IS}$. This refines the characterization of Grothendieck categories as lax colimits [30] and is also consonant to the characterization of Grothendieck institutions as a lax colimit in the 2-category of institutions [30].

Definition 2.119 ($\mathbb{IS}$-lax colimits)**.** *[42] For any pair of functors* $F, G : \langle I, \mathcal{E}\rangle \to \mathbb{IS}^{\mathrm{op}}$ *(from the underlying category of an inclusion system* $\langle I, \mathcal{E}\rangle$*), a* $\mathbb{IS}$-lax natural transformation $\mu : F \Rightarrow G$ *is a lax natural transformation such that*

– *for any object* j *of* $\langle I, \mathcal{E}\rangle$*, the functor* $\mu^j : F(j) \to G(j)$ *is inclusive, and*
– *for any* $u \in I$*, the natural transformation* μ^u *is abstract inclusion (for the inclusion system of the corresponding functor category).*

$\mathbb{IS}$-lax co-cone *and* $\mathbb{IS}$-lax colimits*, respectively, are just lax co-cone and lax colimits, respectively, which are* $\mathbb{IS}$*-lax as natural transformations.*

Proposition 2.120. *[42] For any* $\mathcal{E}$*-invertible* $\mathbb{IS}$*-enriched indexed inclusion system* $B : \langle I, \mathcal{E}\rangle \to \mathbb{IS}^{\mathrm{op}}$*, the Grothendieck inclusion system* $\langle I^\sharp, \mathcal{E}^\sharp\rangle$ *defined by Thm. 2.116 is the* $\mathbb{IS}$*-lax co-limit of* B*.*

2.4 CAFEOBJ

Most of the theoretical achievements reviewed in this chapter have been motivated and inspired by the work on the formal specification language CafeOBJ. Often they appeared as solutions or answers to the foundational problems that have emerged from the design process of the language or from its associated methodologies.

CafeOBJ is, together with CASL [5], a modern algebraic specification language. It is a natural successor of the famous OBJ language [85] that has a direct realisation of some very new important trends in algebraic specification such as rewriting logic and behavioural specification. Its definition [54] and its first implementation have been developed in Japan between 1996-2000 within the framework of the large scale project funded by the government. Various methodologies for formal specification and verification with CafeOBJ are still being developed, and their foundations still constitute a source of interesting research challenges. A recent overview of the process that has lead to the CafeOBJ definition and to the development of some of the most important theoretical contributions of CafeOBJ to specification theory is given in [50].

In this section we overview briefly some of CafeOBJ most important features.

2.4.1 *Equational specification and programming*

Equational specification and programming is inherited from OBJ [85, 67] and constitutes the basis of the language, the other features being somehow built on top of it. As with OBJ, CafeOBJ is *executable* (by term rewriting), which gives an elegant declarative way of functional programming, often referred as *algebraic programming*. Although this paradigm may be used as programming, from the applications point of view, this aspect is secondary to its specification side. As with OBJ, CafeOBJ also permits equational specification modulo several equational theories such as associativity, commutativity, identity, idempotence, and combinations between all these. This feature is reflected at the execution level by term rewriting *modulo* such equational theories. The underlying logic of this specification paradigm in CafeOBJ is order sorted algebra, a refinement of many sorted algebra that supports a limited form of partiality through subsorting; this will be discussed in more detail in a following section.

2.4.2 *Behavioural specification*

CafeOBJ behavioural specification paradigm is based on *coherent hidden algebra* of [55], the institution denoted *HA* above. CafeOBJ directly supports behavioural specification and its proof theory through special language constructs, such as

- hidden sorts (for states of systems),
- behavioural operations (for direct "actions" and "observations" on states of systems),
- behavioural coherence declarations for (non-behavioural) operations (which may be either derived (indirect) "observations" or "constructors" on states of systems), and
- behavioural axioms (stating behavioural satisfaction).

The main behavioural proof method is based on *coinduction*. In CafeOBJ, coinduction can be used either in the classical hidden algebra sense [76] for proving behavioural equivalence of states of objects (see Sect. 2.1.1), or for proving behavioural transitions (see Sect. 1.2.3).

Besides language constructs, CafeOBJ supports behavioural specification and verification by several methodologies. CafeOBJ highlights the methodology for concurrent object composition reviewed in Sect. 2.1.2 which features high reusability not only of specification code but also of verifications [54, 97, 36]. Behavioural specification in CafeOBJ may also be effectively used as an object-oriented (state-oriented) alternative for classical data-oriented specifications. Experiments seem to indicate that an object-oriented style of specification even of basic data types (such as sets, lists, etc.) may lead to higher simplicity of code and drastic simplification of verification process [54, 58].

Behavioural specification is reflected at the execution level by the concept of *behavioural rewriting* [54, 55] which refines ordinary rewriting with a condition ensuring the correctness of the use of behavioural equations in proving strict equalities.

2.4.3 *Rewriting logic specification*

Rewriting logic specification in CafeOBJ is based on a simplified version of Meseguer's *rewriting logic* [113] specification framework for concurrent systems which gives a non-trivial extension of traditional algebraic specification towards concurrency. This simplification means that the CafeOBJ rewriting models are preorders rather than categories, which yields an unlabelled form of rewriting logic called preordered algebra (*POA*; see Sect. 1.2.3). This avoids many of the semantical complications resulting from the labelled version of rewriting logic, the worst of these being the failure of the satisfaction condition of institution theory.

POA incorporates many different models of concurrency in a natural, simple, and elegant way, thus giving CafeOBJ a wide range of applications. The unlabelled aspect of *POA* means that, unlike Maude [23], the CafeOBJ design does not support full reasoning about multiple transitions between states (or system configurations), but provides proof support for reasoning about the *existence* of transitions between states (or configurations) of concurrent systems. This is achieved via a built-in predicate with dynamic definition encoding into equational logic both the proof theory of RWL and the user defined transitions (rules).

From a methodological perspective, CafeOBJ develops the use of *POA* for specifying and verifying the properties of declarative encoding of algorithms (see [54]) as well as for specifying and verifying transition systems.

2.4.4 *Module system*

The principles of the CafeOBJ module system are inherited from OBJ which builds on ideas first realized in the language Clear [18]. However the foundations of the CafeOBJ module system are based upon the most recent developments in the theory of structured specifications including those reviewed in Sect. 2.2. The CafeOBJ module system features

- several kinds of imports,
- sharing for multiple imports,
- parameterised programming allowing
  - multiple parameters,
  - views for parameter instantiation,
  - integration of CafeOBJ specifications with executable code in a lower level language
- module expressions.

Compared to CASL [5], CafeOBJ module system offers more specification possibilities by allowing non-protecting importation modes and sharing when instantiating parameters of parameterised modules.

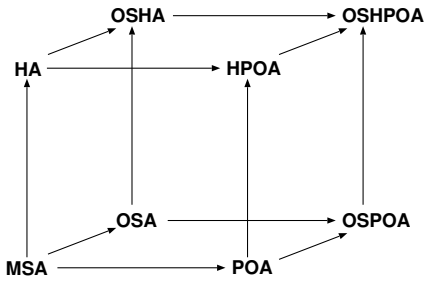
2.4.5 *Type system and partiality*

CafeOBJ has a type system that allows subtypes based on *order sorted algebra* [81, 74]. This provides a mathematically rigorous form of runtime type checking and error handling, giving CafeOBJ a syntactic flexibility comparable to that of untyped languages, while preserving all the advantages of strong typing. CafeOBJ does not directly do partial operations but rather handles them by using error sorts and a sort membership predicate in the style of *membership equational logic* [114].

2.4.6 *Grothendieck institutional semantics*

The design and definition of CafeOBJ represents a submission to the main principle underlying the design and definition of modern formal specification languages, namely that there is an underlying logic in which all language constructs can be rigorously defined as mathematical entities and such that the semantics of specifications or programs is given by the model theory of this underlying logic. Another notable example of a concrete realisation of this principle is CASL [5].

