Available at
www.ElsevierComputerScience.com
POWERED BY SCIENCE DIRECT®

Information Processing Letters 90 (2004) 29–37

Information Processing Letters

www.elsevier.com/locate/ipl

Herbrand theorems in arbitrary institutions

Răzvan Diaconescu

Institute of Mathematics "Simion Stoilow", PO Box 1-764, Bucharest 014700, Romania

Received 4 August 2003; received in revised form 18 November 2003

Communicated by J.L. Fiadeiro

Abstract

The basic logic programming semantic concepts, query, solutions, solution forms, and the fundamental results such as Herbrand theorems, are developed over any logical system, formalised as institution, by employing 'institution-independent' concepts of variable, substitution, quantifier, and atomic formulae. This sets semantical foundations for a uniform development of logic programming over a large variety of computing science logics, allowing for a clean combination of logic programming with other computing paradigms.

Keywords: Institutions; Logic programming; Herbrand theorems; Specification languages; Formal methods

1. Introduction

The theory of institutions [1] is a categorical abstract model theory which formalises the intuitive notion of logical system, including syntax, semantics, and the satisfaction between them. Institutions become a common tool in the study of algebraic specification theory and can be considered its most fundamental mathematical structure. It is already an algebraic specification tradition to have an institution underlying each language or system, in which all language/system constructs and features can be rigorously explained as mathematical entities. This has been first spelt out as a programme with a sample definition of specification language constructs in [2]. Most modern algebraic specification languages follow this tradition, including CASL [3], Maude [4], or CafeOBJ [5]. There is an increasing multitude of logics in use as institutions in algebraic specification and computing science. Some of them, such as first order (in many variants), second order, higher order, Horn, type theoretic, equational, modal (in many variants), infinitary logics, etc., are well known or at least familiar to the ordinary logicians, while others such as behavioural [6–8] or rewriting logics [9] are known and used mostly in computing science. The original goal of institutions is to do as much computing science and model theory as possible, independently of what the underlying logic may be [1]. This paradigm is often called 'institution-independent' computing science or model theory.

The logic programming paradigm [10] in its purely logical form can be described as follows: "Given a universal Horn (finite) presentation (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

E-mail address: razvan.diaconescu@imar.ro.

0020-0190/$ – see front matter
doi:10.1016/j.ipl.2004.01.005

(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 [10], less conventional forms of logic programming extends this to multiple sorts, or even considers first order logic with equality as underlying logic [11–13], 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 [14]. 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 formulae, most of them being part of the 'internal logic' of institutions developed in [15]. The institution-independent concept of substitution is developed for the first time here, being one of the main contributions of this paper.

Our work 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 we also discuss some basic modularisation issues for logic programming.

Other applications of institutions to logic programming include [16], but in a completely different way than our use of institutions.

2. Institutions

Institutions represent a mathematical meta-theory on logics, technically based on category theory, which abstracts the Tarskian concept of truth, and which builds on the idea of the invariance of truth with respect to translation of notation. Hence this work assumes some familiarity with category theory, and generally uses the same notations and terminology as Mac Lane [17], except that composition is denoted by ";" and written in the diagrammatic order. The application of functions (functors) to arguments may be written either normally using parentheses, or else in diagrammatic order without parentheses, or, more rarely, by using sub-scripts or super-scripts. The category of sets is denoted as $\mathbb{S}$et, and the category of categories[1] as $\mathbb{C}$at. The opposite of a category $\mathbb{C}$ is denoted by $\mathbb{C}^{\mathrm{op}}$. The class of objects of a category $\mathbb{C}$ is denoted by $|\mathbb{C}|$; also the set of arrows in $\mathbb{C}$ having the object a as source and the object b as target is denoted as $\mathbb{C}(a, b)$. An object a is *projective* with respect to an arrow $h : b \to c$ when for each arrow $f : a \to c$ there exists an arrow g such that $g; h = f$. Given a category $\mathbb{C}$ and an object $a \in |\mathbb{C}|$, the *comma category* $a/\mathbb{C}$ has arrows $f : a \to b$ as objects and $h \in \mathbb{C}(b, b')$ such that $f; h = f'$ as arrows $h : (f : a \to b) \to (f' : a \to b')$.

