On infinitary equational hybrid logic[1]

Manuel A. Martins

CIDMA-Center for R&D Mathematics and Applications
Mathematics Dep., U. Aveiro
Portugal
martins@ua.pt

Alexandre Madeira

HASLab INESC TEC
QuantaLab
Informatics Dep., U. Minho
Portugal
madeira@di.uminho.pt

Luís S. Barbosa

HASLab INESC TEC
QuantaLab
Informatics Dep., U. Minho
Portugal
lsb@di.uminho.pt

Abstract: Reconfigurable software systems behave differently in different modes of operation and commute between them along their lifetime. Such different behaviours can be modelled by a transition system, to express the overall system's dynamics, but with structured states to capture local properties. We take this path in this paper by endowing states in standard Kripke frames with algebras, each of them modelling a local configuration. An equational hybrid logic, with infinitary formulas, is proposed to express a broad range of properties of those structures, including liveness requirements. The paper develops a number of results on its semantics, including suitable notions of simulation and bisimulation.

Keywords: Infinitary Logic, Equational Hybrid logic, Bisimulation.

[1]This work was supported by ERDF European Regional Development Fund, through the COMPETE Programme, and by National Funds through FCT - Portuguese Foundation for Science and Technology - within projects POCI-01-0145-FEDER-016692 and UID/MAT/04106/2013. A. Madeira is also supported by the FCT individual grant SFRH/BPD/103004/2014.

Beziau, J-Y.; Costa-Leite, A; D'Ottaviano, I. M. L. (orgs.). Aftermath of the Logical Paradise, Coleção CLE, v.81, pp. 173–202, 2018.

1. Introduction

Motivation. Published in 1976, Niklaus Wirth's most influential book *Algorithms + Data Structures = Programs* [40] was the first to systematically draw the community attention to the fundamental interconnection between what one would call now *behaviour* and *data* in software construction. The message is more relevant than ever: systems whose functionality changes along their own computation, in response to varying context conditions, are ubiquitous in modern societies. We classify such systems as *reconfigurable* to emphasise that their behaviour commutes through a set of different run-time modes of operation along their lifetime.

At present such is more the norm than the exception in software systems whose components are frequently reconfigured. A typical, everyday example is offered by cloud based applications that elastically react to client demands. Another example is a modern car in which hundreds of electronic control units must operate in different modes depending on the current situation – such as driving on a highway or finding a parking spot. Switching between these modes is an intuitive example of a dynamic reconfiguration.

Classically, there are two main paradigms to formally capture requirements for this sort of software: one emphasises *behaviour* and its evolution; the other focus on *data* and their transformations. In the former, systems are specified through (some variant of) *state-machines* and their evolution is expressed in terms of event occurrences and their impact in internal state configurations. In the latter, data-oriented approach, the system's functionality is given in terms of input-output relations modelling operations on *data*. A specification is presented as a theory in a suitable logic, expressed over a signature, which captures its syntactic interface. Its semantics is a class of concrete algebras acting as models for the specified theory [20].

The authors' recent work [28] aims at putting together these

Beziau, J-Y.; Costa-Leite, A; D'Ottaviano, I. M. L. (orgs.). Aftermath of the Logical Paradise, Coleção CLE, v.81, pp. 173–202, 2018.

two approaches to pave the way to a specification method for reconfigurable systems. Clearly, the dynamics of reconfiguration of a software system can be described by some sort of transition system, whose states represent configurations and transitions are triggered by whatever conditions enforce the move from a configuration to another. However, one needs also to capture the specific, *local* requirements which characterise each configuration and distinguish one from the others. Formally, such different behaviours can be modelled by imposing additional structure upon states in the transition system which expresses the overall dynamics. For example, starting from a classical state-machine specification, each state can be equipped with an algebra (over the system's interface) of the corresponding functionality. Technically, specifications become *structured* state-machines, states denoting *algebras*, rather than *sets*. This method, introduced in [28], raises a number of technical issues which this paper intends to address.

First of all, a specification of a reconfigurable system should be able to make assertions both about the transition dynamics and, locally, about each particular configuration. This entails the need for an expressive logic to deal with transitional behaviour and data specification. Clearly, this should be a modal language with the ability to refer to individual states, each of which stands for a local configuration. *Hybrid* logic [23, 8, 13] is thus the obvious choice.

In general, hybrid logic adds to a modal language the ability to name, or to explicitly refer to specific states of the underlying Kripke structure. This is done through the introduction of propositional symbols of a new sort, called *nominals*, each of which is true at exactly one possible state. The sentences are then enriched in two directions. On the one hand, nominals are used as simple sentences holding exclusively in the state they name. On the other hand, explicit reference to states is provided through a local satisfaction operator. One may therefore specify (local) properties of specific configurations in the system or even to assert the

Beziau, J-Y.; Costa-Leite, A; D'Ottaviano, I. M. L. (orgs.). Aftermath of the Logical Paradise, Coleção CLE, v.81, pp. 173–202, 2018.

equality between two particular configurations, something which is beyond what can be said in a modal language. Modalities, however, capture state transitions, providing a way to specify the *global* dynamics of reconfigurability.

Historically, hybrid logic was introduced by A. Prior in his book [35]. However, its seminal ideas emerged by the end of the fifties, in a discussion of C.A. Meredith [8]. The theme was later revisited, in the school of Sofia, by S. Passy and T. Tinchev [34]. It achieved global interest within the modal logic community in the nineties, with contributions by P. Blackburn, C. Areces, B. ten Cate, T. Braüner, T. Bolander, among many others (see, e.g., [3, 15, 13, 11]). This lifted the *status* of hybrid logic to an independent branch of modern logic. For an historical account we suggest [8, 13], as well as [9] for a comparison with the original perspective of A. Prior.

For the data part, on the other hand, *equational* logic is widely accepted as a solid, mature specification language. Actually, despite their simplicity, equations are enough to characterise all computable data structures [7] and to describe the semantics of programming languages [24]. Moreover, models for equational logic are (universal) algebras, well known and semantically rich structures. Finally, the equational calculus is complete, and rewriting algorithms provide effective tool support for equational reasoning (as in [16] or [20]).

On the other hand, higher expressive power can be achieved by incorporating in the syntax of the logic infinitary formation rules, namely to generate formulas with (possible) infinite disjunctions and conjunctions. The move to an *infinitary* language [22, 37, 38] provides a suitable way to specify both liveness properties and fairness assumptions most relevant in the presence of concurrency and non determinism inherent to the kind of systems we want to capture. Infinitary formulas also express properties of (possibly infinite) data structures in a natural way: for exam-

Beziau, J-Y.; Costa-Leite, A; D'Ottaviano, I. M. L. (orgs.). Aftermath of the Logical Paradise, Coleção CLE, v.81, pp. 173–202, 2018.

ple, $\bigwedge\{top(pop^n(s)) \approx top(pop^n(s')) : n \geq 0\}$ captures behavioural equivalence of unbounded stacks s and s'.

In the sequel we consider languages with (possible) infinite disjunctions and conjunctions over a countable set of variables. As an extension to first-order logic a family of such languages was first introduced by J. Barwise [5] as a tool for exploring infinite structures. Since then they have found a number of uses in theoretical Computer Science. Typical examples range from the specification of infinite data structures [18, 39] and semantics [27] to graph theory [17] and data base query languages [26, 1].