The heterogeneous nature of CafeOBJ has lead to the enrichment of institution theory with the concept of Grothendieck institution. The indexed institution, or the diagram of institutions underlying the semantics of CafeOBJ, is given by the notorious *CafeOBJ cube* which in its comorphism-based form is depicted below. (The actual CafeOBJ cube consists of the full arrows, the dotted arrows denote the comorphisms from components of the indexed institution to the Grothendieck institution.)



Unfortunately, the time of the writing of [54] was prior to that of the definition of Grothendieck institutions, thus the CafeOBJ definition presented in [54] treated heterogeneity of the language through the theory developed in [28]. Soon this has been upgraded to Grothendieck institutions in [56] which however presented the CafeOBJ institution as morphism-based Grothendieck institution. Since the CafeOBJ cube is an adjoint-indexed institution, Prop. 2.98 applies which means that the CafeOBJ institution can be equally presented as a comorphism-based Grothendieck institution. Recent works on the semantics of CafeOBJ (e.g. [41, 42]) adopt this perspective. The full details of the institutions of the CafeOBJ cube can be found in [56]. Below we present them briefly.

At the bottom of the CafeOBJ cube lies *MSA*, the institution of many-sorted algebra that has 'algebraic signatures' (consisting of sets of sort symbols and sorted function symbols) as signatures, algebras interpreting the sort symbols as sets and the function symbols as functions, and (possibly conditional) uni-

versally quantified equations as sentences. The satisfaction between algebras and equations is the standard Tarskian satisfaction. As in other algebraic specification languages, conditions of equations are encoded as Boolean-values terms, hence in reality *MSA* should be thought like in [40].

OSA extends the *MSA* institution with order sortedness such that the set of sorts of a signature is a partially ordered set rather than a discrete set, and algebras interpret the subsort relationship as set inclusion. The embedding institution comorphism from *MSA* to *OSA* is the obvious one that interprets any *MSA* signature as a discrete *OSA* signature.

POA has been already discussed in Sect. 1.2.3; it has the same signatures as *MSA* but the models interpret the sort symbols as preorders and the function symbols as preorder functors (i.e. functors between preorders). Besides equations, *POA* has other sentences too, the so-called 'transitions' which can be regarded as of one-directional equations not obeying the symmetry rule. Their satisfaction by the models is determined by the preorder relation between the interpretation of the terms of the transition. The embedding institution comorphism from *MSA* to *POA* just forgets the preorder relationship between the elements of models.

The *HA* node of the CafeOBJ cube represents a slight restriction of the *HA* institution discussed above in that the behavioural operations are required to have only one hidden sort in their arity. This nature of this restriction has rather methodological nature, so from the point of view of foundations it does not have any meaning. This means future versions of the CafeOBJ definition may remove this restriction. Besides ordinary strict or behavioural equations, the CafeOBJ *HA* has also 'coherence declarations' which can be regarded as just abbreviations for a special type of conditional behavioural equations (see [55, 56]) for details. The embedding comorphism $MSA \to HA$ just regards the *MSA* signature as *HA* signatures with a trivial behavioural structure.

These extensions of *MSA* towards three different paradigms can be all combined into *HOPOA* (details can be found in [56, 41]) which contains all institutions of the CafeOBJ cube as sub-institutions. Various extensions of CafeOBJ towards new paradigms can be further considered by transforming the CafeOBJ cube into a 'hyper-cube' and by flattening it to a Grothendieck institution, so in this sense the Grothendieck institutional semantics of CafeOBJ is an open one.

2.5 OTHER ACHIEVEMENTS

2.5.1 *Herbrand theorems for abstract logic programming*

The logic programming paradigm [104] in its purely logical form can be described as follows:

> Given a universal Horn (finite) theory (Σ, E) (called 'program', with Σ the 'signature' of the program, i.e., the set of its symbols, and E the set of Σ-sentences) and an existentially quantified conjunction of atoms ρ (called 'query') in $\Sigma \cup Y$ (for Y a new set of 'logical variables'), find a 'solution' ψ for ρ, i.e., values for the variables Y, such that the corresponding instance $\psi[\rho]$ of ρ is satisfied by (Σ, E).

In other words, we need that $(\Sigma, E) \models (\exists Y)\rho$.

In the most conventional form, logic programming is considered over unsorted first order logic without equality [104], less conventional forms of logic programming extends this to multiple sorts, or even considers first order logic with equality as underlying logic [124, 78, 79], this being considered as a new related paradigm, and known under the name of 'equational logic programming'. An extension object-oriented extension of equational logic programming has been proposed in [77]. However, a careful look at the logic programming paradigm shows its semantical foundations are essentially institution-independent.

The basic logic programming concepts, query, solutions, solution forms, and the fundamental results, such as Herbrand theorems, can be developed over any institution by employing institution-independent concepts of variable, substitution, quantifiers, atomic formulaæ, most of them being part of the 'internal logic' of institutions developed in [31]. The institution-independent concept of substitution (see Sect. 1.1.2) is developed for the first time in [33], being one of the main contributions of that work.

The paper [33] sets foundations for an uniform development of logic programming over a large variety of computing science logics, which opens the door for a clean combination between logic programming and various other computing paradigms. In this 'institution-independent' framework in [33] there is also a discussion of some basic modularisation issues for logic programming.

In this section we review the most important developments in [33].

Definition 2.121 (Query). *[33] Given a signature Σ in an arbitrary institution $(\mathrm{Sign}, \mathrm{Sen}, \mathrm{Mod}, \models)$ with a designated class $\mathcal{D}$ of signature morphisms, a* $\mathcal{D}$-query *is any existentially quantified sentence $(\exists \chi)\rho$ such that $\chi \in \mathcal{D}$ is quasi-representable and ρ is a basic sentence.*

Theorem 2.122 (First Herbrand theorem). *[33] In an institution consider a theory (Σ, E) which has an initial model $0_{\Sigma,E}$. Then for each query $(\exists \chi)\rho$,*

$$E \models (\exists \chi)\rho \text{ if and only if } 0_{\Sigma,E} \models (\exists \chi)\rho.$$

Definition 2.123 (Solutions). *[33] Each χ-expansion N' of $0_{\Sigma,E}$ such that $N' \models \rho$ is called a* solution *for the query $(\exists \chi)\rho$.*

Theorem 2.124 (Second Herbrand theorem). *[33] Consider an institution with representable $\mathcal{D}$-substitutions for a class $\mathcal{D}$ of representable signature morphisms such that*

1. *for each theory (Σ, E) with initial model, its signature Σ also has an initial model 0_Σ,*
2. *for any theory (Σ, E) having an initial model $0_{\Sigma,E}$, for each signature morphism $(\chi : \Sigma \to \Sigma') \in \mathcal{D}$ its representation M_χ is projective with respect to all 'quotient' homomorphisms $p_{\Sigma,E} : 0_\Sigma \to 0_{\Sigma,E}$.*

Then for each theory (Σ, E) having an initial model, and for any $\mathcal{D}$-query $(\exists \chi_1)\rho$ we have that

$$E \models (\exists \chi_1)\rho \text{ if and only if there exists a } \mathcal{D}\text{-substitution } \psi : \chi_1 \to \chi_2 \text{ such that } E \models (\forall \chi_2)\psi(\rho)$$
and χ_2 is conservative.

When instantiated to institutions of predefined types like in [29, 47, 38], the abstract Herbrand theorems above provide a clean denotational semantics for constraint logic programming based on initial models. This

shows that at the denotational level constraint programming is just a form of ordinary logic programming when seen from a higher conceptual perspective.

This abstract approach to logic programming proposed in [33] has been recently applied also for providing an institution theoretic semantics to services [26].

2.5.2 *Abstract structural induction*

Since its introduction within computing science by Burstall [16] structural induction has become a major method for performing inductive proofs, which constitute one of the most important formal verification trends. Originally structural induction was confined to proving properties of abstract data types, specified within many-sorted algebra (*MSA*). But over the past decades due to the population explosion of underlying logics for specification formalisms, the meaning and scope of structural induction has been extended to logical systems that are increasingly sophisticated and different from *MSA*. However these structural induction proof methodologies are often developed on a rather ad-hoc basis without clear mathematical foundations, a situation that in our opinion ultimately undermines the credibility of the associated formal methods.