Definition 1. An *institution* $(\mathbb{S}\mathrm{ign}, \mathsf{Sen}, \mathsf{Mod}, \models)$ consists of

1. a category $\mathbb{S}$ign, whose objects are called *signatures*,
2. a functor $\mathsf{Sen} : \mathbb{S}\mathrm{ign} \to \mathbb{S}\mathrm{et}$, giving for each signature a set whose elements are called *sentences* over that signature,
3. a functor $\mathsf{Mod} : \mathbb{S}\mathrm{ign}^{\mathrm{op}} \to \mathbb{C}\mathrm{at}$ giving for each signature Σ a category whose objects are called Σ-*models*, and whose arrows are called Σ-(*model*) *homomorphisms*, and
4. a relation $\models_\Sigma \subseteq |\mathsf{Mod}(\Sigma)| \times \mathsf{Sen}(\Sigma)$ for each $\Sigma \in |\mathbb{S}\mathrm{ign}|$, called Σ-*satisfaction*,

such that for each morphism $\varphi : \Sigma \to \Sigma'$ in $\mathbb{S}$ign, the *satisfaction condition*

$$M' \models_{\Sigma'} \mathsf{Sen}(\varphi)(e) \quad \text{iff} \quad \mathsf{Mod}(\varphi)(M') \models_\Sigma e$$

holds for each $M' \in |\mathsf{Mod}(\Sigma')|$ and $e \in \mathsf{Sen}(\Sigma)$. We may denote the reduct functor $\mathsf{Mod}(\varphi)$ by $\_\restriction_\varphi$ and the sentence translation $\mathsf{Sen}(\varphi)$ simply by $\varphi(\_)$. When $M = M'\restriction_\varphi$ we will say that M' is an *expansion of* M *along* φ.

[1] We steer clear of any foundational problem related to the "category of all categories"; several solutions can be found in the literature; see, for example, [17].

Any carefully defined logic or model theory can be presented as an institution.

Example 1. Let **FOL** be the institution of *many sorted first order logic with equality*. Its signatures (S, F, P) consist of a set of sort symbols S, a set F of function symbols, and a set P of relation symbols. Each function or relation symbol comes with a string of argument sorts, called *arity*, and for functions symbols, a result sort. Signature morphisms map the three components in a compatible way. Models M are first order structures interpreting each sort symbol s as a set M_s, each function symbol σ as a function M_σ from the product of the interpretations of the argument sorts to the interpretation of the result sort, and each relation symbol π as a subset M_π of the product of the interpretations of the argument sorts. Notice that each term t of (S, F, P) can be interpreted in any model M as one of its elements, denoted M_t. If $t = \sigma(t_1, \ldots, t_n)$ then M_t is defined as $M_\sigma(M_{t_1}, \ldots, M_{t_n})$. Sentences are the usual first order sentences built from equational and relational atoms by iterative application of logical connectives and quantifiers. Sentence translations rename the sorts, function, and relation symbols. For each signature morphism φ, the reduct $M'|_\varphi$ of a model M' is defined by $(M'|_\varphi)_x = M'_{\varphi(x)}$ for each x sort, function, or relation symbol from the domain signature of φ. The satisfaction of sentences by models is the usual Tarskian satisfaction defined inductively on the structure of the sentences.

Example 2. The institution **EQL** of equational logic can be obtained by eliminating from the institution **FOL** of first order logic with equality the relation symbols and their interpretations, and by allowing only universally quantified equations (either in conditional or unconditional form) as sentences. The signatures of **EQL** are called *algebraic signatures* and the **EQL** models are called *algebras*.

Definition 2. Let Σ be a signature in an institution $(\mathbb{S}\mathsf{ign}, \mathsf{Sen}, \mathsf{Mod}, \models)$. For each set of Σ-sentences E, let $E^* = \{M \in \mathsf{Mod}(\Sigma) \mid M \models_\Sigma e \text{ for each } e \in E\}$.

Two sentences e and e' of the same signature are *semantically equivalent* (denoted as $e \equiv e'$) if they are satisfied by the same class of models, i.e., $\{e\}^* = \{e'\}^*$.

For any signature Σ, the satisfaction relation between Σ-models and Σ-sentences can be extended to a semantic consequence relation between sets of Σ-sentences, i.e., $E \models_\Sigma E'$ when each model satisfying E satisfies E' too.

Definition 3. A pair (Σ, E) is a *presentation* in an institution when Σ is a signature and E is a set of Σ-sentences. A presentation morphism $\varphi : (\Sigma, E) \to (\Sigma', E')$ is any signature morphism $\varphi : \Sigma \to \Sigma'$ such that $E' \models \varphi(E)$.

The work [15] shows how logical connectives, quantifiers, and atomic formulae can be developed internally to any institution. The case of logical connectives is straightforward (and not needed here), hence we concentrate only on the last two.

In the actual institutions, the 'basic' sentences are the simplest sentences matching the structure of the models of the institution, i.e., which are preserved by the model homomorphisms, and they usually constitute the bricks from which the complex sentences are constructed by using logical connectives and quantification. Notice that the satisfaction of basic sentences is a particular case of 'injectivity' satisfaction in the sense of [18].

Definition 4. Given a signature Σ in any institution, a Σ-sentence e is *basic* if there exists a Σ-model M_e such that for each Σ-model M, $M \models_\Sigma e$ if and only if there exists a model homomorphism $M_e \to M$.