Contributions. The paper contributions are, thus, twofold. First an *infinitary equational hybrid* logic is introduced and its semantics given in terms of Kripke frames whose states are endowed with algebras. Then notions of *simulation* and *bisimulation* [33, 36] between these structures are characterised. A number of preservation results studied for the hybrid propositional case are extended to this richer setting.

Paper structure. The infinitary equational hybrid logic proposed as a *lingua franca* for specifying reconfigurable systems is introduced and briefly illustrated in section 3. Section 4 characterises simulation and bisimulation for the underlying structures and proves the analog to a modal equivalence result for this logic. Finally, section 5 concludes and comments on future work.

2. Preliminares

This section contains a brief summary of concepts and notations for both equational and hybrid propositional logic. For a detailed exposition, the reader is referred to any standard text on each subject, for example [14] and [8, 13], respectively.

2.1 Universal algebra

A *signature* Σ is a family $(\Sigma_n)_{n\in\mathbb{N}}$, where Σ_n is a set of operation symbols of arity n. Given a signature Σ, a Σ-*algebra* A is a nonempty set $|A|$ together with a function $A_f : |A|\times\cdots\times|A| \to |A|$, for each $f \in \Sigma_n$. An homomorphism between two algebras A and A' consists of a map $h : |A| \to |A'|$ such that, for any $f \in \Sigma_n$ and any $a_1,\ldots,a_n \in |A|$, $h(A_f(a_1,\ldots,a_n)) = A'_f(h(a_1),\ldots,h(a_n))$.

For a set X of *variables*, the set of Σ-*terms*, $T(\Sigma,X)$, is the smallest set such that $x \in T(\Sigma,X)$, for each $x \in X$, and $f(t_1,...,t_n) \in T(\Sigma,X)$, for any $f \in \Sigma_n$, and all terms $t_i \in T(\Sigma,X)$, $i = 1,...,n$.

For a given signature Σ and set of variables X, an *equation* is an expression $t \approx t'$, for $t,t' \in T(\Sigma,X)$. We denote the set of equations w.r.t. Σ and X by $\mathrm{Eq}_\Sigma(\mathrm{X})$. An equation $t \approx t'$ is satisfied by an algebra A, in symbols $A \models t \approx t'$, if for any assignment $\sigma : X \to |A|$ we have $\sigma^\#(t) = \sigma^\#(t')$, where $\sigma^\#$ is the unique homomorphism extension of $\sigma : X \to |A|$ to $T(\Sigma,X)$.

Some classes of algebras play an important role in universal algebra. The class K of models of a set of equations E is called an *equational class* (and similarly E is known as an *equational axiomatisation of* K). A class of algebras closed under subalgebras, homomorphisms and products is called a *variety* [14]. The smallest variety containing the class of Σ-algebras K is denoted by $\mathcal{V}(K)$ and is called the *variety generated by* K. When K has a single member A we simply write $\mathcal{V}(A)$. A fundamental result states that, for any class K of Σ-algebras, K and $\mathcal{V}(K)$ satisfy the same equations. This can be proved by using the free algebra over a set of variables.

Definition 2.1 *Let K be a class of Σ-algebras. Given a nonempty set X of variables,* the congruence on $T(\Sigma,X)$ generated by X over K *is*

$$\theta_K = \bigcap \Phi_K(X),$$

where $\Phi_K(X)$ is the set of all congruences ϕ on $T(\Sigma,X)$ such that

Beziau, J-Y.; Costa-Leite, A; D'Ottaviano, I. M. L. (orgs.). Aftermath of the Logical Paradise, Coleção CLE, v.81, pp. 173–202, 2018.

$T(\Sigma, X)/\phi$ *is isomorphic to a subalgebra in* K. *The* K-free algebra over X, $F_K(X)$, *is the quotient algebra* $T(\Sigma, X)/\theta_K$.

We have that

Theorem 2.2 ([14, Theorem 10.12]) *If* K *is a variety, then* $F_K(X) \in K$.

As consequence, for a variety K, we have that the variety generated by $F_K(X)$ is a subset of K. Moreover, in case X is infinite it is exactly K.

Theorem 2.3 ([14]) *If* K *is a variety and* X *is infinite, then* $\mathcal{V}(F_K(X)) = K$.

This theorem is a corollary of the fact that a variety is an equational class. It is straightforward to show that an equational class is a variety. The class of all groups, the class of all rings and the class of all Boolean algebras are examples of equational classes and thus varieties. That every variety is also an equational class is the main part of the following well-known result:

Theorem 2.4 (Birkhoff [14, Theorem 11.9]) *A class of* Σ-*algebras* K *is a variety if and only if* K *is an equational class.*

The theorem, first proven by Birkhoff in the thirties, is one of the many results that characterise syntactic classes of formulas in terms of the closure of their classes of models under certain algebraic constructions. As a corollary any two Σ-algebras which satisfy exactly the same equations generate the same variety.

2.2 Hybrid logic

As briefly mentioned in the Introduction, the qualifier *hybrid* [8, 13] applies to extensions of a modal languages with symbols, called *nominals*, which explicitly refer to individual states in the underlying Kripke frame. A *hybrid signature* is a triple $\Xi = \langle \mathrm{Var}, \mathrm{Nom}, \Lambda \rangle$, where Var and Nom are two countable sets of symbols, of *propositional variables* and *nominals*, respectively, and Λ is a finite set of *modal symbols*. The set of *hybrid formulas over* Ξ

Beziau, J-Y.; Costa-Leite, A; D'Ottaviano, I. M. L. (orgs.). Aftermath of the Logical Paradise, Coleção CLE, v.81, pp. 173–202, 2018.

extends the corresponding modal language with nominals (formula i, for $i \in \text{Nom}$, holding exactly in the state named by i) and formulas of the form $@_i\phi$, for i a nominal and ϕ a formula, asserting that ϕ holds at the state named by i.

If modal logics have been successfully used for specifying reactive systems, the hybrid component adds the possibility to refer to individual states and reasoning about the system's local behaviour at each of them. Typical formulas express equality between states named by i and j ($@_i j$) or the accessibility of the latter from the former through a modality λ ($@_i\langle\lambda\rangle j$). Moreover, hybrid logic is strictly more expressive than its modal fragment. For example, irreflexivity ($@_i\neg\langle\lambda\rangle i$), asymmetry ($@_i\neg\langle\lambda\rangle\langle\lambda\rangle i$), antisymmetry ($i \to [\lambda](\langle\lambda\rangle i \to i)$), or trichotomy ($@_j\langle\lambda\rangle i \vee @_j i \vee @_i\langle\lambda\rangle j$), are properties of the underlying transition structure which are simply not definable in standard modal logic. Note that, however, for the propositional case, the satisfiability problem still is decidable [4].

Hybrid logic, in its multiple flavours, seems suitable to capture complex software requirements, namely of reconfigurable systems as discussed below. The existence of encodings (*i.e.*, conservative comorphisms) to first and second order logics allows the use of computer-supported provers, such as Hets [31], in assisted software verification (see [32]).

3. Infinitary equational hybrid logic

3.1 An infinitary equational hybrid logic

Let us now introduce the specification logic motivated in Section 1. Not only the choice for infinitary formulas [25] and equations, but also other features were motivated from the specification practice. Such is the case of our use of functions, which are a standard tool in algebraic specification. Similarly, we do not consider any rigid component in the logic, allowing ample freedom in specifica-

Beziau, J-Y.; Costa-Leite, A; D'Ottaviano, I. M. L. (orgs.). Aftermath of the Logical Paradise, Coleção CLE, v.81, pp. 173–202, 2018.

tions.