In [44] we develop a generic method for proving inductive properties, that is directly applicable to wide variety of logic based specification formalisms, already in existence or that may be developed in the future. The genericity of our structural induction method is given by the fact that it is developed at the level of abstract institutions, and it therefore lacks a commitment to a particular logical system. The main contributions of this work are as follows:

1. The development of an axiomatic theory of substitutions for abstract institutions that is based upon and refines the general institution-independent concept of substitution introduced in [33] (see also [38]). This serves as the technical ground for the development of the institution-independent structural induction method.
2. The core result of [44] is the structural induction theorem (Thm. 2.129). A particularly important feature of these concrete methodologies emerging from this general results and that differs from other formulations of structural induction in the literature (in fact mostly within *MSA*) is that they allow *simultaneous induction on several variables*. This owes to the fact that we do not restrict X of $(\forall X)\rho$ to a single variable, it may rather represent a block of variables. This comes naturally from approaching the concepts of variable and substitution from an abstract institution theoretic perspective.
3. An abstract theory of constructors that can be adjoined to the main structural induction theorem, and that in concrete situations leads to a reduction in the complexity of the proof process associated to the respective concrete proof methodologies.

The work [44] develops also instances of the abstract concepts and results in a series of non-conventional and non-classical logical systems from specification theory and computer science, such as preordered algebra,

many-valued logic, many-sorted algebra with predefined types (for constraint programming), partial algebra. In what follows we review the main technical developments of [44].

Definition 2.125 (Variables)**.** *For any signature Σ of an institution, a signature morphism $X : \Sigma \to \Sigma'$ is called a* Σ-variable*. Usually we will denote Σ', the target signature of X, by $\Sigma(X)$.*

Definition 2.126 (System of substitutions)**.** *[44] A* system of substitutions *in a given institution consists of a $|\mathrm{Sign}|$-indexed family $\mathcal{S} = \{\mathcal{S}_\Sigma \mid \Sigma \in |\mathrm{Sign}|\}$ such that for each $\Sigma \in |\mathrm{Sign}|$, $\mathcal{S}_\Sigma$ is a sub-category of the category of the Σ-substitutions and such that*

1. *$1_\Sigma \in |\mathcal{S}_\Sigma|$,*
2. *for each $X \in |\mathcal{S}_\Sigma|$ and any signature morphism $\iota : \Sigma \to \Sigma'$ there exists a pushout of signature morphisms*

$$\begin{array}{ccc} \Sigma & \xrightarrow{\iota} & \Sigma' \\ {\scriptstyle X}\downarrow & & \downarrow{\scriptstyle X'} \\ \Sigma(X) & \xrightarrow[\iota(X)]{} & \Sigma'(X') \end{array}$$

   such that $X' \in |\mathcal{S}_{\Sigma'}|$,
3. *for any $X, Y \in |\mathcal{S}_\Sigma|$ and any functor F making the diagram below commute*

$$\begin{array}{ccc} \mathrm{Mod}(\Sigma(Y)) & \xrightarrow{F} & \mathrm{Mod}(\Sigma(X)) \\ & {\scriptstyle \mathrm{Mod}(Y)} \searrow \quad \swarrow {\scriptstyle \mathrm{Mod}(X)} & \\ & \mathrm{Mod}(\Sigma) & \end{array}$$

   there exists a unique $\theta \in \mathcal{S}_\Sigma$ such that $F = \mathrm{Mod}(\theta)$.

The Σ-substitutions that belong to $\mathcal{S}_\Sigma$ are called $\mathcal{S}_\Sigma$-substitutions*.*

Definition 2.127 (Substitutions with depth)**.** *[44] A* depth measure *d for a system $\mathcal{S}$ of substitutions in an institution is a family of functions from the substitutions to the set ω of the natural numbers, $d = \{d_\Sigma : \mathcal{S}_\Sigma \to \omega \mid \Sigma \in |\mathrm{Sign}|\}$, such that*

1. *$d(1_X) = 0$ for any $X \in |\mathcal{S}_\Sigma|$, and*
2. *for any $\theta : X \dashrightarrow Y$ and $\theta' : Y \dashrightarrow Z$ in $\mathcal{S}_\Sigma$ we have that $d(\theta;\theta') \leq d(\theta) + d(\theta')$.*

The substitutions θ with $d(\theta) = 0$ are called flat substitutions*.*

Definition 2.128 (Atomic substitutions)**.** *[44] In an institution, given a system of substitutions $\mathcal{S}$ with a depth measure d, a designated subclass $\mathsf{At}_\Sigma \subseteq \mathcal{S}_\Sigma$, for any signature Σ, is called a subclass of* atomic substitutions *when*

1. *any flat substitution $\theta : X \dashrightarrow 1_\Sigma$ is atomic,*
2. *for each non flat $\mathcal{S}_\Sigma$-substitution θ there are $\mathcal{S}_\Sigma$-substitutions Q and T such that $\theta = Q;T$, $Q \in \mathsf{At}_\Sigma$, and $d(T) < d(\theta)$.*

The following generic structural induction theorem is the main result of [44].

Theorem 2.129 (Structural induction)**.** *[44] Let us consider a semi-exact institution with pushouts of signatures, equipped with:*

- *a system of substitutions $\mathcal{S}$,*
- *a depth measure d for $\mathcal{S}$,*
- *a system of atomic substitutions At for $\mathcal{S}$ and d, and*
- *a binary relation $\sqsubset$ on each set $\mathsf{At}(X,Y)$ such that*

$$\psi \sqsubset Q \text{ implies } \psi \text{ is flat and } Q \text{ is not flat.}$$

Let $\iota : \Omega \to \Sigma$ be a signature morphism, let $X \in |\mathcal{S}_\Omega|$ and let $X' \in |\mathcal{S}_\Sigma|$ be defined by the following pushout square:

$$\begin{array}{ccc} \Omega & \xrightarrow{\iota} & \Sigma \\ {\scriptstyle X}\downarrow & & \downarrow{\scriptstyle X'} \\ \Omega(X) & \xrightarrow[\iota(X)]{} & \Sigma(X') \end{array}$$

Let Γ be a set of Σ-sentences and ρ be a $\Sigma(X')$-sentence such that, for every atomic $\mathcal{S}_\Omega$-substitution $Q : X \dashrightarrow Z$ and every pushout square:

$$\begin{array}{ccc} \Omega & \xrightarrow{\iota} & \Sigma \\ {\scriptstyle Z}\downarrow & & \downarrow{\scriptstyle Z'} \\ \Omega(Z) & \xrightarrow[\iota(Z)]{} & \Sigma(Z') \end{array}$$

with $Z' \in |\mathcal{S}_\Sigma|$ we have:

$$Z'(\Gamma) \cup \{(\psi \star \iota)(\rho) \mid \psi \sqsubset Q\} \models_{\Sigma(Z')} (Q \star \iota)(\rho).$$

Then for all $\mathcal{S}_\Omega$-substitutions $\theta : X \dashrightarrow 1_\Omega$:

$$\Gamma \models_\Sigma (\theta \star \iota)(\rho).$$

Let us make the following comments with respect to Thm. 2.129.

1. In the applications ι represents the so-called 'sub-signatures of constructors'. A general treatment at the level of abstract institutions of this rather well-established concept, followed by examples, is given below. Constructors have only a pure methodological role, namely that of reducing the complexity of the proof process, a 'smaller' Ω leading to a smaller number of substitutions Q and hence a smaller number of proof goals. If we disregarded this efficiency aspect, then we could very well do without constructors, a situation that corresponds to setting ι of Thm. 2.129 to the identity 1_Σ. In such a case, the statement of Thm. 2.129 gets simplified with $\Omega = \Sigma$, $\iota(X)$ and $\iota(Z)$ being identities, $X = X'$ and $Z = Z'$.
2. The parameter $\sqsubset$ represents the main heuristic aspect of Thm. 2.129 and in actual situations the setting of its value is a key factor in defining actual structural induction methodologies. In setting $\sqsubset$ one should consider that a smaller $\sqsubset$ means fewer hypotheses for the proof goals of the associated structural induction methodology, a situation that may result in severe difficulties in the proof process.

