Fundamenta Informaticae 32 (1997) 1-25
IOS Press

Fork Algebras in Algebra, Logic and Computer Science

Marcelo F. Frias
Laboratório de Métodos Formais,
Departamento de Informática,
Pontif'ıcia Universidade Católica do Rio de Janeiro,
Rua Marquês de São Vicente 225, 22453-900,
Rio de Janeiro, RJ, Brazil.
e-mail:mfrias@inf.puc-rio.br

Gabriel A. Baum
Universidad Nacional de La Plata,
LIFIA, Departamento de Informática,
C.C.11, Correo Central, 1900, La Plata,
Provincia de Buenos Aires, República Argentina.
e-mail:gbaum@info.unlp.edu.ar

Armando M. Haeberer
Laboratório de Métodos Formais,
Departamento de Informática,
Pontif'ıcia Universidade Católica do Rio de Janeiro,
Rua Marquês de São Vicente 225, 22453-900,
Rio de Janeiro, RJ, Brazil.
e-mail:armando@inf.puc-rio.br

Abstract. Since the main themes at the Helena Rasiowa memorial were algebra, logic and computer science, we will present a survey of results on fork algebras from these points of view. In this paper we study fork algebras from the points of view of their algebraic and logical properties and applications. These results will prove to be essential, in a future work, for the definition of a wide–spectrum calculus for program construction.

1. Introduction

Fork algebras are algebraic structures that appeared when looking for a relational calculus suitable for program specification and construction [4, 13, 22, 23]. This calculus should fulfill some requirements, as for example

- being expressive enough so that a wide variety of interesting problems for computer science can be specified in this calculus,

- being adequate with respect to the process of program construction, i.e., derivation rules must yield algorithms that are formally correct regarding their specifications,
- being simple enough so that the amount of mathematical or logical knowledge necessary in order to derive algorithms is minimal,
- being well behaved with respect to properties of the problem domain, i.e., properties of the problem should translate in a natural way into program derivation steps,
- and being wide enough so that not only the problems can be specified, but also strategies for solving these problems can be specified too.

At this point the reader may be asking himself "what is the need for a relational calculus, when functional calculi for program construction have existed for years?". The answer is twofolded. First, specifying a problem in a relational framework can lead to more declarative and simpler specifications. Second, nondeterministic problems can be specified naturally, as well as parallel implementations can be developed for these problems.

It was when looking for a programming calculus of this kind that a more theoretical research became mandatory. Analyzing the expressiveness of the calculus led to a result important in itself, as it is the possibility of algebraizing classical first-order logic with equality (FOLE) with this calculus. This result led later on to relational algebraizations of a wide variety of non-classical logics. Also, the study of the portability of properties of the problem domain into the calculus led to the representation theorem for fork algebras in terms of a class of algebras of binary relations (proper fork algebras). This result implies (as will be shown in a future paper) that the calculus is of a great heuristic power, since properties that are valid in the problem domain are suitable to be proved syntactically within the calculus.

The structure of the paper is as follows. In Section 2 we will present the classes of proper and abstract fork algebras and the representation theorem. In Section 3 we will present results concerning the expressiveness of fork algebras and algebraization of classical and non-classical logics, presenting for example a Rasiowa-Sikorski–like calculus for a formalism based on fork algebras. In Section 4 we will describe ongoing research and further work, and finally, in Section 5, we will present the conclusions derived from this research.

2. Fork Algebras in Algebra

In this section we will first define the classes of algebras of binary relations and their abstract counterpart, the class of relation algebras. The study about algebras of binary relations began with the works of Charles Sanders Peirce [47] and Augustus De Morgan [9] and was later on continued by Ernst Schröder [50] when looking for an algebraic counterpart of first-order reasoning, much the same as George Boole developed the now called Boolean algebras as a counterpart to propositional reasoning.

Given a binary relation X in a set A, and $a, b \in A$, we will denote the fact that a and b are related via the relation X by $\langle a, b\rangle \in X$ or aXb.