An *equational hybrid similarity type* τ is a triple $\langle \Sigma, \Lambda, \mathrm{Nom} \rangle$ where Σ is an algebraic signature and Λ and Nom are, as above, the sets of modalities and nominals. Assume a countable infinite, fixed set X of variables. The set of Σ-terms over X is defined in the usual way. We define equational hybrid language $\mathrm{Fm}_H(\tau, X)$, in which $\lambda \in \Lambda$ labels a modal box operator and each nominal $i \in \mathrm{Nom}$ is used to construct another operator $@_i$. The set $\mathrm{Fm}(\tau, X)$ of modal infinitary equational formulas is given recursively as follows

1. if t, t' are Σ-terms then $t \approx t'$ is a formula;
2. if φ is a formula, $\lambda \in \Lambda$, then $\neg\varphi$ and $[\lambda]\varphi$ are formulas;
3. if Γ is a countable set of formulas then $\bigwedge \Gamma$ and $\bigvee \Gamma$ are formulas.

Adding nominals, as follows, leads to the set $\mathrm{Fm}_H(\tau, X)$ generated by the rules above plus the following two:

1. nominals are formulas;
2. if φ is a formula and i is a nominal, then $@_i\varphi$ is a formula.

In $\mathrm{Fm}_H(\tau, X)$ formulas defined by nominals or equations are called *atomic*. We abbreviate the formula $\neg[\lambda]\neg\varphi$ to $\langle\lambda\rangle\varphi$, and $\langle\lambda\rangle\varphi \wedge [\lambda]\varphi$ to $\langle\lambda\rangle^\circ\varphi$. We use $[\lambda]^n\varphi$ to denote the application of the operator $[\lambda]$ n times to φ; more precisely, $[\lambda]^0\varphi := \varphi$ and $[\lambda]^{k+1}\varphi := [\lambda][\lambda]^k\varphi$, for $k \geq 0$ (similarly for $\langle\lambda\rangle^n\varphi$). Moreover, an indexed notation is used with boolean connectives $\bigwedge$ and $\bigvee$: for instance $\bigwedge_{k\in\mathbb{N}}[\lambda]^k\varphi$ denotes the formula $\bigwedge\{[\lambda]^k\varphi : k \in \mathbb{N}\}$

In the sequel, let τ be an equational hybrid similarity type.

Definition 3.1 (Kripke frame) *A* Kripke τ-frame *is a structure $\mathcal{F} = (W, R)$, where W is a nonempty set, $R = (R_\lambda)_{\lambda\in\Lambda}$ is a family of binary relations over W. Relations $R_\lambda \subseteq W^2$ are called the*

Beziau, J-Y.; Costa-Leite, A; D'Ottaviano, I. M. L. (orgs.). Aftermath of the Logical Paradise, Coleção CLE, v.81, pp. 173–202, 2018.

transition relations *in* $\mathcal{F}$ *defined over* states *(also called* worlds *or* modes*) in* W*. A* pointed Kripke frame *is a pair* $(\mathcal{F}, w)$*, with* $w \in W$*.*

By adding structure to states we have,

Definition 3.2 (Algebraic Kripke frame) *An* algebraic Kripke τ-frame *is a structure* $\mathcal{M} = (W, R, (A_w)_{w\in W})$*, where* (W, R) *is a Kripke* τ*-frame and* $(A_w)_{w\in W})$ *is a family of* Σ*-algebras. The family* $(A_w)_{w\in W}$ *is called the* space of configurations*. A* pointed algebraic Kripke frame *is a pair* $(\mathcal{M}, w)$ *with* $w \in W$*.*

Finally, adding interpretations for the nominals, we obtain the following definition:

Definition 3.3 (Algebraic hybrid structure) *An* algebraic hybrid structure *over an algebraic* τ*-frame* $\mathcal{M} = (W, (R_\lambda)_{\lambda\in\Lambda}, (A_w)_{w\in W})$ *is a pair* $\mathcal{H} = \langle \mathcal{M}, V\rangle$*, where* $V : \mathrm{Nom} \to W$ *is called a valuation. For* $i \in \mathrm{Nom}$*,* $w = V(i)$ *means that the state* w *is named by* i*. A* pointed algebraic hybrid structure *is a pair* $(\mathcal{H}, w)$ *with* $w \in W$*.*

Definition 3.4 (Satisfaction) *Let* τ *be an equational hybrid similarity type. The* satisfaction relation $\models \subseteq W \times \mathrm{Fm}_H(\tau, X)$ *on the algebraic hybrid structure* $\mathcal{H} = (W, (R_\lambda)_{\lambda\in\Lambda}, (A_w)_{w\in W}, V)$ *is recursively defined as follows: for any* $w \in W$*,* $\varphi, \psi \in \mathrm{Fm}_H(\tau, X)$*,* $i \in \mathrm{Nom}$*,* $t \approx t' \in \mathrm{Eq}_\Sigma(X)$ *and any countable set of formulas* Γ*:*

1. $\mathcal{H}, w \models i$ *if* $V(i) = w$*;*
2. $\mathcal{H}, w \models t \approx t'$ *if* $A_w \models t \approx t'$*;*
3. $\mathcal{H}, w \models @_i\varphi$ *if* $\mathcal{H}, s \models \varphi$*, where* $V(i) = s$*;*
4. $\mathcal{H}, w \models \neg\varphi$ *if not* $\mathcal{H}, w \models \varphi$*;*
5. $\mathcal{H}, w \models \bigvee\Gamma$ *if* $\mathcal{H}, w \models \varphi$ *for some* $\varphi \in \Gamma$*;*
6. $\mathcal{H}, w \models \bigwedge\Gamma$ *if* $\mathcal{H}, w \models \varphi$ *for every* $\varphi \in \Gamma$*;*

Beziau, J-Y.; Costa-Leite, A; D'Ottaviano, I. M. L. (orgs.). Aftermath of the Logical Paradise, Coleção CLE, v.81, pp. 173–202, 2018.

7. $\mathcal{H}, w \models [\lambda]\varphi$ *if for all* $w' \in W$ *such that* $wR_\lambda w'$ *we have* $\mathcal{H}, w' \models \varphi$.

If $\mathcal{H}, w \models \varphi$*, we say that* φ is true at state w in $\mathcal{H}$*. When* φ *is satisfied at every state of* $\mathcal{H}$*, we say that* φ *is* valid in $\mathcal{H}$ *and we write* $\mathcal{H} \models \varphi$*. Finally,* φ *is* valid *if* $\mathcal{H} \models \varphi$ *for every structure* $\mathcal{H}$*.*

Given an algebraic Kripke τ*-frame* $\mathcal{M}$ *and* $w \in W$*,* φ is true at state w in $\mathcal{M}$*, writing* $\mathcal{M}, w \models \varphi$*, if* $\mathcal{H}, w \models \varphi$ *for every* $\mathcal{H}$ *over* $\mathcal{M}$*. A formula* φ *is* valid in $\mathcal{M}$ *and we write* $\mathcal{M} \models \varphi$ *if* $\mathcal{M}, w \models \varphi$ *for every* $w \in W$*. Satisfaction in Kripke frames is defined similarly.* Given a pointed algebraic hybrid structure $(\mathcal{H}, w)$, we write $(\mathcal{H}, w) \models \varphi$ iff $\mathcal{H}, w \models \varphi$.