On the other hand, it is crucial to ensure the finiteness of the proof process through the finiteness of the set $\{\psi \mid \psi \sqsubset Q\}$. In the concrete instances of Thm. 2.129 presented below in this section $\sqsubset$ is set in a rather uniform way, which means that at the abstract level of Thm. 2.129 at this moment the parameter $\sqsubset$ may be seen mostly as an axiomatization device. However it seems a promising subject of further research to come up with concrete values for $\sqsubset$ leading to concrete structural induction methodologies alternative to those presented below in this section.

3. A crucial aspect of Thm. 2.129 is that it is supposed to represent a *finitary* proof process. We have already discussed one of the conditions for this, namely the finiteness of $\{\psi \mid \psi \sqsubset Q\}$. The other condition is the finiteness of the number of the (atomic) substitutions Q (from the statement of the theorem), which in the actual cases may be guaranteed by the finiteness of the signatures and by the atomicity of the substitutions Q (see the examples below in this section). Note also that the latter finiteness condition should be considered modulo isomorphism classes of Z and of $\Sigma(Z')$.

The remaining part of this section reviews the abstract approach to constructors developed in [44].

Definition 2.130 (Sub-signature of constructors)**.** *[44] In any institution, for any class $\mathcal{E}$ of model homomorphisms, a signature morphism $\iota : \Omega \to \Sigma$ is a* sub-signature of $\mathcal{E}$-constructors for a set Γ *of Σ-sentences when*

- *Γ has an initial model 0_Γ,*
- *the signature Ω has an initial model 0_Ω, and*
- *the unique Ω-homomorphism $(0_\Omega \to 0_\Gamma\restriction_\iota)$ is in $\mathcal{E}$.*

Definition 2.131. *In any institution, given any class $\mathcal{E}$ of model homomorphisms, a model M is* $\mathcal{E}$-projective *when for each homomorphism $(h : A \to B) \in \mathcal{E}$ and each homomorphism $g : M \to B$ there exists a homomorphism $f : M \to A$ such that $f;h = g$.*

$$\begin{array}{ccc} M & \xrightarrow{f} & A \\ & \searrow^{g} & \downarrow{\scriptstyle h} \\ & & B \end{array}$$

Definition 2.132 (Representable substitutions)**.** *[33, 38, 44]*

An institution with a system of substitutions $\mathbb{S}$ has representable $\mathbb{S}$-substitutions *when*

1. *each $\mathbb{S}$-variable $X \in |\mathbb{S}_\Sigma|$ is representable, and*
2. *for each $X, Y \in |\mathbb{S}_\Sigma|$ and $h : M_X \to M_Y$ there exists a $\mathbb{S}_\Sigma$-substitution $\theta : X \dashrightarrow Y$ such that the following diagram commutes:*

$$\begin{array}{ccc} \mathrm{Mod}(\Sigma(Y)) & \xrightarrow{\mathrm{Mod}(\theta)} & \mathrm{Mod}(\Sigma(X)) \\ {\scriptstyle i_Y}\downarrow & & \downarrow{\scriptstyle i_X} \\ M_Y/\mathrm{Mod}(\Sigma) & \xrightarrow[h;(-)]{} & M_X/\mathrm{Mod}(\Sigma) \end{array}$$

The following constitutes the main result about constructors in [44], and represents a result about the correctness of the structural induction method of Thm. 2.129, showing that this actually proves inductive properties.

Proposition 2.133. *[44] In any institution*

1. *with model amalgamation,*
2. *with a designated class $\mathcal{E}$ of model homomorphisms, and*
3. *with a system $\mathbb{S}$ of representable substitutions such that M_X is $\mathcal{E}$-projective for each $X \in |\mathbb{S}_\Sigma|$,*

let $\iota : \Omega \to \Sigma$ be a sub-signature of $\mathcal{E}$-constructors for a set Γ of Σ-sentences and let $X : \Omega \to \Omega(X) \in |\mathbb{S}_\Omega|$. Let E be a set of Σ-sentences such that $0_\Gamma \models E$. Then for any pushout square of signature morphisms such that $X' \in |\mathbb{S}_\Sigma|$

$$\begin{array}{ccc} \Omega & \xrightarrow{\iota} & \Sigma \\ {\scriptstyle X}\downarrow & & \downarrow{\scriptstyle X'} \\ \Omega(X) & \xrightarrow[\iota(X)]{} & \Sigma(X') \end{array}$$

if for some $\Sigma(X')$-sentence ρ we have that $\Gamma \cup E \models (\theta \star \iota)(\rho)$ for each $\mathbb{S}_\Omega$-substitution $\theta : X \dashrightarrow 1_\Omega$ then $0_\Gamma \models (\forall X')\rho$.

Part II

Future Evolution

3

SCIENTIFIC EVOLUTION

Scientific evolution can never be planned in detail. This is mainly because any serious personal scientific agenda is highly dependent upon the global scientific evolutions which are increasingly difficult to predict in a climat dominated by fast technological developments and a high degree of interdependency between scientific areas. All these arguments are even more valid when we talk about computer science related research. Therefore in this section I would like to outline my *intentions* regarding my future scientific activity based upon an evaluation of the current situation.

3.1 GENERAL SCIENTIFIC EVOLUTION

In general terms, the future evolution of my scientific activity will be along the same coordinates of the past thirty years. In other words, I do not envisage a dramatic change of direction or style of research. Below I present a list of arguments supporting this intention.

1. Although more the three decades have passed since the inception of institution theory, this research area continues to enjoy a very dynamic growth, today even at an accelerated pace. This is in sharp contrast to most of the computer science theoretical developments that are often short lived and with very little impact. Some of the reasons for this continuing success of institution theory are as follows:
    - The impact of institution theory and its associated methodology to approach logic, computing and software phenomena has extended over a wide range of scientific interests, from philosophy and logic to applied computing and engineering and even medical sciences. As witness to this impact is that currently institution theory papers have been published in more the thirty (30) journals of good international reputation, from computer science, logic, mathematics, philosophy, and moreover this list continues to grow. An increasing number of doctoral theses are been developed worldwide in relation to institution theory (I have recently supervised myself on of those in Europe and I have been in the committees of several others).
    - The strong advance of the universal trend in (mathematical) logic over the past ten to fifteen years, process led by Jean-Yves Beziau, has coagulated a rather big worldwide community (known as UNILOG after the name of the series of big world congress and schools already organised in different parts of the world) that is taking a genuine interest in institution theory mainly from the logic side. This is something that has not been envisaged by the fathers of institution theory, although conceptually they draw a lot of inspiration from universal logic ideas at the time (although these were not organised under such name). Apart of the series of UNILOG schools and congresses, this community is also publishing a journal and a book series,

both at Springer, and has some dedicated corners in some reputed logic journals. All these have become yet another platform for the institution theoretic research.

- Institution theory displays a conceptual and mathematical elegance difficult to find in other computer science theories, hence it is intellectually very appealing. This aspect has been stated as one of the strength of institution theory by many important researchers worldwide.

2. Whilst the traditional themes in institution theory have not yet exhausted their potential to rise new research challenges, the growth of institution theory has led to the opening of new research avenues.

3. Formal specification is the originally envisaged application for institution theory. In the past years, according to a number of experts, the software engineering community is beginning to take a new interest in formal methods. This means a new cycle in the life of formal methods.

4. Taking into account the situation described above, since for many worldwide my name is associated with the development of institution theory over the past twenty plus years, I have a personal responsibility to continue to support this research area both in theoretical and applied aspects.

3.2 SPECIFIC COORDINATES

Having established an intention for evolution as a natural continuation of my past scientific activity, this does not mean that the emphasis will remain on exactly the same research topics as before. So there will be some change of focus. As a general principle, the personal research interests will be much guided by those of the community interested in institution theory and applications. Genuine research challenges will have to be addressed.