Example 3. In the case of first order logic **FOL**, the ground atoms are basic. Recall that a ground atom is either an equality between ground terms or a relation (predicate) with ground terms as arguments.

If we consider a ground equation $(\forall \emptyset) t = t'$ for an algebraic signature (S, F), then let T_F/E be the (quotient) initial (S, F)-algebra satisfying $(\forall \emptyset) t = t'$. In this case E is the congruence generated by the pair (t, t'). Then, an algebra A satisfies $(\forall \emptyset) t = t'$ if and only if there exists a homomorphism $T_F/E \to A$.

If we consider a ground atomic relation $\pi(t_1 \ldots t_n)$ for a first order logic signature (S, F, P), where $t_1, \ldots, t_n$ is a list of F-terms, then we consider the (S, F, P)-model T such that as an algebra, T is the initial term (S, F)-algebra T_F, and which interprets all relation symbols as the empty relation except $T_\pi = \{(t_1, \ldots, t_n)\}$. Then $M \models \pi(t_1 \ldots t_n)$ if and only

if there exists a homomorphism $T \to M$, for each (S, F, P)-model M.

Notice that conjunctions of ground atoms are also basic.

In other actual institutions basic sentences can be easily identified as (possibly conjunctions) of 'atoms' too, see [15] for rewriting logic, partial algebra, or behavioural logics.

Definition 5. Given a signature morphism $\chi : \Sigma \to \Sigma'$, a Σ-sentence $(\forall \chi)\rho$ is *universal χ-quantification* of the Σ'-sentence ρ if and only if for each Σ-model M

$M \models_\Sigma (\forall \chi)\rho$ if and only if

$\big(M' \models_{\Sigma'} \rho$ for all Σ'-models M' with $M'|_\chi = M\big)$.

Existential quantification $(\exists \chi)\rho$ can be defined similarly by replacing 'all' by 'some' in the definition of the universal quantification.

This very abstract and general concept of quantification, introduced first time by [19], for example, in the particular case of classical model theory includes the second order quantification. Notice that this internalisation of the quantification does not use the ordinary concepts of open formulae and valuations (of unbounded variables), but rather considers the "variables" as part of the signature and treats the "valuations" as model expansions along the signature extension defined by the addition of the "variables" to the signature. This is exactly what happens in applications because each valuation of variables into a model can be regarded as an expansion of the model to the signature extended with the variables. Otherwise said, for quantification we need only to mark a part of the signature over which the quantification is done. Although this way of thinking about variables and quantification is well known in conventional mathematical logic [20, 21] it is quite rare in the usual presentations of classical logic.

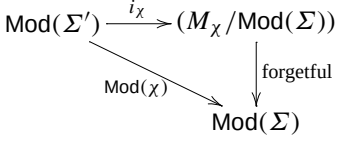
Example 4. Given a signature (S, F, P) in many sorted first order logic **FOL**, the ordinary first order quantification by a set X of variables is the same with the χ-quantification, where $\chi : (S, F, P) \hookrightarrow (S, F \cup X, P)$.

The cases when $\chi : (S, F, P) \hookrightarrow (S', F', P')$ is an arbitrary signature inclusion correspond to the second order quantification by the operations $F' \setminus F$ and predicates (relations) $P' \setminus P$ when $S = S'$, and extends also to sort quantification when $S \subseteq S'$. 'Weak' second order quantification only over *finite* subsets of the models can be obtained by enriching the signatures with (1-ary) symbols denoting finite subsets of models, and consequently the models have to interpret them accordingly.

Quantifications higher than second order can be modelled by Definition 5 provided that the classical concept of first order logic signature is extended in order to accommodate symbols denoting higher order structures.

In the actual institutions, due to its good properties, the restriction of quantification to first order plays an important role. The categorical definition below of signature morphisms corresponding to first order quantification has been introduced in [15]:

Definition 6. 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:



Example 5. In the institution **FOL** of first-order logic each extension of signatures $\chi : (S, F, P) \hookrightarrow (S, F \cup X, P)$ only adding constants X to F is representable by $T_F(X)$, the free (S, F, P)-model over the added constants X.

Similarly, first order variables in other actual institutions correspond to representable signature morphisms [15].

Definition 7. A signature morphism $\chi : \Sigma \to \Sigma'$ is *conservative* if and only if each Σ-model admits at least one expansion along χ.

Example 6. In conventional first order model theory, a signature morphism $\chi : (S, F, P) \to (S', F', P')$ is conservative when φ is injective on the sort, function, and relation symbols and does not add new operations of sorts in S that are 'empty' (i.e., without F-terms). Consequently, if (S, F, P) has only 'non-empty' sorts, then each injective signature morphism $\chi : (S, F, P) \to (S', F', P')$ is conservative.

