A Proof of the Interpretability of P/PML in a Relational Setting

Gabriel A. Baum[1] and Marcelo F. Frias[2]

[1] Universidad Nacional de La Plata, LIFIA, Departamento de Informática.
Calle 50 y 115 - 1er. piso (1900), La Plata, Argentina.
Tel./FAX: +54 - 221 - 422 - 8252, e–mail: gbaum@sol.info.unlp.edu.ar
[2] Universidad de Buenos Aires, Departamento de Computación, and
Universidad Nacional de La Plata, LIFIA, Departamento de Informática.
e–mail: mfrias@sol.info.unlp.edu.ar

Abstract. In [1] we presented the logic P/PML, a formalism suitable for the specification and construction of Real–Time systems. The main algebraic result, namely, the interpretability of P/PML into an equational calculus based on ω-closure fork algebras (which allows to reason about Real–Time systems in an equational calculus) was stated but not proved because of the lack of space.
In this paper we present a detailed proof of the interpretability theorem, as well as the proof of the representation theorem for ω-closure fork algebras which provides a very natural semantics based on binary relations for the equational calculus.

1 Introduction

1.1 Motivations

The motivation for this work is the need to describe *industrial processes* as part of a project for a telecommunications company. We want to be able to give formal descriptions of such processes so as to be able to analyze such descriptions. For example, we want to be able to calculate critical paths for tasks in processes, throughput times of processes, etc. We also want to demonstrate correctness of process descriptions in relation to their specifications (where this is appropriate), derive implementations of process specifications in terms of the available concrete apparatus in the factory, validate (using formal techniques) an implementation against its abstract description, and so on. Available languages for describing processes are unsuitable for various reasons, most having to do with the nature of the formalization of such processes being used in the project.

The method used in the project for describing industrial processes (*the method*) is based on the ideas presented in [12]. This method sees the world as being modeled in terms of two (and only two) kinds of entities: products and processes. A *product* is a description of an entity in the real world (*a referent*) in terms of measurable attributes. (Here, we use *measure* and *measurable* in the traditional sense of science and engineering. See [3][12][18].) A *product instance* is characterized by the values (measures) associated with its attributes (and, implicitly, by the theory of the product, i.e., the defined relationships between the potential measured values of its attributes). Hence, such a product instance may be seen as a *model*, in the sense of logic, of the product. We may see products as being characterized by data types in first order logic, for example.

The distinguishing characteristic of products is that they exist 'independently' at an instant in time, where *time* is used here in its normal scientific sense. (Independence here means that a product is defined without recourse to any other referent or only in terms of other (sub)products. Products of the former kind are called *atomic* products.) In fact, all products have a time attribute whose value in a product instance indicates the time instant at which the values of the attributes were (co)determined, presumably by some appropriate measurement procedures. On the other hand, processes are distinguished entities which do not exist at a time instant, but which have time duration. Further, processes are not independently definable, but are defined in terms of their input and output products.

Processes also model entities of the real world and again are defined in terms of attributes. The method imposes a very restrictive notion of process, namely one in which *all processes have a single input and a single output.* (The reasons for this restriction need not detain us here, except to say that they are methodologically very well motivated. The restriction clearly will have a profound influence on the nature of the language we define below.) Distinguished attributes of a process include the transfer function(s) 'computed' by the process (i.e., how the input product is transformed into the output product), upper and lower bounds on the time taken for the process to execute, a flag indicating whether the process is 'enabled', and so on. The transfer function may be described in terms of an underlying state machine used to organize phases of the process being defined and to 'sense' important external state information required to control the execution of the process. Like products, processes may be defined in terms of 'sub-processes' and we now turn to this language of processes.

We can see an analogy between inputs/outputs and products and between programs and processes. Both programs and processes are intended to model entities that define families of executions on the machine used to execute the program/process. This is exactly how we want to understand processes, i.e., as defining a class of potential executions over some (abstract) machine. We do not envisage a single abstract machine which will underpin all potential processes. Rather, we assume that our abstract machine is provided by an object, in the sense of object oriented programming.

1.2 The Results

In [1], the logic P/PML was defined by extending first order dynamic logic with a parallel combinator and the ability to express real time constraints. The semantics of dynamic logic uses a notion of transition system that is used to represent the underlying abstract machine capable of executing the atomic processes. The logic was extended with variables over processes so that we can specify abstractly the processes we are interested in building. There is a notion of refinement associated with such specifications, allowing us to demonstrate that a process satisfies its specification.

In this paper we demonstrate, using techniques developed in [8][9], how to algebraize this logic and thus obtain an equational proof system for our process formalism. In order to algebraize the logic we will use *omega closure fork* *algebras* (ωCFA). These algebras are extensions of relation algebras [17] with three new operators, a pairing operator called *fork* [7][6], a *choice* operator [15] and the Kleene star. A consequence preserving function mapping formulas of the logic to equations in the language of ωCFA will be defined. The use of the mapping enables the use of equational inference tools in the process of systems construction. Even though having the possibility of using an equational calculus is interesting by itself, this calculus has the particularity of being complete with respect to a very insightful and clean semantics based on binary relations. The last is guaranteed by the proof of a representation theorem for the class ωCFA.

In order to simplify the reading of the paper, the logic and the classes of algebras (already presented in [1]) are presented here once more. The reader interested in more comprehensive motivations and examples is strongly encouraged to read the paper [1].

The paper is organized as follows: in Section 2 we will present a first order formalization of objects. In Section 3 will be presented the logic we propose for specifying and reasoning about the properties of processes. In Section 4 we introduce the class of omega closure fork algebras and prove the representation theorem. In Section 5 we present the algebraization and prove the interpretability theorem. Finally, in Section 6 we present our conclusions about this work.

2 Objects

The first problem we confront when trying to formalize the previous concepts is that of characterizing the 'abstract machine' over which our processes will be defined. These processes are meant to use the underlying capabilities of the organization, as represented by the behaviors displayed by individual components within the organization. (Such individual components may be people, groups, manufacturing machines, etc.) These behaviors are organized (at least in some abstract sense) into a joint behavior which IS our 'abstract machine'. We will assume as given some object (which may be very complex and built as a system from less complex components [4][5]), which represents the potential behaviors of the organization as an abstract machine. The definitions below give a somewhat non standard account of objects in terms of the underlying transition system defining the object's allowed behaviors. However, the standard parts of such descriptions (i.e., methods, state variables, etc) are easily distinguishable.

Definition 1. An *object signature* is a pair $\langle A, \Sigma \rangle$ in which $\Sigma = \langle S, F, P \rangle$ is a many-sorted first-order signature with set of sorts S, set of function symbols F and set of predicate symbols P. Among the sorts, we will single out one sort called the *time sort*, denoted by T. A is a set of *action symbols.* To each $a \in A$ is associated a pair $\langle s_1, s_2 \rangle \in (S^*)^2$ called its *arity.* We will denote the input arity of a by $ia(a)$ and the output arity of a by $oa(a)$.