3.2 Examples

The specification of a reconfigurable system proceeds by identifying a set of configurations (or *modes* of operation), each of them identified by a nominal and endowed with an algebra capturing the relevant, local functionality. Transitions between different configurations are triggered by specifying events encoded as modalities. A small, toy example may help to illustrate the kind of specifications we want to be able to deal with. Consider a calculator with two possible configurations: in one of them an operation $\star$ stands for addition of natural numbers, whereas in the other it corresponds to multiplication. A special button $shift$ leads from one configuration to the other.

This calculator may be viewed as a transition system that alternates between *sum* and *multiplication* modes through an event in $\Lambda = \{shift\}$. Each of its states is associated to a Σ-algebra, where Σ has the following operation symbols $c :\to nat$, $s : nat \to nat$, $p : nat \to nat$ and $\star : nat \times nat \to nat$. Global properties are expressed equationally. For example $p(s(n)) \approx n$ (p and s are the predecessor and successor functions, respectively), $\star(n, k) \approx \star(k, n)$ ($\star$ commutativity) or $\star(n, \star(k, l)) \approx \star(\star(n, k), l)$

Beziau, J-Y.; Costa-Leite, A; D'Ottaviano, I. M. L. (orgs.). Aftermath of the Logical Paradise, Coleção CLE, v.81, pp. 173–202, 2018.

($\star$ associativity).

On the other hand, the specification of local properties, i.e., properties that hold in particular modes, entails the need for the introduction of a nominal, say $\mathrm{Nom} = \{\mathit{ref}\}$, to refer, for instance, to the mode where $\star$ plays the role of a *sum*. Hence, we are able to state, for example

$$@_{\mathit{ref}} \star (n, c) \approx n \quad \text{and} \quad @_{\mathit{ref}} \star (n, s(c)) \approx s(n)$$

or

$$@_{\mathit{ref}} [\mathit{shift}] \star (n, c) \approx c \quad \text{and} \quad @_{\mathit{ref}} [\mathit{shift}] \star (n, s(c)) \approx n$$

Note that we consider the interpretation of c rigid.

Finally, alternation between the two operating modes is captured by modal properties; for example,

$$\neg @_{\mathit{ref}} \langle \mathit{shift} \rangle \mathit{ref} \quad \text{and} \quad @_{\mathit{ref}} [\mathit{shift}][\mathit{shift}] \mathit{ref}$$

Despite the simplicity of this example, infinitary formulas may still be in order, as in, for example, $\star(n, m) \approx l \to \bigwedge\{[\mathit{shift}]^{2k} \star (n, m) \approx l : k \geq 1\}$. Or, for a constant d, $\bigvee\{x^n \approx d : n \geq 0\}$, specifying the existence of a fixed order for every element.

Infinitary formulas are most useful, however, in the specification of more complex systems. For example, let φ and ψ assert some expected properties to be observed periodically in configurations denoted by i and j, respectively. The following property fixes this period along transitions indexed by the modality λ: $\bigvee\{(@_i \langle \lambda \rangle^n \varphi) \wedge (@_j \langle \lambda \rangle^{2n} \psi) : n \geq 0\}$, *i.e.* for some natural n, φ can be observed after n λ-transitions from the state named by i and ψ can be observed after $2n$ λ-transitions from the state named by j. The following examples, in which a unique modality is assumed (and therefore $\langle \lambda \rangle, [\lambda]$ are abbreviated to $\Diamond$ and $\Box$, respectively), illustrate other properties found useful in the speci-

Beziau, J-Y.; Costa-Leite, A; D'Ottaviano, I. M. L. (orgs.). Aftermath of the Logical Paradise, Coleção CLE, v.81, pp. 173–202, 2018.

fication practice:

- i belongs to a finite cycle:

$$\bigvee\{@_i\Diamond^n i : n \geq 1\}$$

- φ is true in at least one state in any finite orbit centered in i:

$$@_i \bigwedge\{\Diamond^n \varphi : n \geq 0\}$$

- (**safety**) φ is true in any state "finitely connected" to i:

$$\bigwedge\{@_i\Box^n \varphi : n \geq 0\}$$

- (**liveness**) φ holds at some state "finitely connected" to i:

$$\bigvee\{@_i\Diamond^n \varphi : n \geq 0\}$$

- there is a fixed order for every element wrt a binary operation:

$$\bigvee\{x^n \approx 1 : n \geq 0\}$$

4. Simulation, bisimulation and bisimilarity

4.1 Simulation

Let τ be an equational hybrid similarity type. Given two algebraic hybrid structures $\mathcal{H} = (W, R, (A_w)_{w\in W}, V)$ and $\mathcal{H}' = (W', R', (A'_w)_{w\in W'}, V')$ over τ, and a binary relation $\precsim$ between Σ-algebras, we say that $\mathcal{H}'$ *simulates* $\mathcal{H}$ modulo $\precsim$ ($\precsim$-simulates, for short) if there is a relation $Z \subseteq W \times W'$ such that

- All points named by nominals are related by Z, i.e., for each $i \in \text{Nom}$, $V(i) Z V'(i)$;
- for every pair $(w, w') \in Z$,

Beziau, J-Y.; Costa-Leite, A; D'Ottaviano, I. M. L. (orgs.). Aftermath of the Logical Paradise, Coleção CLE, v.81, pp. 173–202, 2018.

- $\forall i \in \text{Nom},\ V(i) = w \ \Rightarrow\ V'(i) = w'$,
- $A_w \precsim A'_{w'}$,
- For any $\lambda \in \Lambda$, if $wR_\lambda u$ for some $u \in W$, then there is some $u' \in W'$ such that $w'R'_\lambda u'$ and uZu' **(Zig)**.

The relation Z is called a $\precsim$-*simulation* of $\mathcal{H}$ in $\mathcal{H}'$. There are several natural choices for $\precsim$. We shall consider the following three:

Definition 4.1 *Let A, B be Σ-algebras. Then, define:*

- *$A \precsim_1 B$ iff $B \in \mathcal{V}(A)$*
- *$A \precsim_2 B$ iff $\mathcal{V}(A) = \mathcal{V}(B)$ and*
- *$A \precsim_3 B$ iff $A \cong B$ (i.e. A is isomorphic to B).*

Clearly, $\precsim_3 \subseteq \precsim_2 \subseteq \precsim_1$. Note that the relation $\precsim_2$ corresponds to *elementary equivalence.*

Lemma 4.2 *The $\precsim_i$-simulation is reflexive for $i = 2, 3$ and transitive for $i = 1, 2, 3$.*

Proof Reflexivity of $\precsim_2$ and $\precsim_3$ comes from reflexivity of variety equality and algebra isomorphism, respectively. The other conditions are trivial to check. For transitivity we will show that simulation is closed for relational composition. Let Z_1 and Z_2 be two simulations between $\mathcal{H}$, $\mathcal{H}'$ and $\mathcal{H}'$, $\mathcal{H}''$, respectively, and consider $Z = Z_2 \cdot Z_1$. Clearly, for any nominal i, $V(i)\, Z\, V''(i)$, because Z_1, Z_2 are both simulations.

Consider now $(w, w'') \in Z$. Then, there exists a state $w' \in W'$ such that wZ_1w' and $w'Z_2w''$. Thus,

- $V(i) = w$ implies $V''(i) = w''$, because $V(i) = w$ implies $V'(i) = w'$, as wZ_1w', and the later implies $V''(i) = w''$, as $w'Z_2w''$;

- $A_w \precsim_i A''_{w''}$ because the $\precsim_i$ relations, for $i = 1, 2, 3$, are transitive;

- Finally, suppose $wR_\lambda u$. Z_1 being a simulation and wZ_1w', there exists $u' \in W'$ such that $w'R'_\lambda u'$ and uZ_1u'. On the other hand, Z_2 being also a simulation and $w'Z_2w''$, there exists $u'' \in W''$ such that $w''R''_\lambda u''$ and $u'Z_2u''$. Therefore, $u\,Z\,u''$, as expected.