3. Logic programming over arbitrary institutions

Definition 8. Given a signature Σ in an arbitrary institution, a *Σ-query* is any existentially quantified Σ-sentence $(\exists\chi)\rho$ such that χ is representable and ρ is a basic sentence.

Herbrand theorem reduces the problem of checking the satisfaction of a query by a presentation from all possible models to the initial model only. It relies on the existence of an initial model for the presentation (program), known under the name of 'Herbrand universe' [10] by the classical logic programming community. This initiality requirement determines the restriction of logic programming to universal Horn theories, a well known fact in classical logic, and which has been proved for the first time in an institution-independent setting in [22,19]. This is consistent to the fact that in the actual institutions, the universal Horn sentences are the most complex sentences supporting algorithms for automatic execution of logic programming.

Theorem 1. *In an arbitrary institution consider a presentation (Σ, E) which has an initial model $0_{\Sigma,E}$. Then for each query $(\exists\chi)\rho$*

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

Proof. Let $\chi : \Sigma \to \Sigma'$. Assume that $0_{\Sigma,E} \models (\exists\chi)\rho$ and consider a Σ-model M such that $M \models E$. There exists an expansion N of $0_{\Sigma,E}$ such that $N \models \rho$. Because χ is representable let $f = i_\chi(N) : M_\chi \to 0_{\Sigma,E}$. Let $g : 0_{\Sigma,E} \to M$ be the unique Σ-model homomorphism by the fact that $M \models E$. Let $M' = i_\chi^{-1}(f; g)$.

Then M' is an expansion of M along χ and $i_\chi^{-1}(g) : N \to M'$. Because ρ is basic, there exists a model homomorphism $M_\rho \to N$, and by composing with $i_\chi^{-1}(g)$ there exists a model homomorphism $M_\rho \to M'$, which means that $M' \models \rho$. This shows that $M \models (\exists\chi)\rho$. □

Each expansion M' of $0_{\Sigma,E}$ along χ such that $M' \models \rho$ is called a *solution* for the query $(\exists\chi)\rho$. The importance of Theorem 1 is that is reduces the search space for solution of queries from *all* models to only one model.

Various well known actual Herbrand theorems can be obtained as special cases of this result by varying the underlying institution. For the case of pure Prolog [10] we take unsorted first order logic *without equality*, for Eqlog [12] we take [13] many sorted first order logic with equality, for behavioural logic programming [14] we take hidden algebra, for constraint logic programming we take constraint logic [23]. Herbrand theorem for category-based equational logic can be obtained by considering the institution of category-based equational logic [24,25]. Denotational foundations for new combinations between logic programming and other computing paradigms can be obtained by considering rewriting logic [9], membership logic [26], multialgebra [27], etc.

Substitutions and solution forms

In the following we will develop the Herbrand theorem into another version which reduces the validation of (existential) queries to universally quantified 'atoms'. In the actual situations this opens the door for developing refutation-based algorithms for logic programming execution.

Given a **FOL** signature (S, F, P) and two sets of variables X and Y, a substitution ψ from X to Y, denoted $\psi : X \to Y$, consists of a mapping of the variables X with F-terms over Y (i.e., $\psi : X \to T_F(Y)$ as function).

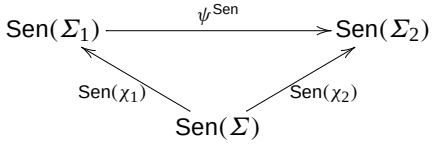
For any substitution $\psi : X \to Y$ (i.e., function $\psi : X \to T_F(Y)$) any interpretation M_2 of variables Y into an (S, F, P)-model M determines an interpretation $\psi^{\mathsf{Mod}}(M_2)$ of the variables X into M by $\psi^{\mathsf{Mod}}(M_2)_x = (M_2)_{\psi(x)}$ for each $x \in X$. On the other hand, each $(S, F \cup X, P)$-sentence ρ can be *instantiated* by ψ to an $(S, F \cup Y, P)$-sentence $\psi^{\mathsf{Sen}}(\rho)$ by replacing each $x \in X$ with the term $\psi(x)$. This phenomenon has the same flavour with sentence and model

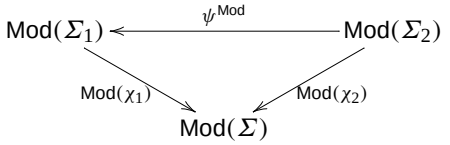
translations along signature morphisms, and we can notice easily that

$$M_2 \models \psi^{\mathsf{Sen}}(\rho) \quad \text{if and only if} \quad \psi^{\mathsf{Mod}}(M_2) \models \rho.$$

This remark is the key to the institution-independent concept of substitution.