Definition 2. Given an object signature $\mathcal{S} = \langle A, \langle S, F, P \rangle \rangle$, an *object structure for $\mathcal{S}$* is a structure $\mathcal{A} = \langle \mathbf{S}, \mathbf{A}, \mathbf{F}, \mathbf{P} \rangle$ in which $\mathbf{S}$ is an S-indexed family of nonempty sets, where the set T is the T-th element in $\mathbf{S}$. In general, the set corresponding to sort s will be denoted by s. $\mathbf{A}$ is an A-indexed family of binary relations satisfying the typing constraints of symbols from A, i.e., if $ia(a) = s_1 \ldots s_m \in S^*$ and $oa(a) = {s'}_1 \ldots {s'}_n \in S^*$, then $a^{\mathcal{A}}$ (as we will denote the a-th element from $\mathbf{A}$) is contained in $(\mathsf{s}_1 \times \cdots \times \mathsf{s}_m) \times (\mathsf{s'}_1 \times \cdots \times \mathsf{s'}_n)$. To each $f : s_1 \ldots s_k \to s$ in F is associated a function $f^{\mathcal{A}} : \mathsf{s}_1 \times \cdots \times \mathsf{s}_k \to \mathsf{s} \in \mathbf{F}$. To each p of arity $s_1 \ldots s_k$ in P is associated a relation $p^{\mathcal{A}} \subseteq \mathsf{s}_1 \times \cdots \times \mathsf{s}_k \in \mathbf{P}$.

Regarding the domain T associated to the time sort T, we will not deepen on the different possibilities for modeling time, but will rather choose some adequate (with respect to the application we have in mind) representation, as for instance the fields of rational or real numbers, extended with a maximum element ∞. We will distinguish some constants, as 0, ϵ, etc.

3 The Logic, the Relational Variables and the Time

In this section we will present the *Product/Process Modeling Logic* (*P/PML*). In order to achieve this goal we will extend a standard notation for specifying and reasoning about programs, namely dynamic logic.

An important aspect of *P/PML* is the real–time aspect. We adapt a real–time logic developed in [2] which presents an extension of the logic presented in [4]. Each basic action is supplemented with a specification of lower and upper time bounds for occurrences of that action. These bounds may have various interpretations, amongst which we have the following: the lower bound is interpreted as the minimum time that must pass before which the action's effects are committed to happen and the upper bound gives a maximum time by which the action's effects are committed to happen. Specifications of processes will also have associated lower and upper bounds, and refinements will be expected to provably meet these bounds.

Consider the formula $\varphi(x) := [x\,A\,x]\beta(x)$ where A is an action term (a binary or n-ary relation) and the notation $[x\,A\,x]\beta$ means that "all executions of action A establish the property β". According to our previous discussion about processes and products, we read φ as stating that β is a truth of the system A, then proving the truth of φ can be seen as the *verification* of the property β in the system described by A. Opposed to the previous view, is the notion of an *implicit specification* of a system, in which A is not a ground term, but rather may contain some relational variables that represent subsystems not yet fully determined. In what follows we will denote by *RelVar* the set of relational variables $\{ R, S, T, \ldots \}$.

Definition 3. Given an object signature $\mathcal{S} = \langle A, \langle S, F, P \rangle \rangle$, the sets of *relational terms* and *formulas* on $\mathcal{S}$ are the smallest sets $RT(\mathcal{S})$ and $For(\mathcal{S})$ such that

1. $a \in RT(\mathcal{S})$ for all $a \in A \cup \mathit{RelVar} \cup \{ 1'_t : t \in S^* \}$.
2. If $r \in RT(\mathcal{S})$ and $ia(r) = oa(r)$, then $r^* \in RT(\mathcal{S})$. We define $ia(r^*) = oa(r^*) = ia(r)$.
3. If $r, s \in RT(\mathcal{S})$, $ia(r) = ia(s)$ and $oa(r) = oa(s)$, then $r + s \in RT(\mathcal{S})$ and $r \cdot s \in RT(\mathcal{S})$. We define $ia(r+s) = ia(r \cdot s) = ia(r)$ and $oa(r+s) = oa(r \cdot s) = oa(r)$.
4. If $r, s \in RT(\mathcal{S})$ and $oa(r) = ia(s)$, then $r;s \in RT(\mathcal{S})$. We define $ia(r;s) = ia(r)$ and $oa(r;s) = oa(s)$.

5. If $\alpha \in For(\mathcal{S})$ is quantifier free and has free variables $x_1, \ldots, x_n$ with x_i of sort s_i, then $\alpha? \in RT(\mathcal{S})$ and $ia(\alpha?) = oa(\alpha?) = s_1 \ldots s_n$.
6. The set of first-order atomic formulas on the signature Σ is contained in $For(\mathcal{S})$.
7. If $\alpha, \beta \in For(\mathcal{S})$, then $\neg\alpha \in For(\mathcal{S})$ and $\alpha \vee \beta \in For(\mathcal{S})$..
8. If $\alpha \in For(\mathcal{S})$ and x is an individual variable of sort s, then $(\exists x : s)\,\alpha \in For(\mathcal{S})$.
9. If $\alpha \in For(\mathcal{S})$, $t \in RT(\mathcal{S})$ with $ia(t) = s_1 \ldots s_m$ and $oa(t) = s'_1 \ldots s'_n$, $\vec{x}= x_1, \ldots, x_m$ with x_i of sort s_i, $\vec{y}= y_1, \ldots, y_n$ with y_i of sort s'_i and l, u are variables of sort T, then $\left\langle \vec{x}\ {}_l t^u\ \vec{y} \right\rangle \alpha \in For(\mathcal{S})$.

Definition 4. Let $R \in RT(\mathcal{S})$ with $ia(R) = s_1 \ldots s_m$ and $oa(R) = s'_1 \ldots s'_n$, $\vec{x}= x_1, \ldots, x_m$ with x_i of sort s_i, $\vec{y}= y_1, \ldots, y_n$ with y_i of sort s'_i, and l, u variables of sort T. An expression of the form $\vec{x}\ {}_l R^u\ \vec{y}$ is called a *timed action term.*

We will assume that a lower and an upper bound are assigned to atomic actions, namely $l_a \in \mathsf{T}$ and $u_a \in \mathsf{T}$ for each action $a \in A$. From the bounds of the atomic actions it is possible to define bounds for complex actions in a quite natural way.

Definition 5. Let $\mathcal{S}$ be an object signature. The functions l and u from $RT(\mathcal{S}) \cup For(\mathcal{S})$ to T are defined as follows[1]:

1. If $a \in A$, then $\mathsf{l}(a) = l_a$ and $\mathsf{u}(a) = u_a$.
2. If $R = X \in RelVar$, then $\mathsf{l}(X) = 0$ and $\mathsf{u}(X) = \infty$.
3. If $R = 1\text{'}_t$, with $t \in S^*$, then $\mathsf{l}(R) = 0$ and $\mathsf{u}(R) = \epsilon$ (ϵ being a constant of sort T).
4. If $R = S^*$, then $\mathsf{l}(R) = 0$ and $\mathsf{u}(R) = \infty$.
5. If $R = S{+}T$, then $\mathsf{l}(R) = \min\{\,\mathsf{l}(S), \mathsf{l}(T)\,\}$ and $\mathsf{u}(R) = \max\{\,\mathsf{u}(S), \mathsf{u}(T)\,\}$.
6. If $R = S{\cdot}T$, then $\mathsf{l}(R) = \max\{\,\mathsf{l}(S), \mathsf{l}(T)\,\}$ and $\mathsf{u}(R) = \max\{\,\mathsf{u}(S), \mathsf{u}(T)\,\}$.
7. If $R = S;T$, then $\mathsf{l}(R) = \mathsf{l}(S)$ and $\mathsf{u}(R) = \mathsf{u}(S) + \mathsf{u}(T)$.
8. If $R = \alpha?$ with $\alpha \in For(\mathcal{S})$ quantifier free and with free variables $\vec{x}$, $\mathsf{l}(R) = \mathsf{l}(\alpha)$ and $\mathsf{u}(R) = \mathsf{u}(\alpha)$.
9. If $\alpha = p(t_1, \ldots, t_k)$, then $\mathsf{l}(\alpha) = l_p \in \mathsf{T}$ and $\mathsf{u}(\alpha) = u_p \in \mathsf{T}$, with $l_p \leq u_p$.
10. If $\alpha = \neg\beta$, then $\mathsf{l}(\alpha) = \mathsf{l}(\beta)$ and $\mathsf{u}(\alpha) = \mathsf{u}(\beta)$.
11. If $\alpha = \beta\mathbf{op}\gamma$ with $\mathbf{op} \in \{\,\vee, \wedge, \rightarrow\,\}$, then $\mathsf{l}(\alpha) = \min\{\,\mathsf{l}(\beta), \mathsf{l}(\gamma)\,\}$ and $\mathsf{u}(\alpha) = \max\{\,\mathsf{u}(\beta), \mathsf{u}(\gamma)\,\}$.
12. If $\alpha = \left\langle \vec{x}\ {}_l R^u\ \vec{y} \right\rangle \beta$, then $\mathsf{l}(\alpha) = \mathsf{l}(R)$ and $\mathsf{u}(\alpha) = \mathsf{u}(R) + \mathsf{u}(\beta)$.