For example there will be less focus on the development of classical model theory in a general and abstract institution theoretic setting. This line of research has made already a lot of progress over the past ten plus years, and moreover in the past few years several younger researchers worldwide have already produced interesting results in this area, so now my presence here is less required. The situation is however very different with respect to non-classical logics versus institution theory. While non-classical logics can be captured properly as institutions, some of their fine aspects may be beyond the conventional institution theory. For example, the institutions of many-valued logics handle the ternary aspect of the satisfaction relation by adjoining truth value to the sentences; in this way we get a binary satisfaction relation. While this works well with respect to most aspects of many-valued logics, it cannot handle graded (non-binary) consequences. The same with modal logics, the conventional concept of institution does not allow for a general semantics of modalities. These shortcomings have to be overcome by extensions of the definition of institution towards non-classical aspects of logics. These hint to some concrete priorities for the next years in the area of non-classical logics and applications.

1. Development of a many-valued in-depth refinement of institution theory with applications to artificial intelligence (including fuzzy logic programming, formal systems for medical sciences [93], etc.).

2. Development of an in-depth refinement of institution theory towards modalities, with applications to novel specification and verification paradigms (such as specifications of dynamically reconfigurable systems [106], but not only).

These envisaged developments will be strongly related with the preparation in the next years of a new edition of [38]. This is an important personal project. Whilst the first edition of [38] has got a great success, the high dynamics of institution theory requires a rather substantial upgrade of this monograph.

The research in formal specification and verification will remain an important component part of my future scientific activity. The context of this is determined by the following factors.

1. In spite of the huge progress in the past few decades on the foundations of formal specification, still a series of foundational questions regarding various of its associated methodologies remain. For example recently we have started a project about the semantics of multiply parameterised specifications [53, 25] with implications to modularisation methodologies, a proper study of the structuring of behavioural specifications has only very recently started [52], etc.

2. A generic logic independent in-depth approach to formal verification has been initiated only recently with the institution theoretic work on structural induction [44]; this is an area that has a good theoretical and applicative potential.

3. The recent development of the heterogeneous specification paradigm, heavily based on institution theory, has opened a new research area. Although much progress have been made both in theoretical and applied aspects, a lot still needs to be done.

4. The increasing sophistication of modern software systems requires new specification paradigms which have to be addressed with adequate theoretical means.

5. Ideas and concepts from algebraic specifications have often found new homes in other computer science area, such as logic programming [78], services [26], ontologies [72, 101, 122] (an ISO standard for ontologies being developed with institution theory playing a crucial role), etc.

4

ETHICS

In the recent years there is a growing awareness in the international scientific community regarding the role of ethics in academia. For example while years ago international publishers used only to issue warnings about plagiarism, these days they often ask authors to sign form declarations and even organise web seminars (like Elsevier does) on this topic. But this is only a gross form of unethical scientific behaviour. In a climate increasingly dominated by the competition and marketisation of research there are many more subtle forms of unethical behaviour than may be much more damaging to the academic environment than gross plagiarism (which anyway it is not so difficult to fight against). The issue of ethics is crucial for maintaining a healthy and conducive academic environment that promotes high intelectual standards. Therefore these days it is obligatory for any senior scientist to make clear his position with respect to relevant ethical issues, and to act accordingly. This is even more valid for those working with students of junior scientists, because the future climate in academia will depend on the ethical scientific behaviour of the young generation of scientists. Therefore any plan of a future academic and scientific evolution should address the ethical component.

Generally speaking, compared with other sciences, computer science has some specific heavy problems regarding academic ethics. These are often related to the perception of outside scientists about computer science as having generally low intelectual and scientific standards. These may be also dramatically amplified across geographical areas where there is a general culture of corruption in the society. Some of these problems may be identified as follows:

- Big publication lists containing mostly very minor contributions published in conferences, often in domestic ones, rather than in international journals of good reputation. The peer-reviewing process for conferences in general is quick, and for most of them very superficial and often based on group interests reflected in the PC committee. Very often there is a big overlapping between the contents of those papers, which shows a form of auto-plagiarism. Of course, the impact of such works is minimal, with very low citation numbers. A discussion on this issue has been initiated at the highest international level in [146], from which I give the following quotation:

  > An old joke tells of a driver, returning home from a party where he had one drink too many, who hears a warning over the radio about a car careening down the wrong side of the highway. “A car?” he wondered aloud, “There are lots of cars on the wrong side of the road!” I am afraid that driver is us, the computing-research community.

- In relation to the previous issue, often computer science publications have long lists of authors, many of them being in fact a kind of “honorary” authors rather than real ones. Senior scientists, leaders of research teams, often use their power position for achieving authorship of papers without an enough

substantial contribution to their respective content. To be such an author is enough to provide a few very general indications, or even just to have a grant from which to hire junior scientists to do the real work. In many cases these senior scientists do not even understand the technical content of the respective papers.

- To be a PhD student in computer science today may be quite different from twenty years ago. These days PhD students are usually hired on project grants and have to work in order to fullfil the research agendas of their professors. There is very little or no concern about the scientific development of the student in accordance with their own talent or scientific inclinations. This is in fact a form of exploitation, which is very different from when I was brought up as scientist during my DPhil years at Oxford. Although founded from a grant, my Professor gave me full freedom to chose my research topic, and was taking a keen interest in trying together with me to discover my main interests in science.

An important number of paramount, highly respected scientists have showed by their life example that there can be an ethical way to high academic achievements. I know very well a few cases: Jean-Yves Beziau, Rod Burstall, Joseph Goguen, Don Sannella, Andrzej Tarlecki, etc. An important part of my future activity is to promote their style of conducting research and bringing up young scientists. This will be done firstly by personal example and secondly by explaining young students and scientists the benefits of a correct scientific behaviour. In particular this means:

- journal publications rather than conference publications;
- international publications in places of good reputation rather than domestic publications;
- authorship based only upon substantial contribution;
- offer students and junior scientists a proper academic rather than a business climate.

BIBLIOGRAPHY

[1] Marc Aiguier and Fabrice Barbier. An institution-independent proof of the Beth definability theorem. *Studia Logica*, 85(3):333–359, 2007.

[2] Marc Aiguier and Răzvan Diaconescu. Stratified institutions and elementary homomorphisms. *Information Processing Letters*, 103(1):5–13, 2007.

[3] Eyal Amir and Sheila McIlraith. Improving the efficiency of reasoning through structure-based reformulation. In B.Y. Choueiry and T.Walsh, editors, *Proceedings of the Symposium on Abstraction, Reformulation and Approximation (SARA'2000)*, volume 1864 of *Lecture Notes in Artificial Intelligence*, pages 247–259. Springer-Verlag Berlin Heidelberg, 2000.

[4] M. Arrais and José L. Fiadeiro. Unifying theories in different institutions. In Magne Haveraaen, Olaf Owe, and Ole-Johan Dahl, editors, *Recent Trends in Data Type Specification*, volume 1130 of *Lecture Notes in Computer Science*, pages 81–101. Springer, 1996.

[5] Edigio Astesiano, Michel Bidoit, Hélène Kirchner, Berndt Krieg-Brückner, Peter Mosses, Don Sannella, and Andrzej Tarlecki. CASL: The common algebraic specification language. *Theoretical Computer Science*, 286(2):153–196, 2002.

[6] Jan Bergstra, Jan Heering, and Paul Klint. Module algebra. *Journal of the Association for Computing Machinery*, 37(2):335–372, 1990.

[7] Jean-Yves Béziau. 13 questions about universal logic. *Bulletin of the Section of Logic*, 35(2/3):133–150, 2006.

[8] Jean-Yves Béziau, editor. *Universal Logic: an Anthology*. Studies in Universal Logic. Springer Basel, 2012.

[9] Michel Bidoit, Rolf Hennicker, and Martin Wirsing. Behavioural and abstractor specifications. *Sci. Comput. Program.*, 25(2-3):149–186, 1995.