Simulations are a basic tool in program development. As a consequence of the previous lemma we have a stepwise refinement process for all candidate relations. Actually, let $\mathrm{PE}(\tau)$ be the set of *positive existential infinitary hybrid formulas* over τ, i.e., the set of the formulas built from the atoms (nominals and equations) by just using countable conjunctions and disjunctions, the @ operator and $\Diamond$. Such formulas are preserved under simulations:

Theorem 4.3 *Positive existential infinitary hybrid formulas are preserved under $\precsim_i$-simulation, for $i = 2, 3$. In other words, let τ be an equational hybrid similarity type and Z a $\precsim_i$-simulation between algebraic hybrid structures over τ, let say $\mathcal{H}$ and $\mathcal{H}'$. Then, for any $w \in W, w' \in W'$ such that wZw' we have*

$$(\mathcal{H}, w) \models \varphi \;\Rightarrow\; (\mathcal{H}', w') \models \varphi, \textit{ for any } \varphi \in \mathrm{PE}(\tau).$$

Proof See the proof of Theorem 4.5 below, as the proof of this result can be obtained in a similar way.

4.2 Bisimulation

To define bisimulation the **(Zag)** condition is added, as usual. Additionally, in the atomic conditions, algebras corresponding to related states are required to generate the same variety. Thus,

Definition 4.4 (Bisimulation) *Consider an equational hybrid similarity type τ and two algebraic hybrid structures $\mathcal{H}$ =*

$(W, (R_\lambda)_{\lambda\in\Lambda}, (A_w)_{w\in W}, V)$ *and* $\mathcal{H}' = (W', (R'_\lambda)_{\lambda\in\Lambda}, (A'_w)_{w\in W'}, V')$. *A* bisimulation *between* $\mathcal{H}$ *and* $\mathcal{H}'$ *is a nonempty relation* $Z \subseteq W \times W'$ *such that*

- *All points named by nominals are related by* Z*, i.e., for each* $i \in \mathrm{Nom}$, $V(i)\, Z\, V'(i)$*;*
- *for every pair* $(w, w') \in Z$*,*
  - *Atomic conditions:*
    - $\forall i \in \mathrm{Nom}$, $V(i) = w$ *iff* $V'(i) = w'$*;*
    - $\mathcal{V}(A_w) = \mathcal{V}(A'_{w'})$*, i.e.,* A_w *and* $A'_{w'}$ *generate the same variety;*
  - *For any* $\lambda \in \Lambda$*, if* $wR_\lambda u$ *for some* $u \in W$*, then there is some* $u' \in W'$ *such that* $w'R'_\lambda u'$ *and* uZu' **(Zig)***;*
  - *Dually, for any* $\lambda \in \Lambda$*, if* $w'R'_\lambda u'$ *for some* $u' \in W'$*, then there is some* $u \in W$ *such that* $wR_\lambda u$ *and* uZu' **(Zag)**.

Two pointed algebraic hybrid structures $(\mathcal{H}, w)$ and $(\mathcal{H}', w')$ are *bisimilar*, if there is a bisimulation Z between $\mathcal{H}$ and $\mathcal{H}'$ such that wZw'.

It is well-known that modal satisfaction is invariant under bisimulation. The following theorem establishes a corresponding result for infinitary equational hybrid logic. Again, a similar result holds for Kripke frames and formulas in $\mathrm{Fm}(\tau, X)$.

Theorem 4.5 *The infinitary equational hybrid logic is invariant under bisimulation: let* τ *be an equational hybrid similarity type and* Z *a bisimulation between the algebraic hybrid structures* $\mathcal{H}$ *and* $\mathcal{H}'$*. Then, for any* $w \in W, w' \in W'$ *such that* wZw' *we have*

$$(\mathcal{H}, w) \models \varphi \text{ iff } (\mathcal{H}', w') \models \varphi, \text{ for any } \varphi \in \mathrm{Fm}_H(\tau, X).$$

Proof The proof is by induction over the formulas' structure. Let wZw'. Then,

$\varphi = i \in \text{Nom}$

$$
\begin{array}{ll}
& (\mathcal{H}, w) \models i \\
\Leftrightarrow & \quad \{ \models \text{defn.} \} \\
& V(i) = w \\
\Leftrightarrow & \quad \{ \text{since } Z \text{ is a bisimulation} \} \\
& V'(i) = w' \\
\Leftrightarrow & \quad \{ \models \text{defn.} \} \\
& (\mathcal{H}', w') \models i
\end{array}
$$

$\varphi = @_i \varphi'$

$$
\begin{array}{ll}
& (\mathcal{H}, w) \models @_i \varphi' \\
\Leftrightarrow & \quad \{ \models \text{defn.} \} \\
& (\mathcal{H}, V(i)) \models \varphi' \\
\Leftrightarrow & \quad \{ \text{I.H., since } V(i) Z V'(i) \} \\
& (\mathcal{H}', V'(i)) \models \varphi' \\
\Leftrightarrow & \quad \{ \models \text{defn.} \} \\
& (\mathcal{H}', s) \models @_i \varphi' \text{ for any } s \in W \\
\Leftrightarrow & \quad \{ \text{in particular} \} \\
& (\mathcal{H}', w') \models @_i \varphi'
\end{array}
$$

Beziau, J-Y.; Costa-Leite, A; D'Ottaviano, I. M. L. (orgs.). Aftermath of the Logical Paradise, Coleção CLE, v.81, pp. 173–202, 2018.

$\varphi = t \approx t'$

$$
\begin{array}{ll}
& (\mathcal{H}, w) \models t \approx t' \\
\Leftrightarrow & \qquad \{ \models \text{defn.} \} \\
& A_w \models t \approx t' \\
\Leftrightarrow & \qquad \{ \, wZw' \text{ implies } \mathcal{V}(A_w) = \mathcal{V}(A'_{w'}); \text{ Birkhoff's theorem on varieties} \} \\
& A'_{w'} \models t \approx t' \\
\Leftrightarrow & \qquad \{ \models \text{defn.} \} \\
& (\mathcal{H}', w') \models t \approx t'
\end{array}
$$

$\varphi = \bigwedge \Gamma$ (case $\varphi = \bigvee \Gamma$ is analogous)

$$
\begin{array}{ll}
& (\mathcal{H}, w) \models \bigwedge \Gamma \\
\Leftrightarrow & \qquad \{ \models \text{defn.} \} \\
& (\mathcal{H}, w) \models \gamma \text{ for any } \gamma \in \Gamma \\
\Leftrightarrow & \qquad \{ \text{I.H.} \} \\
& (\mathcal{H}', w') \models \gamma \text{ for any } \gamma \in \Gamma \\
\Leftrightarrow & \qquad \{ \models \text{defn.} \} \\
& (\mathcal{H}', w') \models \bigwedge \Gamma
\end{array}
$$

Beziau, J-Y.; Costa-Leite, A; D'Ottaviano, I. M. L. (orgs.). Aftermath of the Logical Paradise, Coleção CLE, v.81, pp. 173–202, 2018.