Given a set of sorts $S = \{\,s_1, \ldots, s_k\,\}$ and domains $\mathsf{S} = \{\,\mathsf{s}_1, \ldots, \mathsf{s}_k\,\}$ for these sorts, by a *valuation of the individual variables of sort* s_i we refer to a function $\nu : IndVar_{s_i} \rightarrow \mathsf{s}_i$. A valuation of the relational variables is a function $\mu : RelVar \rightarrow \mathcal{P}\,(\mathsf{S}^* \times \mathsf{S}^*)$.

[1] We will only consider quantifier-free formulas, since these are the ones used for building actions of the form $\alpha?$.

Definition 6. Given a valuation of the individual variables ν and an array of variables $\overrightarrow{x} = x_1, \ldots, x_n$, by $\nu(\overrightarrow{x})$ we denote the tuple $\langle \nu(x_1), \ldots, \nu(x_n) \rangle$.

Let $\mathcal{A}$ be an object structure and μ a valuation of the relational variables. Given valuations of the individual variables ν and ν' and a timed action term $\overrightarrow{x}\ {}_l R^u\ \overrightarrow{y}$, by $\nu\left(\overrightarrow{x}\ {}_l R^u\ \overrightarrow{y}\right)\nu'$ we denote the fact that: $\left\langle \nu(\overrightarrow{x}), \nu'(\overrightarrow{y}) \right\rangle \in R_\mu^\mathcal{A}$ (the denotation of the relational term R, formally defined in Def. 7), for every variable z not occurring in $\overrightarrow{y}$, $\nu'(z) = \nu(z)$, and, $\nu(l) \leq \mathsf{l}(R)$ and $\nu(u) \geq \mathsf{u}(R)$.

The semantics of formulas is now defined relative to valuations of individual variables and relational variables. In the following definition, the notation $\mathcal{A} \models_{P/PML} \alpha[\nu][\mu]$, is to be read "*The formula α is satisfied in the object structure $\mathcal{A}$ by the valuations ν and μ*".

Definition 7. Let us have an object signature $\mathcal{S} = \langle A, \langle S, F, P \rangle \rangle$ and an object structure $\mathcal{A} = \langle \mathbf{S}, \mathbf{A}, \mathbf{F}, \mathbf{P} \rangle$. Let ν be a valuation of individual variables and μ a valuation of relational variables. Then:

1. If $a \in A$ then $a_\mu^\mathcal{A}$ is the element with index a in $\mathbf{A}$.
2. If $R \in \mathit{RelVar}$, then $R_\mu^\mathcal{A} = \mu(R)$.
3. If $R = 1'_t$ with $t = s_1 \ldots s_k$, $R_\mu^\mathcal{A} = \{ \langle \langle a_1, \ldots, a_k \rangle, \langle a_1, \ldots, a_k \rangle \rangle : a_i \in \mathsf{s}_i \}$.
4. If $R = S^*$, with $S \in RT(\mathcal{S})$, then $R_\mu^\mathcal{A}$ is the reflexive-transitive closure of the binary relation $S_\mu^\mathcal{A}$.
5. If $R = S + T$, with $S, T \in RT(\mathcal{S})$, then $R_\mu^\mathcal{A} = S_\mu^\mathcal{A} \cup T_\mu^\mathcal{A}$.
6. If $R = S \cdot T$, with $S, T \in RT(\mathcal{S})$, then $R_\mu^\mathcal{A} = S_\mu^\mathcal{A} \cap T_\mu^\mathcal{A}$.
7. If $R = S;T$, with $S, T \in RT(\mathcal{S})$, then $R_\mu^\mathcal{A}$ is the composition of the binary relations $S_\mu^\mathcal{A}$ and $T_\mu^\mathcal{A}$.
8. If $R = \alpha?$ with $\alpha \in \mathit{For}(\mathcal{S})$ quantifier free and with free variables $\overrightarrow{x} = x_1, \ldots, x_n$, then $R_\mu^\mathcal{A} = \left\{ \left\langle \nu(\overrightarrow{x}), \nu(\overrightarrow{x}) \right\rangle : \mathcal{A} \models_{P/PML} \alpha[\nu][\mu] \right\}$.
9. If $\varphi = p(t_1, \ldots, t_n)$ with $p \in P$, $\mathcal{A} \models_{P/PML} \varphi[\nu][\mu]$ if $\langle t_{1\nu}^{\ \mathcal{A}}, \ldots, t_{n\nu}^{\ \mathcal{A}} \rangle \in p^\mathcal{A}$.
10. If $\varphi = \neg\alpha$, then $\mathcal{A} \models_{P/PML} \varphi[\nu][\mu]$ if $\mathcal{A} \not\models_{P/PML} \alpha[\nu][\mu]$.
11. If $\varphi = \alpha \vee \beta$, $\mathcal{A} \models_{P/PML} \varphi[\nu][\mu]$ if $\mathcal{A} \models_{P/PML} \alpha[\nu][\mu]$ or $\mathcal{A} \models_{P/PML} \beta[\nu][\mu]$.
12. If $\varphi = (\exists x : s)\alpha$, then $\mathcal{A} \models_{P/PML} \varphi[\nu][\mu]$ if there exists $a \in \mathsf{s}$ such that $\mathcal{A} \models_{P/PML} \alpha[\nu_x^a][\mu]$ (ν_x^a, as usual, denotes the valuation that agrees with ν in all variables but x, and satisfies $\nu_x^a(x) = a$).
13. If $\varphi = \left\langle \overrightarrow{x}\ {}_l R^u\ \overrightarrow{y} \right\rangle \alpha$, then $\mathcal{A} \models_{P/PML} \varphi[\nu][\mu]$ if there exists a valuation ν' such that $\nu\left(\overrightarrow{x}\ {}_l R^u\ \overrightarrow{y}\right)\nu'$ and $\mathcal{A} \models_{P/PML} \alpha[\nu'][\mu]$.

4 Omega Closure Fork Algebras

Equational reasoning based on substitution of equals for equals is the kind of manipulation that is performed in many information processing systems. The role of equational logics in development of formal methods for computer science applications is increasingly recognized and various tools have been developed for modeling user's systems and carrying through designs within the equational framework (Gries and Schneider [11], Gries [10]).

In this section we present the *omega calculus for closure fork algebras* (ωCCFA), an extension of the *calculus of relations* (CR) and of the *calculus of relations with fork* [6]. Because of the non enumerability of the theory of dynamic logic, an infinitary equational inference rule will be required in ωCCFA. From the calculus we define the class ωCFA of the *omega closure fork algebras* and a representation theorem is presented, showing that the Kleene star as axiomatized, indeed characterizes reflexive-transitive closure.