[10] Michel Bidoit, Donald Sannella, and Andrzej Tarlecki. Observational interpretation of CASL specifications. *Mathematical Structures in Computer Science*, 18(2):325–371, 2008.

[11] Garrett Birkhoff. On the structure of abstract algebras. *Proceedings of the Cambridge Philosophical Society*, 31:433–454, 1935.

[12] Patrick Blackburn and Jerry Seligman. Hybrid languages. *Journal of Logic, Language and Information*, 4(3):251–272, 1995.

[13] Francis Borceux. *Handbook of Categorical Algebra*. Cambridge University Press, 1994.

[14] Tomasz Borzyszkowski. Logical systems for structured specifications. *Theoretical Computer Science*, 286(2):197–245, 2002.

[15] Peter Burmeister. Partial algebra - an introductory survey. *Algebra Universalis*, 15:306–358, 1982.

[16] Rod Burstall. Proving properties of programs by structural induction. *Computer Journal*, 12(1):41–48, 1969.

[17] Rod Burstall and Joseph Goguen. Putting theories together to make specifications. In Raj Reddy, editor, *Proceedings, Fifth International Joint Conference on Artificial Intelligence*, pages 1045–1058. Department of Computer Science, Carnegie-Mellon University, 1977.

[18] 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.

[19] Walter Carnielli and Marcelo Esteban Coniglio. Combining logics. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. 2011.

[20] Maria-Victoria Cengarle. *Formal specifications with higher-order parameterization*. PhD thesis, Ludwig-Maximilians-Universitat, Munchen, 1994.

[21] Chen-Chung Chang and H. Jerome Keisler. *Model Theory*. North Holland, Amsterdam, 1990.

[22] Petr Cintula and Petr Hájek. On theories and models in fuzzy predicate logic. *Journal of Symbolic Logic*, 71(3):832–863, 2006.

[23] Manuel Clavel, Francisco Durán, Steven Eker, Patrick Lincoln, Narciso Martí-Oliet, José Meseguer, and Carolyn Talcott. *All About Maude - A High-Performance Logical Framework*, volume 4350 of *Lecture Notes in Computer Science*. Springer, 2007.

[24] Mihai Codescu and Daniel Găină. Birkhoff completeness in institutions. *Logica Universalis*, 2(2):277–309, 2008.

[25] Ionuţ Ţuţu. Parameterisation for abstract structured specifications. *Theoretical Computer Science*. DOI:10.1016/j.tcs.2013.11.008.

[26] Ionuţ Ţuţu and Jose Fiadeiro. A logic-programming semantics of services. In R. Heckel and S. Milius, editors, *CALCO 2013*, volume 8089 of *Lecture Notes in Computer Science*, pages 299–313, 2013.

[27] Virgil Emil Căzănescu and Grigore Roşu. Weak inclusion systems. *Mathematical Structures in Computer Science*, 7(2):195–206, 1997.

[28] Răzvan Diaconescu. Extra theory morphisms for institutions: logical semantics for multi-paradigm languages. *Applied Categorical Structures*, 6(4):427–453, 1998. A preliminary version appeared as JAIST Technical Report IS-RR-97-0032F in 1997.

[29] Răzvan Diaconescu. Category-based constraint logic. *Mathematical Structures in Computer Science*, 10(3):373–407, 2000.

[30] 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.

[31] Răzvan Diaconescu. Institution-independent ultraproducts. *Fundamenta Informaticæ*, 55(3-4):321–348, 2003.

[32] Răzvan Diaconescu. Elementary diagrams in institutions. *Journal of Logic and Computation*, 14(5):651–674, 2004.

[33] Răzvan Diaconescu. Herbrand theorems in arbitrary institutions. *Information Processing Letters*, 90:29–37, 2004.

[34] Răzvan Diaconescu. An institution-independent proof of Craig Interpolation Theorem. *Studia Logica*, 77(1):59–79, 2004.

[35] Răzvan Diaconescu. Interpolation in Grothendieck institutions. *Theoretical Computer Science*, 311:439–461, 2004.

[36] Răzvan Diaconescu. Behavioural specification of hierarchical object composition. *Theoretical Computer Science*, 343(3):305–331, 2005.

[37] Răzvan Diaconescu. Proof systems for institutional logic. *Journal of Logic and Computation*, 16(3):339–357, 2006.

[38] Răzvan Diaconescu. *Institution-independent Model Theory*. Birkhäuser, 2008.

[39] Răzvan Diaconescu. An encoding of partial algebras as total algebras. *Information Processing Letters*, 109(23–24):1245–1251, 2009.

[40] Răzvan Diaconescu. Quasi-boolean encodings and conditionals in algebraic specification. *Journal of Logic and Algebraic Programming*, 79(2):174–188, 2010.

[41] Răzvan Diaconescu. Coinduction for preordered algebras. *Information and Computation*, 209(2):108–117, 2011.

[42] Răzvan Diaconescu. Grothendieck inclusion systems. *Applied Categorical Structures*, 19(5):783–802, 2011.

[43] Răzvan Diaconescu. On quasi-varieties of multiple valued logic models. *Mathematical Logic Quarterly*, 57(2):194–203, 2011.

[44] Răzvan Diaconescu. Structural induction in institutions. *Information and Computation*, 209(9):1197–1222, 2011.

[45] Răzvan Diaconescu. An axiomatic approach to structuring specifications. *Theoretical Computer Science*, 433:20–42, 2012.

[46] Răzvan Diaconescu. Borrowing interpolation. *Journal of Logic and Computation*, 22(3):561–586, 2012.

[47] Răzvan Diaconescu. Interpolation for predefined types. *Mathematical Structures in Computer Science*, 22(1):1–24, 2012.

[48] Răzvan Diaconescu. Three decades of institution theory. In Jean-Yves Béziau, editor, *Universal Logic: an Anthology*, pages 309–322. Springer Basel, 2012.

[49] Răzvan Diaconescu. Institutional semantics for many-valued logics. *Fuzzy Sets and Systems*, 218:32–52, 2013.

[50] Răzvan Diaconescu. CafeOBJ traces. In S. Iida, J. Meseguer, and K. Ogata, editors, *Specification, Algebra, and Software*, Lecture Notes in Computer Science. Springer, 2014.

[51] Răzvan Diaconescu. Quasi-varieties and initial semantics in hybridized institutions. *Journal of Logic and Computation*, DOI:10.1093/logcom/ext016.

[52] Răzvan Diaconescu and Ionuţ Ţuţu. Foundations for structuring behavioural specifications. *Journal of Logic and Algebraic Programming*. To appear.

[53] Răzvan Diaconescu and Ionuţ Ţuţu. On the algebra of structured specifications. *Theoretical Computer Science*, 412(28):3145–3174, 2011.

[54] 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.

[55] 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.

[56] Răzvan Diaconescu and Kokichi Futatsugi. Logical foundations of CafeOBJ. *Theoretical Computer Science*, 285:289–318, 2002.

[57] Răzvan Diaconescu, Kokichi Futatsugi, and Shusaku Iida. Component-based algebraic specification and verification in CafeOBJ. In Jeannette M. Wing, Jim Woodcock, and Jim Davies, editors, *FM'99 – Formal Methods*, volume 1709 of *Lecture Notes in Computer Science*, pages 1644–1663. Springer, 1999.

[58] Răzvan Diaconescu, Kokichi Futatsugi, and Shusaku Iida. CafeOBJ Jewels. In Kokichi Futatsugi, Ataru Nakagawa, and Tetsuo Tamai, editors, *Cafe: An Industrial-Strength Algebraic Formal Method*, pages 33–60. Elsevier, 2000.

[59] 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.

[60] Răzvan Diaconescu and Alexandre Madeira. Encoding hybridized institutions into first order logic. *Mathematical Structures in Computer Science*. To appear.

[61] Răzvan Diaconescu and Marius Petria. Saturated models in institutions. *Archive for Mathematical Logic*, 49(6):693–723, 2010.