$\varphi = [\lambda]\varphi'$

$$
\begin{array}{ll}
 & (\mathcal{H}, w) \models [\lambda]\varphi' \\
\Leftrightarrow & \qquad \{ \models \text{defn.} \} \\
 & (\mathcal{H}, s) \models \varphi' \text{ for any } s \text{ such that } wR_\lambda s \\
\Leftrightarrow & \qquad \{ \text{ I.H.} + \textbf{(Zig)} \text{ and } \textbf{(Zag)} \text{ conditions} \} \\
 & (\mathcal{H}', s') \models \varphi' \text{ for any } s' \text{ such that } w'R'_\lambda s' \\
\Leftrightarrow & \qquad \{ \models \text{defn.} \} \\
 & (\mathcal{H}', w') \models [\lambda]\varphi'
\end{array}
$$

$\varphi = \neg\varphi'$

$$
\begin{array}{ll}
 & (\mathcal{H}, w) \models \neg\varphi' \\
\Leftrightarrow & \qquad \{ \models \text{defn.} \} \\
 & \text{it is false that } (\mathcal{H}, w) \models \varphi' \\
\Leftrightarrow & \qquad \{ \text{ I.H.} \} \\
 & \text{it is false that } (\mathcal{H}', w') \models \varphi' \\
\Leftrightarrow & \qquad \{ \models \text{defn.} \} \\
 & (\mathcal{H}', w') \models \neg\varphi'
\end{array}
$$

As in the standard modal case, given two states of two τ-models, the relation defined by the logical equivalence between the corresponding pointed Kripke models is not in general a bisimulation. Such is the case, however, in image-countable Kripke models, as shown below (the same condition was pointed out in [10] for the plain case of modal logic). More precisely, let τ be an equational hybrid similarity type and $\mathcal{H}$ an algebraic hybrid structure, we say that $\mathcal{H}$ is *image-countable* if for each state $w \in W$ and each relation R_λ, $\lambda \in \Lambda$, the set $\{w' \in W : wR_\lambda w'\}$ is countable. No condition is imposed on the number of relations present or the cardinality of

W.

Theorem 4.6 *Let τ be an equational hybrid similarity type. Let $\mathcal{H}$ and $\mathcal{H}'$ be two* image-countable *algebraic hybrid structures. Then, for every $w \in W$ and $w' \in W'$, the following conditions are equivalent:*

1. *$(\mathcal{H}, w)$ and $(\mathcal{H}', w')$ are bisimilar*
2. *for any $\varphi \in \mathrm{Fm}_H(\tau, X)$, $(\mathcal{H}, w) \models \varphi$ iff $(\mathcal{H}', w') \models \varphi$.*

Proof The proof that 1. implies 2. was already proved in Theorem 4.5. To prove the converse, suppose that for any $\varphi \in \mathrm{Fm}_H(\tau, X)$, $(\mathcal{H}, w) \models \varphi$ iff $(\mathcal{H}', w') \models \varphi$.

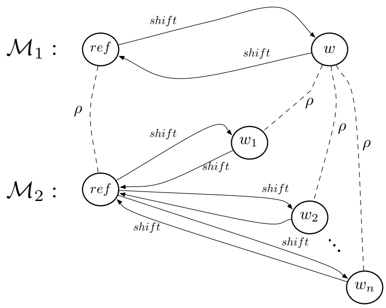
Let $Z := \big\{(w, w') \in W \times W' :$ for any $\varphi \in \mathrm{Fm}_H(\tau, X)$, $(\mathcal{H}, w) \models \varphi \ \Leftrightarrow\ (\mathcal{H}', w') \models \varphi\big\}$. The atomic conditions trivially hold.

For the **(Zig)** condition, let $\lambda \in \Lambda$. Assume that wZw' and let $u \in W$ such that $wR_\lambda u$. To obtain a contradiction, suppose that there is no $u' \in W'$ with $w'R_\lambda u'$ and uZu'. As in the standard case, from the condition of image-countable, the set $S' = \{u' : w'R_\lambda u'\}$ is countable. Moreover, S' cannot be empty since in such case $(\mathcal{H}', w') \models [\lambda]\neg @_i i$ (equivalently, $(\mathcal{H}', w') \models \neg\langle\lambda\rangle @_i i$), which is incompatible with the fact that $(\mathcal{H}, w) \models \langle\lambda\rangle(@_i i)$, which holds since $wR_\lambda u$.

By assumption, for every $v \in S'$ there is a formula ψ_v such that $(\mathcal{H}, u) \models \psi_v$ and it is false that $(\mathcal{H}', v) \models \psi_v$ (it can be at reverse order but in such case we take the negation of the formula).

Consider now the conjunction $\psi = \bigwedge_{v \in S'} \psi_v$ of all of these formulas. Then,

- $(\mathcal{H}, w) \models \langle\lambda\rangle\psi$ and
- for all $v \in S'$ it is false that $(\mathcal{H}, v) \models \langle\lambda\rangle\psi$.



ed.

Figure 1: Two bisimilar models.

This contradicts the fact that wZw'. The **(Zag)** condition can be shown in a similar way.

A consequence of the previous theorem is that any two algebraic hybrid structures with a countable set of states and the same theory are bisimilar. Back to the calculator example, it is obvious that we may construct models for the specification given with any number of states. Actually, any $Z = \big\{(w,v) \mid v \in W_2 \setminus \{V_2(\mathit{ref})\}\big\} \cup \{(V_1(\mathit{ref}), V_2(\mathit{ref}))\}$ is a bisimulation between the algebraic hybrid structures $\mathcal{H}_1$ and $\mathcal{H}_2$ (see Figure 1). Observe that the algebras associated to states w_i may not be the same, but different possible realizations the local theory of the respective mode. For instance, in a loose semantics setting, it is possible to have the free realization of naturals in some mode and the trivial algebra in another one, all in the same model. This situation, which is common in loose semantics specification, may be controlled by imposing some constraints on the definition of models. For instance, in [28] we impose common universes for the realization of system nodes.

Our next result entails the existence of a largest bisimulation, which we call, as usual, the *bisimilarity* relation and denote by

$\sim_{\mathcal{H},\mathcal{H}'}$ (simply, $\sim_{\mathcal{H}}$ if $\mathcal{H} = \mathcal{H}'$). Two elements $p, q \in W$ are said to be *bisimilar* if there is a bisimulation relating them.

Theorem 4.7 *Consider two algebraic hybrid structures over τ, $\mathcal{H} = (W, (R_\lambda)_{\lambda\in\Lambda}, (A_w)_{w\in W}, V)$ and $\mathcal{H}' = (W', (R'_\lambda)_{\lambda\in\Lambda}, (A'_w)_{w\in W'}, V')$. The set of bisimulations between $\mathcal{H}$ and $\mathcal{H}'$ is closed under unions.*

Proof Let Z_1, Z_2 be two bisimulations between $\mathcal{H} = (W, R, (A_w)_{w\in W}, V)$ and $\mathcal{H}' = (W', R', (A'_w)_{w\in W'}, V')$. Relation $Z = Z_1 \cup Z_2$ is also a bisimulation because

- Clearly, all points named by nominals are related by Z as they are either by Z_1 or Z_2.
- The atomic conditions also hold because, if $(w, w') \in Z$, the pair also belongs to Z_1 or Z_2 which are bisimulations.
- A similar argument applies to both the **(Zig)** and the **(Zag)** conditions. For the former (the latter being similar) consider $(w, w') \in Z$ and suppose that, for a $\lambda \in \Lambda$, $wR_\lambda u$ for some $u \in W$. Then there exists $u' \in W'$ such that $w'R'_\lambda u'$ and either uZ_1u' or uZ_2u'. In any case uZu'.