In the following paragraphs we will introduce the *Omega Calculus for Closure Fork Algebras* (ωCCFA).

Definition 8. Given a set of relation symbols R, the set of ωCCFA terms on R is the smallest set $T\omega\mathsf{CCFA}(R)$ satisfying: $R \cup \mathit{RelVar} \cup \{0, 1, 1'\} \subseteq T\omega\mathsf{CCFA}(R)$. If $x \in T\omega\mathsf{CCFA}(R)$,then $\{\breve{x}, x^*, x^\diamond\} \subseteq T\omega\mathsf{CCFA}(R)$. If $x, y \in T\omega\mathsf{CCFA}(R)$,then $\{x+y, x\cdot y, x;y, x\nabla y\} \subseteq T\omega\mathsf{CCFA}(R)$.

The symbol $^\diamond$ denotes a *choice* function (see [15, §3]), which is necessary in order to prove Thm. 2.

Definition 9. Given a set of relation symbols R, the set of ωCCFA formulas on R is the set of identities $t_1 = t_2$, with $t_1, t_2 \in T\omega\mathsf{CCFA}(R)$.

Definition 10. Given terms $x, y, z, w \in T\omega\mathsf{CCFA}(R)$, the identities defined by the following conditions are axioms:

Identities axiomatizing the relational calculus [17],

The following three axioms for the fork operator:

$$x \nabla y = (x; (1' \nabla 1)) \cdot (y; (1 \nabla 1')), \tag{Ax. 1}$$

$$(x \nabla y);(z \nabla w)^{\smile} = (x;\breve{z}) \cdot (y;\breve{w}), \tag{Ax. 2}$$

$$(1' \nabla 1)^{\smile} \nabla (1 \nabla 1')^{\smile} \leq 1'. \tag{Ax. 3}$$

The following three axioms for the choice operator, taken from [15, p. 324]:

$$x^\diamond;1;\breve{x^\diamond} \leq 1', \tag{Ax. 4}$$

$$\breve{x^\diamond};1;x^\diamond \leq 1', \tag{Ax. 5}$$

$$1;(x \cdot x^\diamond);1 = 1;x;1. \tag{Ax. 6}$$

The following two axioms for the Kleene star:

$$x^* = 1' + x;x^*, \tag{Ax. 7}$$

$$x^*;y \leq y + x^*;(\overline{y} \cdot x;y). \tag{Ax. 8}$$

Let us denote by $1'_U$ the partial identity $Ran\left(\overline{1\nabla 1}\right)$. Then, the following axiom is added

$$1;1'_U;1 = 1, \tag{Ax. 9}$$

which states the existence of a nonempty set of non splitting elements (that we will call *urelements*).

Theorem 1. *The following properties are derivable in the calculus* ωCCFA*.*

1. * *is a monotonic operator, i.e., if* $x \leq y$*, then* $x^* \leq y^*$*.*
2. *If* x *is reflexive and transitive, then* $x^* = x$*.*
3. *If* y *is reflexive and transitive and* $x \leq y$*, then* $x^* \leq y$*.*

The proof of Thm. 1 requires simple equational manipulations.

Notice that from Thm. 1, x^* is the smallest reflexive and transitive relation that includes x. The rules of inference for the calculus ωCCFA are those of equational logic adding the following inference rule[2]:

$$\frac{\vdash 1' \leq y \qquad x^i \leq y \vdash x^{i+1} \leq y}{\vdash x^* \leq y}$$

Definition 11. We define the class of the *omega closure fork algebras* (ωCFA) as the models of the identities provable in ωCCFA.

The standard models of the ωCCFA are the *Proper Closure Fork Algebras* (PCFA for short). In order to define the class PCFA, we will first define the class •PCFA.

Definition 12. Let E be a binary relation on a set U, and let R be a set of binary relations. A •PCFA is a two sorted structure with domains R and U $\langle R, U, \cup, \cap, ^-, \emptyset, E, ;, Id, \breve{}, \nabla, ^\diamond, ^*, \star\rangle$ such that

1. $\bigcup R \subseteq E$,
2. $\star : U \times U \to U$ is an injective function when its domain is restricted to the set E,
3. If we denote by Id the identity relation on the set U, then $\emptyset$, E and Id belong to R,
4. R is closed under set choice operator defined by the condition:

$$x^\diamond \subseteq x \qquad \text{and} \qquad |x^\diamond| = 1 \iff x \neq \emptyset.$$

5. R is closed under set union ($\cup$), intersection ($\cap$), complement relative to E ($^-$), composition of binary relations ($;$), converse ($\breve{}$), reflexive-transitive closure (*) and fork, the last being defined by the formula

$$S \nabla T = \{\langle x, \star(y,z)\rangle : x S y \text{ and } x T z\}.$$

Note that $x^\diamond$ denotes an arbitrary pair in x. That is why $x^\diamond$ is called a choice operator. We will call the set U in Def. 12 the *field* of the algebra, and will denote the field of an algebra $\mathbf{A}$ by $U_{\mathbf{A}}$. We will also denote the set of urelements of $\mathbf{A}$ by $Urel_{\mathbf{A}}$.

[2] Given $i > 0$, by x^i we denote the relation inductively defined as follows: $x^1 = x$, and $x^{i+1} = x;x^i$.

Definition 13. We define the class PCFA as $\mathbf{Rd}\bullet\mathsf{PCFA}$ where $\mathbf{Rd}$ takes reducts to structures of the form $\langle R, \cup, \cap, ^{-}, \emptyset, E, ;, Id, ^{\smile}, \nabla, ^{\diamond}, ^{*} \rangle$.

Notice that given $\mathbf{A} \in \mathsf{PCFA}$, the terms $(1\text{'}\nabla 1)^{\smile}$ and $(1\nabla 1\text{'})^{\smile}$ denote respectively the binary relations $\{ \langle a \star b, a \rangle : a, b \in A \}$ and $\{ \langle a \star b, b \rangle : a, b \in A \}$. Thus, they behave as projections with respect to the injection $\star$. We will denote these terms by π and ρ, respectively.

From the operator fork we define $x \otimes y = (\pi ; x) \nabla (\rho ; y)$. The operator $\otimes$ (*cross*), when interpreted in an proper closure fork algebra behaves as a parallel product: $x \otimes y = \{ \langle a \star b, c \star d \rangle : \langle a, c \rangle \in x \ \wedge \ \langle b, d \rangle \in y \}$.

A relation R is *constant* if it satisfies: $\breve{R} ; R \leq 1\text{'}$, $1 ; R = R$, and $R ; 1 = 1$. Constant relations are alike constant functions, i.e., they relate every element from the domain to a single object[3]. We will denote the constant whose image is the value a by C_a.

Definition 14. We denote by FullPCFA the subclass of PCFA in which the relation E equals $U \times U$ for some set U and R is the set of all binary relations contained in E.

Similarly to the relation algebraic case, where every proper relation algebra (PRA) $\mathbf{A}$ belongs to[4] $\mathbf{ISP}$FullPRA, it is easy to show that every PCFA belongs to $\mathbf{ISP}$FullPCFA. We finally present the representation theorem for $\omega\mathsf{CFA}$.