Definition 2.1. Let E be a binary relation on a set A, and let R be a set of binary relations satisfying:

1. $\bigcup R \subseteq E$,
2. Let Id denote the identity relation on the set A. Then $\emptyset$, E and Id belong to R,
3. R is closed under set union ($\cup$), intersection ($\cap$) and complement relative to E ($^-$),
4. R is closed under relational composition (denoted by $|$) and converse (denoted by $^\smile$). These two operations are defined by

$$X|Y = \{\, \langle a, b\rangle : \exists c \text{ such that } aXc \ \wedge \ cYb \,\}$$

$$(X)^\smile = \{\, \langle a, b\rangle : bXa \,\}.$$

Then, the structure $\langle R, \cup, \cap, \bar{}, \emptyset, E, |, Id, \breve{} \rangle$ is called an *algebra of binary relations*. We will denote the class of algebras of binary relations by ABR.

In 1941 Alfred Tarski introduced [51] the elementary theory of binary relations (ETBR) as a logical formalization of the theory of binary relations. The formalism ETBR is a formal theory where two different sorts of variables are present. The set $IndVar = \{ v_1, v_2, v_3, \dots \}$ contains the so-called *individual variables*, and the set $RelVar = \{ R, S, T, \dots \}$ contains the so-called *relation variables*. If we add the *relation constants* 0, 1 and 1' to the relation variables and close this set under the unary operators $\bar{}$ and $\breve{}$, and the binary operators $+$, $\cdot$ and ;, we obtain the set of *relation designations*. Examples of such objects are for instance $\breve{R}$ (to be read "the *converse* of R") and $R;S$ (to be read "the *relative product* – or Peircean product – of R and S"). Atomic formulas are expressions of the form xRy (where x, y are arbitrary individual variables and R is an arbitrary relation designation) or $R \equiv S$ (with R and S arbitrary relation designations). From the atomic formulas we obtain compound formulas as usual, by closing the atomic formulas under the unary logical constants $\neg$, $\forall x$, $\forall y, \dots$, $\exists x$, $\exists y \dots$ (x, y, $\dots$ individual variables) and the binary logical constants $\vee$, $\wedge$, $\Rightarrow$, $\Leftrightarrow$. We will choose a standard set of logical axioms and inference rules (see for example [12]). As the axioms that explain the meaning of the relational symbols 0, 1, 1', $\bar{}$, $\breve{}$, $+$, $\cdot$ and ;, we single out the following formulas, in which x, y, z are arbitrary individual variables and R, S, T are arbitrary relation designations.

$$\forall x \forall y \, (x \, 1 \, y) \qquad \forall x \forall y \, (\neg x \, 0 \, y)$$
$$\forall x \, (x \, 1' \, x) \qquad \forall x \forall y \forall z \, ((xRy \ \wedge \ y \, 1' \, z) \Rightarrow xRz)$$
$$\forall x \forall y \, (x \overline{R} y \ \Leftrightarrow \ \neg xRy) \qquad \forall x \forall y \left(x \breve{R} y \ \Leftrightarrow \ yRx\right)$$
$$\forall x \forall y \, (xR{+}Sy \ \Leftrightarrow \ xRy \vee xSy) \qquad \forall x \forall y \, (xR{\cdot}Sy \ \Leftrightarrow \ xRy \wedge xSy)$$
$$\forall x \forall y \, (xR;Sy \ \Leftrightarrow \ \exists z \, (xRz \wedge zSy)) \qquad R \equiv S \ \Leftrightarrow \ \forall x \forall y \, (xRy \Leftrightarrow xSy)$$

In further work we will use an extension of the ETBR as a language for program specification.

From ETBR Tarski introduced in [51] the *calculus of relations* (CR). The calculus of relations is defined as a restriction of ETBR. Formulas of CR are those formulas of ETBR where no variables over individuals occur. As axioms of CR Tarski singled out a subset of formulas without variables over individuals valid in ETBR. The formulas Tarski chose as axioms are, besides a set of axioms for the logical constants, the following formulas.