[62] Răzvan Diaconescu and Petros Stefaneas. Ultraproducts and possible worlds semantics in institutions. *Theoretical Computer Science*, 379(1):210–230, 2007.

[63] Theodosis Dimitrakos. *Formal Support for Specification Design and Implementation*. PhD thesis, Imperial College, 1998.

[64] Theodosis Dimitrakos and Tom Maibaum. On the role of interpolation for stepwise refinement, 1997. In *Proceeding First Panhellenic Logic Symposium*.

[65] Theodosis Dimitrakos and Tom Maibaum. On a generalized modularization theorem. *Information Processing Letters*, 74:65–71, 2000.

[66] Werner Fey. Pragmatics, concepts, syntax, semantics and correctness notions of ACT TWO: An algebraic module specification and interconnection language. Technical Report 88–26, Technical University of Berlin, Fachbereich Informatik, 1988.

[67] Kokichi Futatsugi, Joseph Goguen, Jean-Pierre Jouannaud, and Jose Meseguer. Principles of OBJ2. In *Proceedings of the 12th ACM Symposium on Principles of Programming Languages*, pages 52–66. ACM, 1985.

[68] Dov M. Gabbay and Larisa Maksimova. *Interpolation and Definability: modal and intuitionistic logics*. Oxford University Press, 2005.

[69] Giangiacomo Gerla. Fuzzy logic programming and fuzzy control. *Studia Logica*, 79(231–254), 2005.

[70] Joseph Goguen. Reusing and interconnecting software components. *Computer*, 19(2):16–28, February 1986. Reprinted in *Tutorial: Software Reusability*, Peter Freeman, editor, IEEE Computer Society, 1987, pages 251–263, and in *Domain Analysis and Software Systems Modelling*, Rubén Prieto-Díaz and Guillermo Arango, editors, IEEE Computer Society, 1991, pages 125–137.

[71] 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.

[72] Joseph Goguen. Data, schema, ontology and logic integration. *Journal of IGPL*, 13(6):685–715, 2006.

[73] 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.

[74] Joseph Goguen and Răzvan Diaconescu. An Oxford survey of order sorted algebra. *Mathematical Structures in Computer Science*, 4(4):363–392, 1994.

[75] 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.

[76] Joseph Goguen and Grant Malcolm. A hidden agenda. *Theoretical Computer Science*, 245(1):55–101, 2000.

[77] Joseph Goguen, Grant Malcolm, and Tom Kemp. A hidden Herbrand theorem: Combining the object, logic and functional paradigms. *Journal of Logic and Algebraic Programming*, 51(1):1–41, 2002.

[78] Joseph Goguen and José Meseguer. Eqlog: Equality, types, and generic modules for logic programming. In Douglas DeGroot and Gary Lindstrom, editors, *Logic Programming: Functions, Relations and Equations*, pages 295–363. Prentice-Hall, 1986.

[79] Joseph Goguen and José Meseguer. Models and equality for logical programming. In Hartmut Ehrig, Giorgio Levi, Robert Kowalski, and Ugo Montanari, editors, *Proceedings, TAPSOFT 1987*, volume 250 of *Lecture Notes in Computer Science*, pages 1–22. Springer, 1987.

[80] Joseph Goguen and José Meseguer. Unifying functional, object-oriented and relational programming, with logical semantics. In Bruce Shriver and Peter Wegner, editors, *Research Directions in Object-Oriented Programming*, pages 417–477. MIT, 1987. Preliminary version in *SIGPLAN Notices*, Volume 21, Number 10, pages 153–162, October 1986.

[81] Joseph Goguen and José Meseguer. Order-sorted algebra I: Equational deduction for multiple inheritance, overloading, exceptions and partial operations. *Theoretical Computer Science*, 105(2):217–273, 1992. Also, Programming Research Group Technical Monograph PRG–80, Oxford University, December 1989.

[82] Joseph Goguen and Grigore Roşu. Hiding more of hidden algebra. In Jeannette M. Wing, Jim Woodcock, and Jim Davies, editors, *FM'99 – Formal Methods*, volume 1709 of *Lecture Notes in Computer Science*, pages 1704–1719. Springer, 1999.

[83] Joseph Goguen and Grigore Roşu. Institution morphisms. *Formal Aspects of Computing*, 13:274–307, 2002.

[84] Joseph Goguen and Grigore Roşu. Composing hidden information modules over inclusive institutions. In Olaf Owe, Stein Krogdahl, and Tom Lyche, editors, *From Object-Orientation to Formal Methods*, volume 2635 of *Lecture Notes in Computer Science*, pages 96–123. Springer, 2004.

[85] Joseph Goguen, Timothy Winkler, José Meseguer, Kokichi Futatsugi, and Jean-Pierre Jouannaud. Introducing OBJ. In Joseph Goguen and Grant Malcolm, editors, *Software Engineering with OBJ: algebraic specification in action*. Kluwer, 2000.

[86] Joseph Goguen and David Wolfram. On types and FOOPS. In Robert Meersman, William Kent, and Samit Khosla, editors, *Object Oriented Databases: Analysis, Design and Construction*, pages 1–22. North Holland, 1991. Proceedings, IFIP TC2 Conference, Windermere, UK, 2–6 July 1990.

[87] Alexandre Grothendieck. Catégories fibrées et descente. In *Revêtements étales et groupe fondamental, Séminaire de Géométrie Algébraique du Bois-Marie 1960/61, Exposé VI*. Institut des Hautes Études Scientifiques, 1963. Reprinted in Lecture Notes in Mathematics, Volume 224, Springer, 1971, pages 145–94.

[88] Daniel Găină and Kokichi Futatsugi. Initial semantics in logics with constructors. *Journal of Logic and Computation*, DOI:10.1093/logcom/exs044.

[89] Daniel Găină and Marius Petria. Completeness by forcing. *Journal of Logic and Computation*, 20(6):1165–1186, 2010.

[90] Daniel Găină and Andrei Popescu. An institution-independent generalization of Tarski's Elementary Chain Theorem. *Journal of Logic and Computation*, 16(6):713–735, 2006.

[91] Daniel Găină and Andrei Popescu. An institution-independent proof of Robinson consistency theorem. *Studia Logica*, 85(1):41–73, 2007.

[92] Petr Hájek. *Metamathematics of Fuzzy Logic*. Kluwer, 1998.

[93] Robert Helgesson. *Generalized General Logics*. PhD thesis, Umeå University, 2013.

[94] 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.

[95] Heinrich Hussmann. *Nondeterminism in Algebraic Specifications and Algebraic Program*. Birkhaüser, 1993.

[96] Shusaku Iida, Kokichi Futatsugi, and Răzvan Diaconescu. Component-based algebraic specifications: - behavioural specification for component based software engineering -. In *7th OOPSLA Workshop*

on Behavioral Semantics of OO Business and System Specification, pages 167–182, 1998. Also in the technical report of Technical University of Munich TUM-I9820.

[97] Shusaku Iida, Kokichi Futatsugi, and Răzvan Diaconescu. Component-based algebraic specification: - behavioural specification for component-based software engineering -. In *Behavioral specifications of businesses and systems*, pages 103–119. Kluwer, 1999.

[98] B. Jacobs and J.M. Rutten. A tutorial on (co)algebras and (co)induction. *Bulletin of EATCS*, 62:222–259, 1997.

[99] Ranjit Jhala, Rupak Majumdar, and Ru-Gang Xu. State of the union: Type inference via Craig interpolation. In *Tools and Algorithms for the Construction and Analysis of Systems*, volume 4424 of *Lecture Notes in Computer Science*, pages 553–567. Springer, 2007.

[100] Oliver Kutz and Till Mossakowski. Modules in transition. Conservativity, Composition, and Colimits. In *Proceedings, Second International Workshop on Modular Ontologies*, 2007.

[101] Oliver Kutz, Till Mossakowski, and Dominik Lücke. Carnap, Goguen, and the hyperontologies - logical pluralism and heterogeneous structuring in ontology design. *Logica Universalis*, 4(2):255–333, 2010.