Theorem 2. *Given* $\mathbf{A} \in \omega\mathsf{CFA}$*, there exists* $\mathbf{B} \in \mathsf{PCFA}$ *such that* $\mathbf{A}$ *is isomorphic to* $\mathbf{B}$*.*

Proof. Let us consider the fork algebra reduct $\mathbf{A}'$ of the algebra $\mathbf{A}$. By the representation theorem for fork algebras [7] there exists a proper fork algebra $\mathbf{C}'$ such that $\mathbf{A}'$ is isomorphic to $\mathbf{C}'$ (we will denote by $\underline{\nabla}$ the fork operation in $\mathbf{C}'$). Let $h : \mathbf{A}' \to \mathbf{C}'$ be an isomorphism. Let us consider the structure $\mathbf{C} = \langle \mathbf{C}', ^{\underline{\diamond}}, ^{\underline{*}} \rangle$, where $^{\underline{\diamond}}$ and $^{\underline{*}}$ are defined by the conditions:

$$x^{\underline{\diamond}} = h\left(\left(h^{-1}(x)\right)^{\diamond}\right) \text{ and } x^{\underline{*}} = h\left(\left(h^{-1}(x)\right)^{*}\right) \quad \text{for each } x \in \mathbf{C}'.$$

It is easy to check that $h : \mathbf{A} \to \mathbf{C}$ is an $\omega\mathsf{CFA}$ isomorphism. It is also easy to check using the infinitary rule that for $x \in \mathbf{A}$, $x^{*} = \mathbf{lub}\left\{ x^{i} : i \in \omega \right\}$. Thus, using the isomorphism h, we can prove that for each $x \in \mathbf{C}$,

$$x^{\underline{*}} = \mathbf{lub}\left\{ x^{i} : i \in \omega \right\}. \tag{1}$$

Notice that it is not necessarily the case that for $x \in \mathbf{C}$, $x^{\underline{*}}$ equals the reflexive-transitive closure of x, since infinite sums may not correspond to the infinite union of the binary relations[5].

[3] This comment is in general a little strong and applies to *simple* algebras, but is nevertheless useful as an intuitive aid for the non specialist.

[4] By $\mathbf{I}$, $\mathbf{S}$ and $\mathbf{P}$ we denote the closure of an algebraic class under isomorphic copies, subalgebras and direct products, respectively.

[5] Examples of proper relation algebras in which $\mathbf{lub}A$ does not agree with $\bigcup A$ (for some set A) are given in [13][14] and elsewhere.

Let $\mathbf{D}$ be $\mathbf{C}$ RA reduct. Since $\mathbf{D}$ is point-dense (see [15][16] for details on point-density), by [15, Thm. 8] $\mathbf{D}$ has a complete representation $\mathbf{D}'$. Let $g : \mathbf{D} \to \mathbf{D}'$ be an isomorphism satisfying $g\left(\sum_{i\in I} x_i\right) = \bigcup_{i\in I} g\left(x_i\right)$.

Let $\mathbf{B} = \langle \mathbf{D}', \nabla, {}^{\diamond}, {}^{*} \rangle$, where ∇, ${}^{\diamond}$ and * are defined as follows:

$$\begin{aligned} x\nabla y &= g\left(g^{-1}(x)\underline{\nabla}g^{-1}(y)\right),\\ x^{\diamond} &= g\left(\left(g^{-1}(x)\right)^{\underline{\diamond}}\right),\\ x^{*} &= g\left(\left(g^{-1}(x)\right)^{\underline{*}}\right). \end{aligned}$$

Then, $g : \mathbf{C} \to \mathbf{B}$ is also an isomorphism. That ∇ as defined is a fork operation on $\mathbf{B}$ follows from the fact the relations $g(\pi)$ and $g(\rho)$ are a pair of quasi-projections [15][19] in $\mathbf{B}$. Since ${}^{\diamond}$ satisfies Ax. 4–Ax. 6, it is easy to prove that it is a choice operator. Finally, let us check that x^{*} is the reflexive-transitive closure of x, for each $x \in \mathbf{B}$.

$$\begin{aligned} x^{*} &= g\left(\left(g^{-1}\left(x\right)\right)^{\underline{*}}\right) && \text{(by Def. } x^{*}\text{)}\\ &= g\left(\sum_{i\geq 0}\left(g^{-1}(x)\right)^{i}\right) && \text{(by (1))}\\ &= \bigcup_{i\geq 0} g\left(\left(g^{-1}(x)\right)^{i}\right) && (g \text{ complete representation})\\ &= \bigcup_{i\geq 0}\left(g\left(g^{-1}(x)\right)\right)^{i} && (g \text{ homomorphism})\\ &= \bigcup_{i\geq 0} x^{i}. && (g \text{ one-to-one}) \end{aligned}$$

The last union computes the reflexive-transitive closure of the relation x, as was to be proved.

Finally, the mapping $f : \mathbf{A} \to \mathbf{B}$, $f(x) = g(h(x))$, is an isomorphism between the $\omega\mathsf{CFA}$ $\mathbf{A}$ and the PCFA $\mathbf{B}$.

5 Interpretability of P/PML in ωCCFA

In this section we will show how theories on P/PML can be interpreted as equational theories in $\omega\mathsf{CCFA}$. This is very useful because allows to reason equationally in a logic with variables over two different sorts (individuals and relations).

Definition 15. Let S, F and P be sets consisting of sort, function and relation symbols, respectively. By $\omega\mathsf{CCFA}^{+}(S, A, F, P)$ we denote the extension of $\omega\mathsf{CCFA}$ obtained by adding the following equations as axioms.

1. For each $s, s' \in S$ $(s \neq s')$, the equations $1\text{'}_{s} + 1\text{'}_{U} = 1\text{'}_{U}$ and $1\text{'}_{s} \cdot 1\text{'}_{s'} = 0$ (elements from types do not split, and different types are disjoint).
2. For each $a \in A$ with $ia(a) = s_1 \dots s_k$ and $oa(a) = s'_1 \dots s'_n$, the equation $\left(1\text{'}_{s_1} \otimes \cdots \otimes 1\text{'}_{s_k}\right);a;\left(1\text{'}_{s'_1} \otimes \cdots \otimes 1\text{'}_{s'_n}\right) = a$.

3. For each $f : s_1 \dots s_k \to s \in F$, $\breve{f};f + 1'_s = 1'_s$ and $(1'_{s_1} \otimes \cdots \otimes 1'_{s_k});f = f$, stating that f is a functional relation of the right sorts.
4. For each p of arity $s_1 \dots s_k$ in P, the equation $(1'_{s_1} \otimes \cdots \otimes 1'_{s_k});p;1 = p$, stating that p is a right-ideal relation expecting inputs of the right sorts.

Definition 16. A *model* for the calculus $\omega\mathsf{CCFA}^+(S, A, F, P)$ is a structure $\mathcal{A} = \langle \langle \mathbf{A}, S^{\mathcal{A}}, A^{\mathcal{A}}, F^{\mathcal{A}}, P^{\mathcal{A}} \rangle, m \rangle$ where: $\mathbf{A} \in \omega\mathsf{CFA}$. $S^{\mathcal{A}}$ is a set of disjoint partial identities, one for each sort symbol in S. $A^{\mathcal{A}}$ is a set of binary relations, one for each action symbol $a \in A$. Besides, if $ia(a) = s_1 \dots s_k$ and $oa(a) = s'_1 \dots s'_n$, then $a^{\mathcal{A}}$ satisfies the condition in item 2 of Def. 15. $F^{\mathcal{A}}$ is a set of functional relations, one for each function symbol in F. Besides, if $f : s_1 \dots s_k \to s$, then $f^{\mathcal{A}}$ satisfies the conditions in item 3 of Def. 15. $P^{\mathcal{A}}$ is a set of right ideal relations, one for each predicate symbol $p \in P$. Besides, if p has arity $s_1 \dots s_k$, then $p^{\mathcal{A}}$ satisfies the conditions in item 4 of Def. 15. $m : \mathit{RelVar} \to \mathbf{A}$.

Notice that the mapping m in a $\omega\mathsf{CCFA}^+(S, A, F, P)$ model extends homomorphically to arbitrary relational terms. For the sake of simplicity, we will use the same name for both.