1. $(R \equiv S \wedge R \equiv T) \ \Rightarrow \ S \equiv T$
2. $R \equiv S \ \Rightarrow \ (R{+}T \equiv S{+}T \ \wedge \ R{\cdot}T \equiv S{\cdot}T)$
3. $R{+}S \equiv S{+}R \ \wedge \ R{\cdot}S \equiv S{\cdot}R$
4. $(R{+}S){\cdot}T \equiv (R{\cdot}T) + (S{\cdot}T) \ \wedge \ (R{\cdot}S) {+} T \equiv (R{+}T) \cdot (S{+}T)$
5. $R{+}0 \equiv R \ \wedge \ R{\cdot}1 \equiv R$
6. $R{+}\overline{R} \equiv 1 \ \wedge \ R{\cdot}\overline{R} \equiv 0$
7. $\overline{1} \equiv 0$
8. $\breve{R} \equiv R$
9. $(R;S)\breve{} \equiv \breve{S};\breve{R}$
10. $(R;S) \, ;T \equiv R; \, (S;T)$
11. $R;1' \equiv R$
12. $(R;S) \cdot T \equiv 0 \ \Rightarrow \ (S;T) \cdot \breve{R} \equiv 0$
13. $R;1 \equiv 1 \ \vee \ 1;\overline{R} \equiv 1$

Axioms (1)–(7) are an axiomatization for Boolean algebras, axioms (8)–(12) axiomatize the relative operators. Finally, formula (13) forces models to be simple (see [30]). The models of the CR motivate the following definition.

Definition 2.2. A *relation algebra* (RA for short) is an algebraic structure of type $\langle A, +, \cdot, \bar{}, 0, 1, ;, 1', \breve{} \rangle$, where $+, \cdot$ and $;$ are binary operations, $\bar{}$ and $\breve{}$ are unary, and $0, 1$ and $1'$ are distinguished elements. Furthermore, the reduct $\langle A, +, \cdot, \bar{}, 0, 1 \rangle$ is a Boolean algebra, and the following identities are satisfied for all $x, y, z \in A$:

$$x;(y;z) = (x;y);z, \tag{Ax. 1}$$

$$(x+y);z = x;z + y;z, \tag{Ax. 2}$$

$$(x+y)\breve{} = \breve{x}+\breve{y}, \tag{Ax. 3}$$

$$\breve{\breve{x}} = x, \tag{Ax. 4}$$

$$x;1' = 1';x = x, \tag{Ax. 5}$$

$$(x;y)\breve{} = \breve{y};\breve{x}, \tag{Ax. 6}$$

$$x;y \cdot z = 0 \quad \text{iff} \quad z;\breve{y} \cdot x = 0 \quad \text{iff} \quad \breve{x};z \cdot y = 0. \tag{Ax. 7}$$

If we add formula (13) to the axiomatization of RA, we obtain the class of simple relation algebras. It is proved in [8] that axioms (1)–(12) are provable in RA and viceversa. We will denote by $\leq$ the ordering induced by the Boolean structure.

As an immediate consequence of Defs. 2.1 and 2.2 we obtain the following theorem.

Theorem 2.3. Every algebra of binary relations is a relation algebra.

At the end of his paper [51], Tarski asked the following questions.

1. Is every model of CR isomorphic to an algebra of binary relations? (representability)
2. Is it true that every formula of the calculus of relations that is valid in ABR is provable in CR?
3. Is it true that every formula of ETBR can be transformed into an equivalent formula of CR?

The answer to these questions is in all cases "No". Question 1 was answered negatively by Roger Lyndon [33, 34] by exhibiting a non-representable relation algebra. Question 2 was answered by McKenzie [36] by presenting a formula of CR (an equation as a matter of fact) valid in ABR but that fails in a non-representable relation algebra. Regarding question 3, a result due to Korselt and whose proof is included in [32] shows that the expressive power of CR is that of a proper restriction of first-order logic (see [53] Ch. 3.4 for a complete discussion). In [51] Tarski presented the following formula, which is not equivalent to any sentence of CR.

$$\forall x \forall y \forall z \exists u \, (uRx \ \wedge \ uRy \ \wedge \ uRz). \tag{1}$$

If we attempt to eliminate the individual variables from formula (1), we can proceed as follows. By using the definition of the converse in ETBR, we can transform formula (1) into

$$\forall x \forall y \forall z \exists u \left(x\breve{R}u \ \wedge \ uRy \ \wedge \ uRz \right). \tag{2}$$

If we now try to eliminate the existential quantifier $\exists u$ from formula (2) by using the definition of the relative product in ETBR, this will leave us with a free occurrence of the individual variable u

$$\forall x \forall y \forall z \left(x\breve{R};Ry \ \wedge \ uRz \right), \tag{3}$$

and leave us also with a formula not equivalent to the original one.