Definition 9. In any institution, a *substitution* $\psi : \chi_1 \to \chi_2$ from $\chi_1 : \Sigma \to \Sigma_1$ to $\chi_2 : \Sigma \to \Sigma_2$ signature morphisms is a pair $(\psi^{\mathsf{Sen}}, \psi^{\mathsf{Mod}})$ consisting of a sentence translation $\psi^{\mathsf{Sen}} : \mathsf{Sen}(\Sigma_1) \to \mathsf{Sen}(\Sigma_2)$ and a model translation $\psi^{\mathsf{Mod}} : \mathsf{Mod}(\Sigma_2) \to \mathsf{Mod}(\Sigma_1)$ both of them "preserving" Σ, i.e., the diagrams below commute:





and such that the following satisfaction condition holds:

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

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

Our institution-independent concept of substitution accommodates also second order substitutions (which replace function symbols with terms) and even higher order substitutions in the case of institutions of higher-order logics.

Remark 1. Assume χ_1 and χ_2 are representable. For any substitution $\psi : \chi_1 \to \chi_2$, there exists a Σ-model homomorphism $M_\psi : M_{\chi_1} \to M_{\chi_2}$ such that, for any Σ_2-model M_2,

$$\psi^{\mathsf{Mod}}(M_2) = i_{\chi_1}^{-1}\big(M_\psi; i_{\chi_2}(M_2)\big).$$

In the actual example of first-order substitutions $\psi : X \to Y$ in first-order logic, $M_\psi : T_F(X) \to T_F(Y)$ is the unique extension of $\psi : X \to T_F(Y)$ to an (S, F, P)-model homomorphism. Moreover, *each* first-order substitution is thus determined by a model homomorphism between representations.

Definition 10. An institution has *representable substitutions* when for all representable signature morphisms $\chi_1 : \Sigma \to \Sigma_1$ and $\chi_2 : \Sigma \to \Sigma_2$ each model homomorphism $h : M_{\chi_1} \to M_{\chi_2}$ determines a substitution $\psi_h : \chi_1 \to \chi_2$ such that $\psi_h^{\mathsf{Mod}}(M_2) = i_{\chi_1}^{-1}(h; i_{\chi_2}(M_2))$ for each Σ_2-model M_2.

By the satisfaction condition for substitutions, we notice easily that ψ_h is unique modulo *substitution equivalence*, i.e., for any substitutions ψ_h and ψ'_h determined by h, $\psi_h^{\mathsf{Mod}} = \psi'^{\mathsf{Mod}}_h$ and $\psi_h^{\mathsf{Sen}}(\rho) \equiv \psi'^{\mathsf{Sen}}_h(\rho)$ for each Σ_1-sentence ρ.

The Herbrand theorem below can be interpreted in the various institutions which constitute application domains for Herbrand Theorem 1 and which have been discussed above.

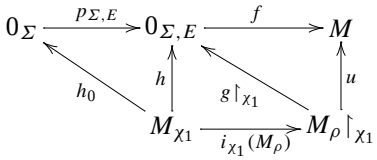
Theorem 2. *Consider an institution with representable substitutions such that*

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

Then for each presentation (Σ, E) having an initial model, and for any query $(\exists\chi_1)\rho$

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

Proof. Assume $E \models (\exists\chi_1)\rho$ and let $\chi_1 : \Sigma \to \Sigma_1$. By Herbrand Theorem 1, we have that $0_{\Sigma,E} \models (\exists\chi_1)\rho$. Let M_1 be the expansion of $0_{\Sigma,E}$ along χ_1 such that $M_1 \models \rho$ and let $h = i_{\chi_1}(M_1) : M_{\chi_1} \to 0_{\Sigma,E}$. By the projectivity property of M_{χ_1}, there exists $h_0 : M_{\chi_1} \to 0_\Sigma$ such that $h_0; p_{\Sigma,E} = h$. We show that $E \models (\forall 1_\Sigma)\psi_{h_0}^{\mathsf{Sen}}(\rho)$ (notice also that 1_Σ is trivially conservative), where ψ_{h_0} is a substitution $\chi_1 \to 1_\Sigma$ determined by h_0.



Let M be a model such that $M \models E$ and let $f : 0_{\Sigma,E} \to M$ be the unique (Σ, E)-model homomorphism. Because $i_{\chi_1}^{-1}(h) = M_1 \models \rho$ there exists $g : M_\rho \to i_{\chi_1}^{-1}(h) = M_1$ and we have that $i_{\chi_1}(M_\rho); g\restriction_{\chi_1} = h$. Let $u = g\restriction_{\chi_1}; f$. We have $h; f = i_{\chi_1}(M_\rho); u$, hence there exists a model homomorphism $M_\rho \to i_{\chi_1}^{-1}(h; f)$. It therefore follows that $i_{\chi_1}^{-1}(h; f) \models \rho$, which means that $i_{\chi_1}^{-1}(h_0; p_{\Sigma,E}; f) \models \rho$ which means that $i_{\chi_1}^{-1}(h_0; i_{1_\Sigma}(M)) \models \rho$. By the definition of $\psi_{h_0}^{\mathsf{Mod}}(M)$ this implies that $\psi_{h_0}^{\mathsf{Mod}}(M) \models \rho$. By the satisfaction condition for substitutions we therefore deduce that $M \models \psi_{h_0}^{\mathsf{Sen}}(\rho)$.