In the following paragraphs we will define a function mapping formulas from $P/\mathit{PML}(S, A, F, P)$ to $\omega\mathsf{CCFA}^+(S, A, F, P)$ formulas. In the next definitions, σ is a sequence of numbers increasingly ordered. Intuitively, the sequence σ contains indices of those individual variables that appear free in the formula (or term) being translated. By $\mathit{Ord}(n, \sigma)$ we will denote the position of the index n in the sequence σ, by $[\sigma \oplus n]$ we denote the extension of the sequence σ with the index n, and by $\sigma(k)$ we denote the element in the k-th position of σ. In what follows, $t^{;n}$ is an abbreviation for $t; \cdots ;t$ (n times). For the sake of completeness, $t^{;0}$ is defined as 1'. We will denote by $\mathit{IndTerm}(F)$ the set of terms from P/PML built from the set of constant and function symbols F. By $\mathit{RelDes}(K)$ we denote the set of terms from $\omega\mathsf{CCFA}$ that are built from the set of relation constants K.

Definition 17. The function $\delta_\sigma : \mathit{IndTerm}(F) \to \mathit{RelDes}(F)$, mapping individual terms into relation designations, is defined inductively by the conditions:

1. $\delta_\sigma(v_i) = \begin{cases} \rho^{;\mathit{Ord}(i,\sigma)-1};\pi & \text{if } i \text{ is not the last index in } \sigma, \\ \rho^{;\mathit{Length}(\sigma)-1} & \text{if } i \text{ is the last index in } \sigma. \end{cases}$
2. $\delta_\sigma(f(t_1, \dots, t_m)) = (\delta_\sigma(t_1) \nabla \cdots \nabla \delta_\sigma(t_m));f$ for each $f \in F$.

Given a sequence σ such that $\mathit{Length}(\sigma) = l$ and an index n ($n < \omega$) such that v_n has sort s, we define the term $\Delta_{\sigma,n}$ ($n < \omega$) by the condition[6]

$$\Delta_{\sigma,n} = \begin{cases} \delta_\sigma(v_{\sigma(1)}) \nabla \cdots \nabla \delta_\sigma(v_{\sigma(k-1)}) \nabla 1_s \nabla \delta_\sigma(v_{\sigma(k+1)}) \nabla \cdots \nabla \delta_\sigma(v_{\sigma(l)}) \\ \qquad \text{if } k = \mathit{Ord}(n, [\sigma \oplus n]) < l, \\ \delta_\sigma(v_{\sigma(1)}) \nabla \cdots \nabla \delta_\sigma(v_{\sigma(l-1)}) \nabla 1_s \qquad \text{if } \mathit{Ord}(n, [\sigma \oplus n]) = l. \end{cases}$$

Notation 1 Let σ be a sequence of indices of individual variables of length n. Let $\vec{x} = \langle x_1, \dots, x_k \rangle$ be a vector of variables whose indices occur in σ. We will denote by $\Pi_{\sigma,\vec{x}}$ the relation that given a tuple of values for the variables whose indices appear in σ, projects the values corresponding to the variables appearing in $\vec{x}$. For example, given $\sigma = \langle 2, 5, 7, 9 \rangle$ and $\vec{x} = \langle v_2, v_7 \rangle$,

[6] By 1_s we denote the relation $1;1'_s$.

$\mathit{\Pi}_{\sigma,\vec{x}} = \{\, \langle a_1 \star a_2 \star a_3 \star a_4, a_1 \star a_3 \rangle : a_1, a_2, a_3, a_4 \in A \,\}$. Similarly, $\mathit{Arrange}_{\sigma,\vec{x}}$ denotes the relation that, given two tuples of values (one for the variables with indices in σ and the other for the variables in $\vec{x}$), produces a new tuple of values for the variables with indices in σ updating the old values with the values in the second tuple. For the previously defined σ and $\vec{x}$, we have $\mathit{Arrange}_{\sigma,\vec{x}} = \{\, \langle (a_1 \star a_2 \star a_3 \star a_4) \star (b_1 \star b_2), b_1 \star a_2 \star b_2 \star a_4 \rangle : a_1, a_2, a_3, a_4, b_1, b_2 \in A \,\}$. Note that these two relations can be easily defined using the projections π and ρ previously defined.

Definition 18. The mappings $M : \mathit{RT}(\mathcal{S}) \to \mathit{RelDes}(A)$ and $T_\sigma : \mathit{For}(\mathcal{S}) \to \mathit{RelDes}(A \cup F \cup P)$ are mutually defined by

$$\begin{array}{ll}
M(a) = a \text{ for each } a \in A \cup \mathit{RelVar}, & M(1'_{s_1 \ldots s_k}) = 1'_{s_1} \otimes \cdots \otimes 1'_{s_k}, \\
M(R^*) = M(R)^*, & M(R{+}S) = M(R){+}M(S), \\
M(R{\cdot}S) = M(R){\cdot}M(S), & M(R;S) = M(R);M(S), \\
M(\alpha?) = T_{\sigma_\alpha}(\alpha){\cdot}1', & \\
T_\sigma(p(t_1, \ldots, t_k)) = (\delta_\sigma(t_1) \nabla \cdots \nabla \delta_\sigma(t_k));p, & T_\sigma(\neg\alpha) = \overline{T_\sigma(\alpha)}, \\
T_\sigma((\exists v_n : s)\,\alpha) = \mathit{\Delta}_{\sigma,n};T_{[\sigma \oplus n]}(\alpha), & T_\sigma(\alpha \vee \beta) = T_\sigma(\alpha){+}T_\sigma(\beta),
\end{array}$$

$$T_\sigma\left(\left\langle \vec{x}\;{}_l R^u\;\vec{y} \right\rangle \alpha\right) =$$

$$\begin{pmatrix} 1' \\ \nabla \\ \mathit{\Pi}_{\sigma,\vec{x}};M(R) \end{pmatrix};\mathit{Arrange}_{\sigma,\vec{y}};T_\sigma(\alpha) \cdot \left((v_l;\, \leq) \cdot C_{\mathsf{l}(R)}\right);1 \cdot \left((v_u;\, \geq) \cdot C_{\mathsf{u}(R)}\right);1.$$

Notation 2 We will denote by:

- $\vdash_{\omega\mathsf{CCFA}}$ the provability relation in the calculus $\omega\mathsf{CCFA}$.
- $\models_{\mathrm{Full}}$ the validity relation on the class of full $\omega\mathsf{CCFA}$ models.
- $\models_{\omega\mathsf{CCFA}}$ the validity relation on the class of $\omega\mathsf{CCFA}$ models.

Notation 3 Given an object structure $\mathcal{A} = \langle\, \mathbf{S}, \mathbf{A}, \mathbf{F}, \mathbf{P} \,\rangle$, a valuation of the individual variables ν, $\mathbf{A} \in \mathsf{PCFA}$ such that $\mathit{Urel}_{\mathbf{A}} \supseteq \bigcup \mathbf{S}$, and a sequence of indices σ, by $s_{\nu,\sigma}$ we denote the element $a_1 \star \cdots \star a_i \star \cdots \star a_n \in U_{\mathbf{A}}$ such that:

1. $n = \mathit{Length}(\sigma)$,
2. $a_i = \nu(v_{\sigma(i)})$ for all i, $1 \leq i \leq n$.

In case $\alpha = \langle\rangle$, $s_{\nu,\sigma}$ denotes an arbitrary element from $U_{\mathbf{A}}$. Given a formula or term α, by σ_α we denote the sequence of indices of variables with free occurrences in α, sorted in increasing order.