In order to overcome this problem, it seems enough to have some operator ∇ that allows to perform the following transformation

$$uRx \ \wedge \ uRy \qquad \Longleftrightarrow \qquad uR\nabla R\,\langle x,y\rangle\,. \tag{4}$$

If such operator is available, we can then proceed as follows.

$$\begin{aligned}
\forall x\forall y\forall z\exists u\,(uRx \ \wedge \ uRy \ \wedge \ uRz) &\Longleftrightarrow \forall x\forall y\forall z\exists u\,(uR\nabla R\,\langle x,y\rangle \ \wedge \ uRz)\\
&\Longleftrightarrow \forall x\forall y\forall z\exists u\,(\langle x,y\rangle\,(R\nabla R)^{\smile}u \ \wedge \ uRz)\\
&\Longleftrightarrow \forall x\forall y\forall z\,(\langle x,y\rangle\,(R\nabla R)^{\smile};Rz)\\
&\Longleftrightarrow \forall x\forall y\forall z\left(z\check{R};\,(R\nabla R)\,\langle x,y\rangle\right)\\
&\Longleftrightarrow \check{R};\,(R\nabla R)\equiv 1\nabla 1.
\end{aligned}$$

We can define the operator ∇ by the following formula from ETBR^1.

$$\forall x\forall y\,(xR\nabla Sy \ \Longleftrightarrow \ \exists u\exists v\,(y=\langle u,v\rangle \ \wedge \ xRu \ \wedge \ xSv))\,. \tag{5}$$

It was proved by Mikulás, Sain and Simon [37] that the class of ABR extended with an operator ∇ defined as in (5) is not finitely axiomatizable with formulas from CR. However, it was proved in [41, 31] that this result is only a consequence of the axiom of foundation we use nowadays. Namely, in a non-well-founded version of ZFC with (a weak version of) Boffa's anti foundation axiom (BAFA), ABR with ∇ as defined in (5) is in fact finitely axiomatizable. Furthermore, as was shown in [31], this change of set theories does not affect the ordinary theorems of mathematics, e.g., the isomorphism invariant theorems of universal algebra. Since changing set theories is not a solution for the negative result from [37] when aiming to provide a foundation for a calculus for program construction, we will use the following alternative definition

$$\forall x\forall y\,(xR\nabla_{\star}Sy \ \Longleftrightarrow \ \exists u\exists v\,(y=\star\,(u,v) \ \wedge \ xRu \ \wedge \ xSv))\,, \tag{6}$$

where $\star$ is an arbitrary injective mapping, i.e., $\star$ satisfies the sentence

$$\forall x\forall y\forall u\forall v\,(\star(x,y)=\star(u,v) \ \Rightarrow \ x=u \ \wedge \ y=v)\,. \tag{7}$$

We will call the operator $\nabla_{\star}$ defined in (6) *fork*.

In the following subsections we will get into the technical details in order to define the classes of proper (also called standard) fork algebras and abstract fork algebras, as well as prove the representation theorem.

2.1. Proper and Abstract Fork Algebras

In order to define the class of proper fork algebras (denoted by PFA), we will first define the class of Full$\star$PFA as follows.

Definition 2.4. A Full$\star$PFA is a two sorted structure with domains[2] $\mathcal{P}\,(U\times U)$ and U

$$\langle\,\mathcal{P}\,(U\times U)\,,U,\cup,\cap,^{-},\emptyset,U\times U,|,Id,^{\smile},\underline{\nabla},*\,\rangle$$

such that

[1] For more results concerning algebras of binary relations extended with a fork operator as the one defined in (5), see [29, 38].

[2] By $\mathcal{P}\,(A)$ we denote the power set of a set A.