For the converse, we assume there exists a substitution $\psi : \chi_1 \to \chi_2$ such that $E \models (\forall \chi_2)\psi^{\mathsf{Sen}}(\rho)$. Because χ_2 is conservative we can find a model homomorphism $u : M_{\chi_2} \to 0_\Sigma$. We show that $M_1 = i_{\chi_1}^{-1}(M_\psi; u; p_{\Sigma,E})$ is an expansion of $0_{\Sigma,E}$ along χ_1 such that $M_1 \models \rho$.

Let $M_2 = i_{\chi_2}^{-1}(u; p_{\Sigma,E})$. Because $E \models (\forall \chi_2)\psi^{\mathsf{Sen}}(\rho)$ and $M_2\restriction_{\chi_2} = 0_{\Sigma,E}$, we have that $M_2 \models \psi^{\mathsf{Sen}}(\rho)$. Notice that $M_1 = i_{\chi_1}^{-1}(M_\psi; i_{\chi_2}(M_2)) = \psi^{\mathsf{Mod}}(M_2)$. By the satisfaction condition for the substitution ψ this implies that $M_1 \models \rho$.

Therefore $0_{\Sigma,E} \models (\exists \chi_1)\rho$. By Herbrand Theorem 1 we now have that $E \models (\exists \chi_1)\rho$. □

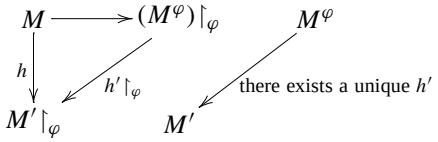
The substitutions ψ of Herbrand Theorem 2 are called *solution forms*. The proof of the 'only if' part of this theorem shows that each solution for a query is an instance of a solution form for the query, while the proof of the 'if' part shows that each instance of any solution form for a query to the initial model gives a solution for the query.

Notice that the supplementary conditions of Herbrand Theorem 2 are very mild in the actual examples. For example, in first order logic **FOL**, the second condition is easily satisfied in the presence of an axiom of choice because the representation models of the representable signature morphisms are free over the set of variables and the 'quotient' homomorphisms $0_\Sigma \to 0_{\Sigma,E}$ are surjective.

4. Modularisation

Now we introduce a couple of concepts playing an important role in th study of structured software specifications or programming modules.

Definition 11. A signature morphism $\varphi : \Sigma \to \Sigma'$ is *liberal* if and only if the reduct functor $\_\restriction_\varphi : \mathsf{Mod}(\Sigma') \to \mathsf{Mod}(\Sigma)$ has a left-adjoint, denoted $(\_)^\varphi$.



In the actual institutions the liberality of signature morphisms holds in general easily.

Exactness properties for institutions formalise the possibility of amalgamating models of different signatures when they are consistent on some kind of 'intersection' of the signatures (formalised as a pushout square):

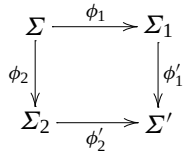
Definition 12. An institution is *exact* if and only if its model functor $\mathsf{Mod} : \mathbb{S}\mathrm{ign}^{\mathrm{op}} \to \mathbb{C}\mathrm{at}$ preserves finite limits. The institution is *semi-exact* if and only if Mod preserves pullbacks.

Semi-exactness is everywhere.[2] Virtually all institutions formalising conventional or non-conventional logics are at least semi-exact. In general the institutions of many-sorted logics are exact, while those of unsorted (or one-sorted) logics are only semi-exact [28]. However, in applications the important amalgamation property is the semi-exactness rather than the full exactness.

The following amalgamation property is a direct consequence of semi-exactness:

Definition 13. The commuting square of signature morphisms

[2] At least in the *weak* form requiring only the existence but not the uniqueness.

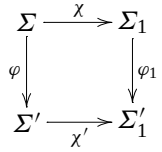


is an *amalgamation square* if and only if for each Σ_1-model M_1 and a Σ_2-model M_2 such that $M_1{\restriction}_{\phi_1} = M_2{\restriction}_{\phi_2}$, there exists an unique Σ'-model M' such that $M'{\restriction}_{\phi'_1} = M_1$ and $M'{\restriction}_{\phi'_2} = M_2$. The model M' is called the *amalgamation* of M_1 and M_2.