Throughout the next theorems we will assume $\mathcal{S} = \langle\, A, \langle\, S, F, P \,\rangle \,\rangle$ is a fixed but arbitrary object signature.

Theorem 3. *Let $\alpha \in \mathit{For}(\mathcal{S})$, let $\sigma = \sigma_\alpha$, let $\mathcal{A}$ be an object structure for $\mathcal{S}$ and let μ be a valuation of the relational variables. Then there exists a model $\mathcal{B} = \langle\, \langle\, \mathbf{B}, S^{\mathcal{B}}, A^{\mathcal{B}}, F^{\mathcal{B}}, P^{\mathcal{B}} \,\rangle, m' \,\rangle$ for $\omega\mathsf{CCFA}^+(S, A, F, P)$ such that*

$$\mathcal{A} \models_{P/PML} \alpha[\nu][\mu] \qquad \Longleftrightarrow \qquad s_{\nu,\sigma} \in \mathsf{dom}\left(m'\left(T_\sigma(\alpha)\right)\right).$$

Proof. Assume $\mathcal{A} = \langle\, \mathbf{S}, \mathbf{A}, \mathbf{F}, \mathbf{P} \,\rangle$. Let us define $\mathcal{B}$ as follows.

1. Let $\mathbf{B}$ be the FullPCFA with set of urelements $\bigcup \mathbf{S}$,

2. For each sort $s \in S$, $1'_s = \{ \langle x, x \rangle : x \in \mathsf{s} \}$,
3. $a^{\mathcal{B}} = \{ \langle a_1 \star \cdots \star a_m, b_1 \star \cdots \star b_n \rangle : \langle \langle a_1, \ldots, a_m \rangle, \langle b_1, \ldots, b_n \rangle \rangle \in a^{\mathcal{A}} \}$, for each $a \in A$,
4. $f^{\mathcal{B}} = \{ \langle a_1 \star \cdots \star a_n, b \rangle : f^{\mathcal{A}}(a_1, \ldots, a_n) = b \}$, for each $f \in F$,
5. $p^{\mathcal{B}} = \{ \langle a_1 \star \cdots \star a_n, b \rangle : p^{\mathcal{A}}(a_1, \ldots, a_n)$ and $b \in U_{\mathbf{B}} \}$, for each $p \in P$,
6. $m'(R) = \{ \langle a_1 \star \cdots \star a_m, b_1 \star \cdots \star b_n \rangle : \langle \langle a_1, \ldots, a_m \rangle, \langle b_1, \ldots, b_n \rangle \rangle \in R^{\mathcal{A}}_{\mu} \}$ for all $R \in \mathit{RelVar}$.

The proof follows by simultaneous induction on the structure of terms from $RT(\mathcal{S})$ and formulas from $For(\mathcal{S})$.

Corollary 1. *Let $\alpha \in For(\mathcal{S})$ without free variables over individuals, let $\mathcal{A}$ be an object structure for $\mathcal{S}$, and let μ be a valuation of the relational variables. Then there exists a full model $\mathcal{B} = \langle \langle \mathbf{B}, S^{\mathcal{B}}, A^{\mathcal{B}}, F^{\mathcal{B}}, P^{\mathcal{B}} \rangle, m \rangle$ for $\omega\mathsf{CCFA}^+(S, A, F, P)$ such that*

$$\mathcal{A} \models_{P/PML} \alpha[\mu] \qquad \Longleftrightarrow \qquad m\left(T_{\langle\rangle}(\alpha)\right) = 1.$$

Notation 4 Given $\mathbf{A} \in \mathsf{PCFA}$, $s = a_1 \star \cdots \star a_k$ ($a_i \in \mathit{Urel}_{\mathbf{A}}$ for all i, $1 \le i \le k$) and a sequence σ of indices increasingly sorted and of length k, by $\nu_{s,\sigma}$ we denote the set of valuations of individual variables ν satisfying $\nu(v_{\sigma(i)}) = a_i$. In case $\sigma = \langle\rangle$, for each $s \in U_{\mathbf{A}}$ $\nu_{s,\sigma}$ denotes the set of all valuations. Given an array of variables $\overrightarrow{x}$ of length n whose indices occur in σ, $(a_1 \star \cdots \star a_k)_{\sigma,\overrightarrow{x}} = b_1 \star \cdots \star b_n$ where $b_i = a_{Ord(j,\sigma)}$, with j the index of the individual variables x_i. By $\mathsf{tuple}\,(a_1 \star \cdots \star a_k)$ we denote the tuple $\langle a_1, \ldots, a_k \rangle$.

Theorem 4. *Let $\alpha \in For(\mathcal{S})$, $\sigma = \sigma_\alpha$ and $\mathcal{A} = \langle \langle \mathbf{A}, S^{\mathcal{A}}, A^{\mathcal{A}}, F^{\mathcal{A}}, P^{\mathcal{A}} \rangle, m \rangle$ be a full $\omega\mathsf{CCFA}^+(S, A, F, P)$ model. Then there exists an object structure $\mathcal{B}$ and a valuation of relational variables μ such that*

$$s \in \mathsf{dom}\left(m\left(T_\sigma(\alpha)\right)\right) \qquad \Longleftrightarrow \qquad \mathcal{B} \models_{P/PML} \alpha[\nu][\mu] \text{ for all } \nu \in \nu_{s,\sigma}.$$

Proof. Since $\mathbf{A} \in \mathrm{Full}\mathsf{PCFA}$, for each sort s, let $\mathsf{s} = \{ a \in \mathit{Urel}_{\mathbf{A}} : a \in 1'_s \}$.

For each $a \in A$, define

$$a^{\mathcal{B}} = \{ \langle \langle a_1, \ldots, a_m \rangle, \langle b_1, \ldots, b_n \rangle \rangle : \langle a_1 \star \cdots \star a_m, b_1 \star \cdots \star b_n \rangle \in m(a) \}.$$

For each $f : s_1 \ldots s_k \to s \in F$,

$$f^{\mathcal{B}}(a_1, \ldots, a_k) = b \qquad \text{iff} \qquad \langle a_1 \star \cdots \star a_k, b \rangle \in f^{\mathcal{A}}.$$

For each p of arity $s_1 \ldots s_k$ in P,

$$\langle a_1, \ldots, a_k \rangle \in p^{\mathcal{B}} \qquad \text{iff} \qquad a_1 \star \cdots \star a_k \in \mathsf{dom}\left(p^{\mathcal{A}}\right).$$

For each $R \in \mathit{RelVar}$ with arity $\langle s_1 \ldots s_m, s'_1 \ldots s'_n \rangle$,

$$\mu(R) = \{ \langle \langle a_1, \ldots, a_m \rangle, \langle b_1, \ldots, b_n \rangle \rangle : \langle a_1 \star \cdots \star a_m, b_1 \star \cdots \star b_n \rangle \in m(R) \}.$$

The remaining part of the proof follows by induction on the structure of the formula α.