1. $|$, Id, $\breve{}$ and $\bar{}$ stand for composition between binary relations, the diagonal relation on U, the converse of binary relations, and set complementation w.r.t. $U \times U$, respectively. Thus, the reduct to the substructure $\langle \mathcal{P}(U \times U), \cup, \cap, \bar{}, \emptyset, U \times U, |, Id, \breve{} \rangle$ is an ABR,
2. $* : U \times U \to U$ is an injective function,
3. $R \underline{\nabla} S = \{ \langle x, *(y,z) \rangle : x R y \text{ and } x S z \}$.

Definition 2.5. We define FullPFA as RdFull$*$PFA where Rd takes reducts to the similarity type $\langle \cup, \cap, \bar{}, \emptyset, U \times U, |, Id, \breve{}, \underline{\nabla} \rangle$, and define the class PFA as S P FullPFA, where S takes subalgebras and P closes a class under direct product.

Definition 2.6. Notice that according to Def. 2.4 each $\mathfrak{A} \in$ PFA contains a set U on which the binary relations are defined. This set will be called the *base* of $\mathfrak{A}$, and will be denoted by U .

Notation 2.7. By $Dom(R)$ we denote the term $\left(R;\breve{R}\right) \cdot 1'$ and by $Ran(R)$ we denote the term $\left(\breve{R};R\right) \cdot 1'$.

A relation R is called *functional* if $\breve{R};R \leq 1'$.

In Defs. 2.4 and 2.5, the function $*$ performs the role of pairing, encoding pairs of objects into single objects. It is important to notice that there are $*$ functions which are distinct from set-theoretical pair formation, i.e., $*(x,y)$ differs from $\{ x, \{ x, y \} \}$.

Notice that in order to define a FullPFA it suffices to provide the set U and an injective mapping $* : U \times U \to U$.

Given $\mathfrak{A} \in$ PFA with base U , it is possible to single out those elements that do not represent pairs (if there are any). Notice that the term $\overline{1 \nabla 1}$ stands for the binary relation $\{ \langle x, y \rangle : x \in U \ \wedge \ \forall u, v \in U \ (y \neq *(u,v)) \}$. Thus, the term $Ran\left(\overline{1 \nabla 1}\right)$ distinguishes those objects from the base that are not pairs. In what follows we will denote by $1'_{\mathsf{U}}$ the term $Ran\left(\overline{1 \nabla 1}\right)$, by 1_{U} the term $1;1'_{\mathsf{U}}$ and by ${}_{\mathsf{U}}1$ the term $1';1'_{\mathsf{U}};1$. We will call the elements from the base in the domain of $1'_{\mathsf{U}}$ *urelements*, and will denote the set of urelements in a fork algebra $\mathfrak{A}$ by $Urel$.

Under the previous definitions, the equation

$$1;1'_{\mathsf{U}};1 \equiv 1 \tag{8}$$

is valid in a $\mathfrak{A} \in$ PFA only in case $Urel$ is nonempty. We will denote by PFAU the class of those algebras in PFA with a nonempty set of urelements. Also, we denote by SPFAU the class of simple proper fork algebras with urelements. It is easy to show that SPFAU = S FullPFAU.

Given a pair of binary relations, the operation called *cross* performs a kind of parallel product. A graphic representation of cross is given in Fig. 1. Its set theoretical definition is given by

$$R \otimes S = \{ \langle *(x,y), *(w,z) \rangle : x R w \wedge y S z \} .$$

It is not difficult to check that cross is definable from the other relational operators with the use of fork. It is a simple exercise to show that

$$R \otimes S = ((Id \underline{\nabla} (U \times U))\breve{} \mid R) \ \underline{\nabla} \ (((U \times U) \underline{\nabla} Id)\breve{} \mid S) .$$

By $1'_{\mathsf{U}}{}^{k}$ we will denote the term $\underbrace{1'_{\mathsf{U}} \otimes \cdots \otimes 1'_{\mathsf{U}}}_{k \text{ times}}$.

Much the same as relation algebras [30, 35, 51] are an abstract version of algebras of binary relations, proper fork algebras have also their abstract counterpart. The class of abstract fork algebras (denoted by AFA) is a finitely based variety, i.e., its axiomatization is given by a finite set of equations.