Lemma 4.8 $\sim_{\mathcal{H}}$ *is an equivalence relation.*

Proof Clearly the identity relation and the converse of a bisimulation are bisimulations (for the latter consider the **(Zig)** and **(Zag)** conditions interchangeably). The relation composition of bisimulations is also a bisimulation, with the proof being similar to that of Lemma 4.2 extended to the **(Zag)** condition.

Let τ be an equational hybrid similarity type. Given an algebraic hybrid structure $\mathcal{H} = (W, (R_\lambda)_{\lambda\in\Lambda}, (A_w)_{w\in W}, V)$ over τ, the *reduction* of $\mathcal{H}$ is $\mathcal{H}/\sim_{\mathcal{H}} = (W', (R'_\lambda)_{\lambda\in\Lambda}, (A'_w)_{w\in W'}, V')$ where

- $W' = W/\sim_{\mathcal{H}}$,
- $[w_1]_{\sim_{\mathcal{H}}} R'_\lambda [w_2]_{\sim_{\mathcal{H}}}$ if there are $u_1 \in [w_1]_{\sim_{\mathcal{H}}}$ *and* $u_2 \in [w_2]_{\sim_{\mathcal{H}}}$ such that $u_1 R_\lambda u_2$,
- for any $\bar{w} \in W/\sim_{\mathcal{H}}$,
  $A'_{[w]} = F_{\mathcal{V}(A_w)}(X)$ - the $\mathcal{V}(A_w)$-free algebra over X,
- $V'(i) = [V(i)]_{\sim_{\mathcal{H}}}$.

Note that the definition of $A'_{[w]}$ is sound since, for any $u_1, u_2 \in [w]_{\sim_{\mathcal{H}}}$, $\mathcal{V}(A_{u_1}) = \mathcal{V}(A_{u_2})$. As an example, consider the following useful bisimulation:

Lemma 4.9 *Relation* $Z = \{(w, [w]_{\sim_{\mathcal{H}}}) : w \in W\}$ *is a bisimulation.*

Proof Relation $Z = \{(w, [w]_{\sim_{\mathcal{H}}}) : w \in W\}$ is a bisimulation because

- For any nominal i, $V(i)\, Z\, [V(i)]_{\sim_{\mathcal{H}}}$ by definition of Z, thus, $V(i)\, Z\, V'(i)$;
- The atomic conditions hold trivially;
- For the **(Zig)** condition, let $w\, Z\, [w]_{\sim_{\mathcal{H}}}$ and suppose $w R_\lambda u$, for $\lambda \in \Lambda$. Then $[w]_{\sim_{\mathcal{H}}} R'_\lambda [u]_{\sim_{\mathcal{H}}}$, by definition of R', and clearly $u\, Z\, [u]_{\sim_{\mathcal{H}}}$;
- For the **(Zag)** condition, let $w\, Z\, [w]_{\sim_{\mathcal{H}}}$ and suppose $[w]_{\sim_{\mathcal{H}}} R'_\lambda [u]_{\sim_{\mathcal{H}}}$, for $\lambda \in \Lambda$. This means that there exists $w_1 \in [w]_{\sim_{\mathcal{H}}}$ and $u_1 \in [u]_{\sim_{\mathcal{H}}}$ such that $w_1 R_\lambda u_1$. But because w_1 is bisimilar to w, there is also u_2 bisimilar to u_1 such that $w R_\lambda u_2$. Finally, as u_2 is bisimilar to u, we conclude that $u_2\, Z\, [u]_{\sim_{\mathcal{H}}}$.

5. Conclusions

The paper introduced a logic to specify reconfigurable systems, by extending the classical (propositional) hybrid logic with equations (over an algebraic signature) and infinitary formulas.

As a framework for specifying reconfigurable systems, the approach described here is only part of the picture. Several extensions may indeed be considered. For example, as the reader may have noticed, reconfigurations are triggered by a set of events independent of the local algebras. Often in practice, however, such reconfigurations are guarded by local properties (e.g. by local variables taking specific values). This limitation may be overcome by imposing common data universes on modes and fixing globally a set of rigid variables as proposed in reference [28].

One may go even further, however, and choose the local semantic structure in terms of the specific problem requirements. For example, instead of algebras, one may find necessary to have *partial algebras* to deal with systems with partial operations, or *multi-algebras* for non-determinism, etc. In fact, this choice does not change the essence of the method: at a more abstract level, the authors developed in [30] a method to "hybridise" any logic suitable to specify the local configurations. This process is called *hybridisation*. The basic idea is quite simple: to develop, on top of a given logic, called the *base* logic and framed as an institution [19], the characteristic features of hybrid logic, both at the level of syntax (i.e. modalities, nominals, etc.) and of the semantics (i.e., possible worlds), together with first-order encodings of such hybridised institutions. In particular, given a *base* institution 'encodable' in the institution of theories in first-order logic, the *hybridisation* method provides a systematic construction of a similar encoding for the corresponding hybridised institution ([21]).

Another interesting question is the study of a sound and complete calculus for the logic introduced in this paper with respect to this semantics. The first steps in this direction are reported in [6] where the authors present a sound and complete calculus for non infinitary equational hybrid logic. Related work on equational hybrid type theory is reported in [29].

A final remark is in order to other specification frameworks based on modal versions of first-order logic, also combining data and behaviour features. Abstract state machines [12], popularised through the B method [2], is a prime example. In this case, however, states are identified with the sets of values of the system variables at a given point of execution, evaluated in a unique first-order structure. The approach motivating the research discussed in this paper is fundamentally different in the sense that each state carries its own algebra modelling completely whatever functionality the system offers at such a stage.

References

[1] ABITEBOUL, S., VARDI, M. Y. & VIANU, V. Computing with infinitary logic. Fourth International Conference on Database Theory (ICDT '92). *Theoretical Computer Science*, 149(1), pp. 101 - 128, 1995.

[2] ABRIAL, J. R. *The B Book: Assigning Programs to Meanings.* Cambridge University Press, 1996.

[3] ARECES, C. & BLACKBURN, P. Bringing them all together. *J. Log. Comput.*, 11(5), pp. 657–669, 2001.

[4] ___________ & MARX, M. A road-map on complexity for hybrid logics. In J. Flum and M. Rodriguez-Artalejo, editor, Computer Science Logic (CSL 1999) volume 1683 of *Lecture Notes in Computer Science*, Springer, pp. 307–321, 1999.

Beziau, J-Y.; Costa-Leite, A; D'Ottaviano, I. M. L. (orgs.). Aftermath of the Logical Paradise, Coleção CLE, v.81, pp. 173–202, 2018.

[5] BARWISE, J. On Moschovakis closure ordinals. *J. Symb. Log.*, 42(2), pp. 292–296, 1977.

[6] BARBOSA, L. S., CARRETEIRO, M. & MARTINS, M. A. A Hilbert-style axiomatisation for equational hybrid logic. *Journal of Logic, Language and Information*, 23 (1), pp. 31–52, 2014.

[7] BERGSTRA, J. A. & TUCKER, J. V. A characterisation of computable data types by means of a finite equational specification method. In *Proc. 7th Colloquium on Automata, Languages and Programming*, London, UK. Springer-Verlag, pp. 76–90, 1980.