Corollary 2. *Let $\alpha \in For(\mathcal{S})$ without free variables over individuals and let $\mathcal{A} = \langle \langle \mathbf{A}, S^{\mathcal{A}}, A^{\mathcal{A}}, F^{\mathcal{A}}, P^{\mathcal{A}} \rangle, m \rangle$ be a full $\omega\mathsf{CCFA}^{+}(S, A, F, P)$ model. Then there exists an object structure $\mathcal{B}$ and a valuation of the relational variables μ such that*

$$\mathcal{B} \models_{P/PML} \alpha[\mu] \qquad \Longleftrightarrow \qquad m\left(T_{\langle\rangle}(\alpha)\right) = 1.$$

Theorem 5. *Let $\Gamma \cup \{\varphi\}$ be a set of $P/PML(S, A, F, P)$ formulas without variables over individuals. Then,*

$$\Gamma \models_{P/PML} \varphi \qquad \Longleftrightarrow \qquad \left\{ T_{\langle\rangle}(\gamma) = 1 : \gamma \in \Gamma \right\} \models_{\text{Full}} T_{\langle\rangle}(\varphi) = 1.$$

Proof. In order to prove the theorem, it suffices to show that

$$\Gamma \not\models_{P/PML} \varphi \qquad \Longleftrightarrow \qquad \left\{ T_{\langle\rangle}(\gamma) = 1 : \gamma \in \Gamma \right\} \not\models_{\text{Full}} T_{\langle\rangle}(\varphi) = 1. \qquad (2)$$

Formula (2) follows from Cors. 1 and 2.

Theorem 6. *Let $\mathcal{V}$ be the variety generated by* $\text{Full}\mathsf{PCFA}$. *Then, $\mathcal{V} = \omega\mathsf{CFA}$.*

Proof. From Thm. 2 and the fact $\mathsf{PCFA} = \mathbf{ISP}\text{Full}\mathsf{PCFA}$, $\omega\mathsf{CFA} = \mathbf{ISP}\text{Full}\mathsf{PCFA}$.

The next theorem states the interpretability of theories from P/PML as equational theories in $\omega\mathsf{CCFA}$.

Theorem 7. *Let $\Gamma \cup \{\varphi\}$ be a set of P/PML formulas without free individual variables. Then, $\Gamma \models_{P/PML} \varphi \iff \left\{ T_{\langle\rangle}(\gamma) = 1 : \gamma \in \Gamma \right\} \vdash_{\omega\mathsf{CCFA}} T_{\langle\rangle}(\varphi) = 1$.*

Proof. By Thm. 5,

$$\Gamma \models_{P/PML} \varphi \qquad \Longleftrightarrow \qquad \left\{ T_{\langle\rangle}(\gamma) = 1 : \gamma \in \Gamma \right\} \models_{\text{Full}} T_{\langle\rangle}(\varphi) = 1. \qquad (3)$$

By Thm. 6,

$$\begin{aligned} &\left\{ T_{\langle\rangle}(\gamma) = 1 : \gamma \in \Gamma \right\} \models_{\text{Full}} T_{\langle\rangle}(\varphi) = 1 \\ &\qquad\qquad \Longleftrightarrow \qquad \left\{ T_{\langle\rangle}(\gamma) = 1 : \gamma \in \Gamma \right\} \models_{\omega\mathsf{CCFA}} T_{\langle\rangle}(\varphi) = 1. \qquad (4) \end{aligned}$$

By the definition of the class $\omega\mathsf{CFA}$ and the definition of $\models_{\omega\mathsf{CCFA}}$,

$$\begin{aligned} &\left\{ T_{\langle\rangle}(\gamma) = 1 : \gamma \in \Gamma \right\} \models_{\omega\mathsf{CCFA}} T_{\langle\rangle}(\varphi) = 1 \\ &\qquad\qquad \Longleftrightarrow \qquad \left\{ T_{\langle\rangle}(\gamma) = 1 : \gamma \in \Gamma \right\} \vdash_{\omega\mathsf{CCFA}} T_{\langle\rangle}(\varphi) = 1. \qquad (5) \end{aligned}$$

Finally, by (3), (4) and (5),

$$\Gamma \models_{P/PML} \varphi \qquad \Longleftrightarrow \qquad \left\{ T_{\langle\rangle}(\gamma) = 1 : \gamma \in \Gamma \right\} \vdash_{\omega\mathsf{CCFA}} T_{\langle\rangle}(\varphi) = 1.$$

6 Conclusions

In [1] we presented a logic (P/PML) for formal real–time systems specification and construction. In this paper we presented the proof of interpretability of P/PML in an equational calculus ($\omega\mathsf{CCFA}$), thus enabling the use of equational inference tools in the process of systems construction. The calculus $\omega\mathsf{CCFA}$ defines the class of ω-closure fork algebras, and the proof of the representability theorem for ω-closure fork algebras hereby presented provides a very insightful and clean semantics based on binary relations for the calculus.

References

1. Baum, G.A., Frias, M.F. and Maibaum, T.S.E., *A Logic for Real–Time Systems Specification, Its Algebraic Semantics and Equational Calculus*, in Proceedings of AMAST'98, Springer–Verlag, LNCS 1548, pp. 91–105, 1999.
2. Carvalho, S.E.R., Fiadeiro, J.L. and Haeusler, E.H., *A Formal Approach to Real–Time Object Oriented Software*. In Proceedings of the Workshop on Real–Time Programming, pp. 91–96, sept/1997, Lyon, France, IFAP/IFIP.
3. Fenton, N. E., *Software Metrics. A Rigorous Approach*, International Thomson Computer Press, 1995.
4. Fiadeiro, J. L. L. and Maibaum, T. S. E., *Temporal Theories as Modularisation Units for Concurrent System Specifications*, Formal Aspects of Computing, Vol. 4, No. 3, (1992), 239–272.
5. Fiadeiro, J. L. L. and Maibaum, T. S. E., *A Mathematical Toolbox for the Software Architect*, in Proc. 8th International Workshop on Software Specification and Design, J. Kramer and A. Wolf, eds., (1995) (IEEE Press), 46–55.
6. Frias M. F., Baum G. A. and Haeberer A. M., *Fork Algebras in Algebra, Logic and Computer Science*, Fundamenta Informaticae Vol. 32 (1997), pp. 1–25.
7. Frias, M. F., Haeberer, A. M. and Veloso, P. A. S., *A Finite Axiomatization for Fork Algebras*, Logic Journal of the IGPL, Vol. 5, No. 3, 311–319, 1997.
8. Frias, M. F. and Orlowska, E., *A Proof System for Fork Algebras and its Applications to Reasoning in Logics Based on Intuitionism*, Logique et Analyse, vol. 150–151–152, pp. 239–284, 1995.
9. Frias, M. F. and Orlowska, E., *Equational Reasoning in Non–Classical Logics*, Journal of Applied Non Classical Logic, Vol. 8, No. 1–2, 1998.
10. Gries, D., *Equational logic as a tool*, LNCS 936, Springer–Verlag, 1995, pp. 1-17.
11. Gries, D. and Schneider, F. B., *A Logical Approach to Discrete Math.*, Springer–Verlag, 1993.
12. Kaposi, A. and Myers, M., *Systems, Models and Measures*, Springer–Verlag London, Formal Approaches to Computing and Information Technology, 1994.
13. Lyndon, R., *The Representation of Relational Algebras*, Annals of Mathematics (Series 2) vol.51 (1950), 707–729.
14. Maddux, R.D., *Topics in Relation Algebras*, Doctoral Dissertation, Univ. California, Berkeley, 1978, pp. iii+241.
15. Maddux, R.D., *Finitary Algebraic Logic*, Zeitschr. f. math. Logik und Grundlagen d. Math. vol. 35, pp. 321–332, 1989.
16. Maddux, R.D., *Pair-Dense Relation Algebras*, Transactions of the A.M.S., vol. 328, No. 1, pp. 83–131, 1991.
17. Maddux, R.D., *Relation Algebras*, Chapter 2 of Relational Methods in Computer Science, Springer Wien New York, 1997.
18. Roberts, F. S., *Measurement Theory, with Applications to Decision–Making, Utility and the Social Sciences*, Addison–Wesley, 1979.
19. Tarski, A. and Givant, S.,*A Formalization of Set Theory without Variables*, A.M.S. Coll. Pub., vol. 41, 1987.