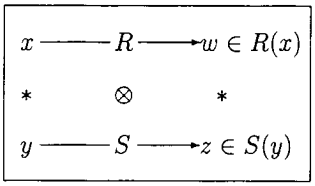

Figure 1 The operator *cross*.

Definition 2.8. An abstract fork algebra is an algebraic structure

$$\langle R, +, \cdot, ^-, 0, 1, ;, 1', ^\smile, \nabla \rangle$$

satisfying the following set of axioms

Axioms stating that the reduct $\langle R, +, \cdot, ^-, 0, 1, ;, 1', ^\smile \rangle$ is a relation algebra in which $\langle R, +, \cdot, ^-, 0, 1 \rangle$ is the Boolean reduct (where $\leq$ denotes the induced partial ordering), $\langle R, ;, 1' \rangle$ is the monoid reduct, and $^\smile$ stands for relational converse,

$$r \nabla s = (r; (1' \nabla 1)) \cdot (s; (1 \nabla 1')), \tag{Ax. 8}$$

$$(r \nabla s)\,;(t \nabla q)^\smile = \left(r; \breve{t}\right) \cdot \left(s; \breve{q}\right), \tag{Ax. 9}$$

$$(1' \nabla 1)^\smile \nabla (1 \nabla 1')^\smile + 1' = 1'. \tag{Ax. 10}$$

Once we have defined AFA we define AFAU (the class of abstract fork algebras with urelements) as the class of algebras from AFA that satisfy equation (8). We also define SAFAU (the class of simple abstract fork algebras with urelements) as those algebras from AFAU that satisfy formula 13 from the axiomatization of CR.

From the abstract definition of fork induced by the axioms in Def. 2.8, it is possible to define cross by the equation

$$R \otimes S \equiv ((1' \nabla 1)^\smile ; R) \ \nabla \ ((1 \nabla 1')^\smile ; S).$$

2.2. Representability

If we recall the second problem posed by Tarski on the relationship between relation algebras and algebras of binary relations, the conclusion (proved by Roger Lyndon in [33]) was that not every relation algebra is isomorphic to an algebra of binary relations. In this subsection we will show that the answer to the same problem is positive when fork algebras are considered instead of relation algebras. A first proof of this representation theorem for complete and atomistic abstract fork algebras is given in [14, 15]. Later on Gyuris [21] and Frias et al. [16] proved a representation theorem for the whole class by using a result developed by Tarski on the representability of quasi-projective relation algebras [52]. Strictly speaking, since proper fork algebras are not a concrete class because of the hidden operation $*$, this can be considered as a weak-representation theorem[3]. Anyway, the theorem provides all the machinery we need for developing our theory. A strong representation theorem for fork algebras was proved in [31, 40, 41, 42], although in non-well founded set theories.

Definition 2.9. We say that a class of algebras $\mathcal{K}_1$ is representable in a class of algebras $\mathcal{K}_2$ if for every algebra $\mathfrak{A} \in \mathcal{K}_1$ there exists an algebra $\mathfrak{B} \in \mathcal{K}_2$ such that $\mathfrak{A}$ is isomorphic to $\mathfrak{B}$.

[3]This was pointed out by Andréka and Németi in a private communication.

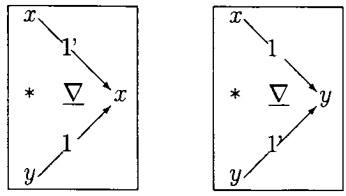

Figure 2 The projections π and ρ.

We will distinguish two constants π and ρ that behave in the standard models as projections. We define $\pi := (1' \nabla 1)^{\smile}$ and $\rho := (1 \nabla 1')^{\smile}$ (see Fig. 2). The following properties of the relations π and ρ will be useful in the proof of the representation theorem. They establish that the relations π and ρ satisfy the definition of quasi-projections as given in [53] p. 96. Thus, we call π the *first quasi-projection* and ρ the *second quasi-projection* of an abstract fork algebra.

Lemma 2.10. The relations π and ρ are functional. Moreover, the equation $\breve{\pi}; \rho \equiv 1$ holds in every abstract fork algebra.

If we recall that quasi-projective relation algebras [53] are relation algebras containing a couple of relations satisfying the properties annunciated in Lemma 2.10, we immediately obtain the following proposition.