[8] BLACKBURN, P. Representation, reasoning, and relational structures: a hybrid logic manifesto. *Logic Journal of IGPL*, 8(3), pp. 339–365, 2000.

[9] ―――――― Arthur Prior and hybrid logic. *Synthese*, 150(3), pp.329–372, 2006.

[10] ―――――― & VAN BENTHEM, J. Modal logic: a semantic perspective. In P. Blackburn, F. Wolter, and J. van Benthem, editors, *Handbook of Modal Logic*, Studies in Logic and Practical Reasoning (volume 3). Elsevier, pp. 1–82, 2007.

[11] BOLNADER, T. & BRAÜNER, T. Tableau-based decision procedures for hybrid logic. *J. Log. Comput.*, 16(6), pp.737–763, 2006.

[12] BÖRGER, E. & STÄRK, R. F. *Abstract State Machines. A Method for High-Level System Design and Analysis.* Springer, 2003.

Beziau, J-Y.; Costa-Leite, A; D'Ottaviano, I. M. L. (orgs.). Aftermath of the Logical Paradise, Coleção CLE, v.81, pp. 173–202, 2018.

[13] BRAUNER, T. *Hybrid Logic and its Proof-Theory.* Applied Logic Series. Springer, 2010.

[14] BURRIS, S. & SANKAPPANAVAR, H. P. *A course in universal algebra.* Graduate Texts in Mathematics, Vol. 78. New York - Heidelberg Berlin: Springer-Verlag., 1981.

[15] TEN CATE, B. & FRANCESCHET, M. On the complexity of hybrid logics with binders. In C.-H. L. Ong, editor, *Computer Science Logic (CSL 2005 - Oxford, UK, August 22-25, 2005*, volume 3634 of *Lecture Notes in Computer Science.* Springer, pp. 339–354, 2005.

[16] CLAVEL, M., DURÁN, F., EKER, S., LINCOLN, P., MARTÍ-OLIET, N., MESEGUER, J. & TALCOTT, C. L., editors *All About Maude - A High-Performance Logical Framework, How to Specify, Program and Verify Systems in Rewriting Logic*, volume 4350 of *Lecture Notes in Computer Science.* Springer, 2007.

[17] DAWAR, A. & GRÄDEL, E. Properties of almost all graphs and generalized quantifiers. *Fundam. Inform.*, 98(4), pp. 351–372, 2010.

[18] ROUGEMONT, M. Second-order and inductive definability on finite structures. *Zeitschrift fur Matematische Logik und Grundlagen der Mathematik*, 33, pp.47–63, 1987.

[19] DIACONESCU, R. *Institution-independent Model Theory.* Birkhäuser Basel, 2008.

Beziau, J-Y.; Costa-Leite, A; D'Ottaviano, I. M. L. (orgs.). Aftermath of the Logical Paradise, Coleção CLE, v.81, pp. 173–202, 2018.

[20] ———— & FUTATSUGI, K. Logical foundations of CafeOBJ. *Theor. Comput. Sci.*, 285(2), pp. 289–318, 2002.

[21] ———— & MADEIRA, A. Encoding Hybridised Institutions into First Order Logic. *Mathematical Structures in Computer Science*, Cambrigde Press.(in print)

[22] DICKMANN, M. A. *Large Infinitary Languages.* North-Holland, Amsterdam, 1975.

[23] INDRZEJCZAK, A. Modal hybrid logic. *Logic and Logical Philosophy*, 16, pp.147–257, 2007.

[24] GOGUEN, J. & MALCOLM, G. *Algebraic semantics of imperative programs.* MIT Press Series in the Foundations of Computing. Cambridge, 1996.

[25] KEISLER, H. *Model theory for infinitary logic.* North-Holland, 1971.

[26] KOLAITIS, P. G. & VARDI, M. Y. On the expressive power of datalog: Tools and a case study. In D. J. Rosenkrantz and Y. Sagiv, editors, *Proceedings of the Ninth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, April 2-4, 1990, Nashville, Tennessee, USA.* ACM Press, pp. 61–71, 1990.

[27] ———— & ———— Infinitary logic for computer science. In W. Kuich, editor, *Automata, Languages and Programming, 19th International Colloquium, ICALP92, Vienna, Austria, July 13-17, 1992 - ICALP*, volume 623 of *Lecture Notes in Computer Science*. Springer, pp. 450–473, 1992.

[28] MADEIRA, A., FARIA, J. M., MARTINS, M. A. & BARBOSA, L. S. Hybrid specification of reactive systems: An institutional approach. In G. Barthe, A. Pardo, and G. Schneider, editors, *Software Engineering and Formal Methods (SEFM 2011, Montevideo, Uruguay, November 14-18, 2011)*, volume 7041 of *Lecture Notes in Computer Science*. Springer, pp. 269–285, 2011.

[29] MANZANO, M., MARTINS, M. A. & HUERTAS, A. A Semantics for Equational Hybrid Propositional Type Theory. *Bulletin of the Section of Logic*. 43:3/4, pp. 121–138, 2014.

[30] MARTINS, M. A., MADEIRA, A., DIACONESCU, R. & BARBOSA, L. S. Hybridization of institutions. In A. Corradini, B. Klin, and C. Cîrstea, editors, *Algebra and Coalgebra in Computer Science (CALCO 2011, Winchester, UK, August 30 - September 2, 2011)*, volume 6859 of *Lecture Notes in Computer Science*. Springer, pp. 283–297, 2011.

[31] MOSSAKOWSKI, T., MAEDER, C.& LÜTTICH, K. The heterogeneous tool set, Hets. In O. Grumberg and M. Huth, editors, *Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2007 - Braga, Portugal, March 24 - April 1, 2007)*, volume 4424 of *Lecture Notes in Computer Science*. Springer, pp. 519–522, 2007.

[32] NEVES, R., MADEIRA, A., MARTINS, M. A. & BARBOSA, L. S. Hybridisation at work. In *CALCO TOOLS*, volume 8089 of *Lecture Notes in Computer Science*. Springer, 2013.

[33] PARK, D. Concurrency and automata on infinite sequences. In P. Deussen, editor, *Theoretical Computer Science (5th GI-*

Conference, Karlsruhe, Germany, March 23-25, 1981), volume 104 of *Lecture Notes in Computer Science*. Springer, pp. 167–183, 1981.

[34] PASSY, S. & TINCHEV, T. An essay in combinatory dynamic logic. *Inf. Comput.*, 93(2), pp. 263–332, 1991.

[35] PRIOR, A. N. *Past, Present and Future.* Oxford University Press, 1967.

[36] SANGIORGI, D. *Introduction to bisimulation and conduction.* Cambridge Computer Science Texts. Cambridge University Press, 2012.

[37] SHELAH, S. Nice infinitary logics. Preprin available at `http://arxiv.org/pdf/1005.2806v2.pdf`, 2011.

[38] TANAKA, Y. Kripke completeness of infinitary predicate multimodal logics. *Notre Dame Journal of Formal Logic*, 40(3), pp.326–340, 1999.

[39] TUCKER, J. V. & ZUCKER, J. I. Infinitary initial algebraic specifications for stream algebras. In C. T. W. Sieg, R. Somer, editor, *Reflections on the foundations of mathematics: Essays in honour of Solomon Feferman*. Lecture Notes in Logic, volume 15, Association for Symbolic Logic, pp. 234–253, 2002.

[40] WIRTH, N. *Algorithms + Data Structures = Programs.* Prentice-Hall, 1976.