Corollary 1. *In a semi-exact institution each pushout square of signature morphisms is an amalgamation square.*

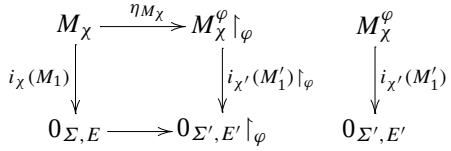
Logic programming modules can be abstracted to presentations, and module imports to presentation morphisms. Various other module compositions can be expressed as co-limits of presentation morphisms [28].

If we assume that the institution is semi-exact and all signature morphisms are liberal, for any signature morphism $\varphi : \Sigma \to \Sigma'$, we can translate any Σ-query $(\exists\chi)\rho$ to the Σ'-query $(\exists\chi')\varphi_1(\rho)$ where



is a pushout of signatures. Then it is rather easy to show that $\varphi_1(\rho)$ is basic (by the fact that liberal signature morphisms translate basic sentences to basic sentences) and that χ' is representable (by defining $M_{\chi'} = M_\chi^\varphi$), hence $(\exists\chi')\varphi_1(\rho)$ is indeed a query.

On the other hand, for each presentation morphism $\varphi : (\Sigma, E) \to (\Sigma', E')$, each expansion M_1 of the initial model $0_{\Sigma,E}$ is translated to the expansion M'_1 of the initial model $0_{\Sigma',E'}$ such that the diagram below commutes:



where η is the unit of the adjunction between the categories of models of Σ and Σ'.

Proposition 1. *If M_1 is a solution for the query $(\exists\chi)\rho$, then M'_1 is a solution for the query $(\exists\chi')\varphi_1(\rho)$.*

Moreover, if the presentation morphism φ is conservative for initiality, *i.e., $0_{\Sigma',E'}{\restriction}_\varphi = 0_{\Sigma,E}$, then for each solution N for $(\exists\chi')\varphi_1(\rho)$, there exists a solution M_1 for $(\exists\chi)\rho$ such that $M'_1 = N$.*

Proof. Let us assume M_1 is a solution for $(\exists\chi)\rho$. Then by the definition of M'_1 we have that there exists a model homomorphism $M_1 \to M'_1{\restriction}_{\varphi_1}$. Because $M_1 \models \rho$ we have that there exists a model homomorphism $M_\rho \to M_1$, which implies that there exists a model homomorphism $M_\rho \to M'_1{\restriction}_{\varphi_1}$. By the universal property of the adjunction corresponding to φ_1 we have that there exists a model homomorphism $M_{\varphi_1(\rho)} = M_\rho^{\varphi_1} \to M'_1$, which shows that $M'_1 \models \varphi_1(\rho)$.

For the second part, we assume that N is a solution for $(\exists\chi')\varphi_1(\rho)$. We define $M_1 = i^{-1}(\eta_{M_\chi}; i_{\chi'}(N){\restriction}_\varphi)$. □

The actual meaning of the first part of this proposition is that each module import 'preserves' the solution of queries, while the meaning of the second part is that the module imports which 'protect' the initial model of the imported module also 'protect' the solution of queries.

5. Conclusions and future research

We have developed denotational foundations for logic programming independently of the details of the underlying institution by employing internal institution-independent concepts of variable, substitution, quantification, and query.

At this level of generality we have proved the Herbrand theorem in two versions, the second one reducing the existence of solutions for queries to existence of solution forms. In the actual institutions this opens the door for execution of logic programming by refutation algorithms such as resolution, paramodulation, etc.

We have also analysed modularisation issues such as preservation and protection of solutions for queries along presentation morphisms.

The importance of this work resides in the fact that it sets denotational foundations for a uniform development of logic programming over a large variety

of computing science logics. Moreover, by employing concepts of mappings between institutions [29] we plan to develop a theory about borrowing of various aspects of the logic programming paradigm between logic programming languages or systems defined over different logics. We also plan to extend this institution-independent study from denotational to operational semantics of logic programming.

References

[1] J. Goguen, R. Burstall, Institutions: Abstract model theory for specification and programming, J. ACM 39 (1) (1992) 95–146.

[2] D. Sannella, A. Tarlecki, Specifications in an arbitrary institution, Inform. Control 76 (1988) 165–210; earlier version in: Proceedings, International Symposium on the Semantics of Data Types, in: Lecture Notes in Computer Science, vol. 173, Springer, Berlin, 1985.

[3] E. Astesiano, M. Bidoit, H. Kirchner, B. Krieg-Brückner, P. Mosses, D. Sannella, A. Tarlecki, CASL: The common algebraic specification language, Theoret. Comput. Sci. 286 (2) (2002) 153–196.

[4] J. Meseguer, A logical theory of concurrent objects and its realization in the Maude language, in: G. Agha, P. Wegner, A. Yonezawa (Eds.), Research Directions in Concurrent Object-Oriented Programming, MIT Press, Cambridge, MA, 1993.