[102] Yngve Lamo. *The Institution of Multialgebras – a general framework for algebraic software development*. PhD thesis, University of Bergen, 2003.

[103] Saunders Mac Lane. *Categories for the Working Mathematician*. Springer, second edition, 1998.

[104] John Lloyd. *Foundations of Logic Programming*. Springer, Berlin, 1988. Second, Extended edition.

[105] Jerzy Łoś. Quelques remarques, théorèmes et problèmes sur les classes définissables d'algèbres. In *Mathematical Interpretation of Formal Systems*, pages 98–113. North-Holland, Amsterdam, 1955.

[106] Alexandre Madeira. *Foundations and techniques for software reconfigurability*. PhD thesis, Universidades do Minho, Aveiro and Porto (Joint MAP-i Doctoral Programme), 2013.

[107] Anatoly Malcev. *The Metamathematics of Algebraic Systems*. North-Holland, 1971.

[108] 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.

[109] Günter Matthiessen. Regular and strongly finitary structures over strongly algebroidal categories. *Canadian Journal of Mathematics*, 30:250–261, 1978.

[110] Sheila McIlraith and Eyal Amir. Theorem proving with structured theories. In *Proceedings of the 17th Intl. Conf. on Artificial Intelligence (IJCAI-01)*, pages 624 – 631, 2001.

[111] Kenneth McMillan. Applications of Craig interpolants in model checking. In *Proceedings TACAS'2005*, volume 3440 of *Lecture Notes in Computer Science*, pages 1–12. Springer, 2005.

[112] José Meseguer. General logics. In H.-D. Ebbinghaus et al., editors, *Proceedings, Logic Colloquium, 1987*, pages 275–329. North-Holland, 1989.

[113] José Meseguer. Conditional rewriting logic as a unified model of concurrency. *Theoretical Computer Science*, 96(1):73–155, 1992.

[114] José Meseguer. Membership algebra as a logical framework for equational specification. In F. Parisi-Pressice, editor, *Proc. WADT'97*, volume 1376 of *Lecture Notes in Computer Science*, pages 18–61. Springer, 1998.

[115] T. Mossakowski, S. Autexier, and D. Hutter. Development graphs - proof management for structured specification. *Journal of Logic and Algebraic Programming*, 67(1-2):114–145, 2006.

[116] T. Mossakowski, C. Maeder, and K. Lütich. The heterogeneous tool set. In *Lecture Notes in Computer Science*, volume 4424, pages 519–522. 2007.

[117] Till Mossakowski. Comorphism-based Grothendieck logics. In K. Diks and W. Rytter, editors, *Mathematical foundations of computer science*, volume 2420 of *Lecture Notes in Computer Science*, pages 593–604. Springer, 2002.

[118] Till Mossakowski. Relating CASL with other specification languages: the institution level. *Theoretical Computer Science*, 286:367–475, 2002.

[119] Till Mossakowski. Heterogeneous specification and the heterogeneous tool set. Habilitation thesis, University of Bremen, 2005.

[120] Till Mossakowski, Răzvan Diaconescu, and Andrzej Tarlecki. What is a logic translation? *Logica Universalis*, 3(1):59–94, 2009.

[121] Till Mossakowski, Joseph Goguen, Răzvan Diaconescu, and Andrzej Tarlecki. What is a logic? In Jean-Yves Béziau, editor, *Logica Universalis*, pages 113–133. Birkhäuser, 2005.

[122] Till Mossakowski, Oliver Kutz, and Christoph Lange. Semantics of Distributed Ontology Language: Institutes and institutions. In *Recent Trends in Algebraic Development Techniques*, volume 7841 of *Lecture Notes in Computer Science*, pages 212–230, 2013.

[123] Greg Nelson and Derek Oppen. Simplication by cooperating decision procedures. *ACM Trans. Program. Lang. Syst.*, 1(2):245–257, 1979.

[124] Michael J. O'Donnell. *Computing in systems described by equation*, volume 58 of *Lecture Notes in Computer Science*. Springer, 1977.

[125] Derek Oppen. Complexity, convexity and combinations of theories. *Theoretical Computer Science*, 12:291–302, 1980.

[126] David Parnas. Information distribution aspects of design methodology. *Information Processing '72*, 71:339–344, 1972. Proceedings of 1972 IFIP Congress.

[127] David Parnas. On the criteria to be used in decomposing systems into modules. *Communications of the Association for Computing Machinery*, 15:1053–1058, 1972.

[128] David Parnas. A technique for software module specification. *Communications of the Association for Computing Machinery*, 15:330–336, 1972.

[129] Marius Petria and Răzvan Diaconescu. Abstract Beth definability in institutions. *Journal of Symbolic Logic*, 71(3):1002–1028, 2006.

[130] Florian Rabe. A framework for combining model and proof theory. *Mathematical Structures in Computer Science*, 23(5):945–1001, 2013.

[131] Horst Reichel. Behavioural equivalence – a unifying concept for initial and final specifications. In *Proceedings, Third Hungarian Computer Science Conference*. Akademiai Kiado, 1981. Budapest.

[132] Horst Reichel. *Initial Computability, Algebraic Specifications, and Partial Algebras*. Clarendon, 1987.

[133] Grigore Roşu. *Hidden Logic*. PhD thesis, University of California at San Diego, 2000.

[134] Grigore Roşu. Axiomatisability in inclusive equational logic. *Mathematical Structures in Computer Science*, 12(5):541–563, 2002.

[135] Grigore Roşu and Dorel Lucanu. Circular coinduction: A proof theoretical foundation. In Alexander Kurz, Marina Lenisa, and Andrzej Tarlecki, editors, *Algebra and Coalgebra in Computer Science*, volume 5728 of *Lecture Notes in Computer Science*, pages 127–144, 2009.

[136] Donald Sannella and Andrzej Tarlecki. Specifications in an arbitrary institution. *Information and Control*, 76:165–210, 1988.

[137] Donald Sannella and Andrzej Tarlecki. *Foundations of Algebraic Specifications and Formal Software Development*. Springer, 2012.

[138] Klaus Schneider. *Verification of reactive systems*. Springer, 2004.

[139] Joseph Shoenfield. *Mathematical Logic*. Addison-Wesley, 1967.

[140] 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.

[141] Andrzej Tarlecki. On the existence of free models in abstract algebraic institutions. *Theoretical Computer Science*, 37:269–304, 1986.

[142] Andrzej Tarlecki. Quasi-varieties in abstract algebraic institutions. *Journal of Computer and System Sciences*, 33(3):333–360, 1986.

[143] Alfred Tarski and Robert Vaught. Arithmetical extensions of relational systems. *Compositio Mathematicæ*, 13:81–102, 1957.

[144] Shahab Tasharrofi and Eugenia Ternovska. A semantic account for modularity in multi-language modelling of search problems. In *Frontiers of combining systems*, volume 6989 of *Lecture Notes in Computer Science*, pages 259–274, 2011.

[145] Johan van Bentham. *Modal Logic and Classical Logic*. Humanities Press, 1988.

[146] Moshe Vardi. Conferences vs. journals in computing research. *Communications of the ACM*, 52(5), 2009.

[147] Paulo Veloso. On pushout consistency, modularity and interpolation for logical specifications. *Information Processing Letters*, 60(2):59–66, 1996.

[148] Peter Vojtás. Fuzzy logic programming. *Fuzzy Sets and Systems*, 124:361–370, 2001.

[149] Michał Walicki. *Algebraic Specification of Nondeterminism*. PhD thesis, Department of Informatics, University of Bergen, 1993.

[150] Michał Walicki and Sigurd Meldal. Algebraic approaches to nondeterminism - an overview. *ACM Computing Surveys*, 29, 1997.

[151] Christoph Weidenbach, Uwe Brahm, Thomas Hillenbrand, Enno Keen, Christian Theobald, and Dalibor Topic. Spass version 2.0. In *Proceedings of the 18th International Conference on Automated Deduction*, CADE-18, pages 275–279, London, UK, 2002. Springer-Verlag.