[5] R. Diaconescu, K. Futatsugi, Logical foundations of CafeOBJ, Theoret. Comput. Sci. 285 (2002) 289–318.

[6] R. Diaconescu, K. Futatsugi, Behavioural coherence in object-oriented algebraic specification, Universal Comput. Sci. 6 (1) (2000) 74–96; first version appeared as JAIST Technical Report IS-RR-98-0017F, June 1998.

[7] J. Goguen, G. Roşu, Hiding more of hidden algebra, in: J.M. Wing, J. Woodcock, J. Davies (Eds.), FM '99—Formal Methods, in: Lecture Notes in Computer Science, vol. 1709, Springer, Berlin, 1999, pp. 1704–1719.

[8] R. Hennicker, M. Bidoit, Observational logic, in: A.M. Haeberer (Ed.), Algebraic Methodology and Software Technology, in: Lecture Notes in Computer Science, vol. 1584, Springer, Berlin, 1999, pp. 263–277.

[9] J. Meseguer, Conditional rewriting logic as a unified model of concurrency, Theoret. Comput. Sci. 96 (1) (1992) 73–155.

[10] J. Lloyd, Foundations of Logic Programming, second extended ed., Springer, Berlin, 1988.

[11] M.J. O'Donnell, Computing in systems described by equations, in: Lecture Notes in Computer Science, vol. 58, Springer, Berlin, 1977.

[12] J. Goguen, J. Meseguer, Eqlog: Equality, types, and generic modules for logic programming, in: D. DeGroot, G. Lindstrom (Eds.), Logic Programming: Functions, Relations and Equations, Prentice-Hall, Englewood Cliffs, NJ, 1986, pp. 295–363; an earlier version appears in: J. Logic Programming 1 (2) (1984) 179–210.

[13] J. Goguen, J. Meseguer, Models and equality for logical programming, in: H. Ehrig, G. Levi, R. Kowalski, U. Montanari (Eds.), Proceedings, 1987 TAPSOFT, in: Lecture Notes in Computer Science, vol. 250, Springer, Berlin, 1987, pp. 1–22.

[14] J. Goguen, G. Malcolm, T. Kemp, A hidden Herbrand theorem: Combining the object, logic and functional paradigms, J. Logic Algebraic Programming 51 (1) (2002) 1–41.

[15] R. Diaconescu, Institution-independent ultraproducts, Fund. Inform. 55 (3-4) (2003) 321–348.

[16] F. Orejas, E. Pino, H. Ehrig, Institutions for logic programming, Theoret. Comput. Sci. 173 (1997) 485–511.

[17] S. MacLane, Categories for the Working Mathematician, second ed., Springer, Berlin, 1998.

[18] H. Andréka, I. Németi, A general axiomatizability theorem formulated in terms of cone-injective subcategories, in: B. Csakany, E. Fried, E. Schmidt (Eds.), Universal Algebra, in: Colloquia Mathematics Societas János Bolyai, vol. 29, North-Holland, Amsterdam, 1981, pp. 13–35.

[19] A. Tarlecki, Quasi-varieties in abstract algebraic institutions, J. Comput. System Sci. 33 (3) (1986) 333–360; original version: University of Edinburgh, Report CSR-173-84.

[20] J. Shoenfield, Mathematical Logic, Addison-Wesley, Reading, MA, 1967.

[21] W. Hodges, Model Theory, Cambridge Univ. Press, Cambridge, 1993.

[22] A. Tarlecki, On the existence of free models in abstract algebraic institutions, Theoret. Comput. Sci. 37 (1986) 269–304; preliminary version: University of Edinburgh, Computer Science Department, Report CSR-165-84, 1984.

[23] R. Diaconescu, Category-based constraint logic, Math. Struct. Comput. Sci. 10 (3) (2000) 373–407.

[24] R. Diaconescu, Completeness of category-based equational deduction, Math. Struct. Comput. Sci. 5 (1) (1995) 9–41.

[25] R. Diaconescu, Category-based semantics for equational and constraint logic programming, Ph.D. thesis, University of Oxford (1994).

[26] J. Meseguer, Membership algebra as a logical framework for equational specification, in: F. Parisi-Pressice (Ed.), Proc. WADT '97, in: Lecture Notes in Computer Science, vol. 1376, Springer, Berlin, 1998, pp. 18–61.

[27] Y. Lamo, The institution of multialgebras—a general framework for algebraic software development, Ph.D. thesis, University of Bergen, 2002.

[28] R. Diaconescu, J. Goguen, P. Stefaneas, Logical support for modularisation, in: G. Huet, G. Plotkin (Eds.), Logical Environments, Cambridge Univ. Press, Cambridge, 1993, pp. 83–130.

[29] J. Goguen, G. Roşu, Institution morphisms, Formal Aspects Comput. 13 (2002) 274–307.