Proposition 2.11. The relation algebra reduct of any AFA is a quasi-projective relation algebra.

Theorem 2.12. AFA is representable in PFA.

Proof:
We will present here a sketch of the proof. The complete proof is given in [16].

From Prop. 2.11, the relation algebra reduct of a fork algebra is a quasi-projective relation algebra. Thus, by Thm. 8.4(iii), p. 242 of [53], the relation algebra reduct of a fork algebra is a representable relation algebra.

Let $\mathfrak{A} \in$ AFA, and let us denote by $\mathfrak{B}$ its relation algebra reduct. Let $\mathfrak{C}$ be an algebra of binary relations, and let $h : \mathfrak{B} \to \mathfrak{C}$ be a relation algebra isomorphism.

Let π and ρ be the projections in $\mathfrak{A}$, and let $p := h(\pi)$ and $q := h(\rho)$. Let $*$ be defined by

$$*(a, b) = c \iff \langle a, c \rangle \in \breve{p} \text{ and } \langle c, b \rangle \in q.$$

It is not difficult to prove (using the AFA axioms) that $*$, as defined, is an injective function.

Let $\mathfrak{D} = \langle C, \underline{\nabla} \rangle$ be an extension of $\mathfrak{C}$ in which the operation $\underline{\nabla}$ is defined by the formula

$$R \underline{\nabla} S = \{ \langle x, *(y, z) \rangle : x R y \ \wedge \ x S z \}.$$

As a final step, we prove that

$$h(R \nabla S) = h(R) \underline{\nabla} h(S).$$

3. Fork Algebras in Logic

Tarski defined the formalism ETBR from the algebras of binary relations. In a similar way we will define an elementary theory of fork relations (ETFR for short) from the class of proper fork algebras.

Definition 3.1. Given a set of constant symbols C and a set of function symbols F with arity, the set of *individual terms* (denoted by $IndTerm(C,F)$) is the smallest set A satisfying:

1. $IndVar \cup C \subseteq A$,
2. If $f \in F$ has arity k and $t_1, \ldots, t_k \in A$, then $f(t_1, \ldots, t_k) \in A$,

Definition 3.2. Given a set of constant relation symbols P, the set of *relation designations* on P (denoted by $RelDes(P)$) is the smallest set A satisfying:

1. $RelVar \cup \{ 0, 1, 1' \} \cup P \subseteq A$,
2. If $R \in A$, then $\left\{ \overline{R}, \breve{R} \right\} \subseteq A$,
3. If $R, S \in A$, then $\{ R+S, R \cdot S, R;S, R \nabla S \} \subseteq A$.

Definition 3.3. Let u be a symbol, then $A^\star$ is the smallest set satisfying (a) $u \in A^\star$, (b) if $x, y \in A^\star$, then the expression $\star(x,y)$ belongs also to $A^\star$. Elements of $A^\star$ are called arities. The arity $\star(u, \star(u, \cdots))$ with k occurrences of u will be denoted by the number k. We will in general write $u \star \cdots \star u$ (k occurrences of u) instead of $\star(u, \star(u, \cdots))$. For example, we have $4 = \star(u, \star(u, \star(u,u))) = u \star u \star u \star u$.

Definition 3.4. Given a set of constant symbols C and a set of function symbols F, by $IndTerm(C,F)^\star$ we denote the smallest set A satisfying:

1. $IndTerm(C,F) \subseteq IndTerm(C,F)^\star$,
2. If $t_1, t_2 \in IndTerm(C,F)^\star$, then $\star(t_1,t_2) \in IndTerm(C,F)^\star$.

Definition 3.5. Given $t \in IndTerm(C,F)^\star$ the arity of t (denoted by $arity(t)$) is inductively defined as follows:

1. If $t \in IndTerm(C,F)$, then $arity(t) = u$,
2. if $t \in IndTerm(C,F)^\star$ and $t = \star(t_1,t_2)$, then

$$arity(t) = \star(arity(t_1), arity(t_2)).$$

Definition 3.6. Given a set of constant symbols C, a set of function symbols F and a set of constant relation symbols P the set of atomic formulas of ETFR is the smallest set A satisfying:

1. $R \equiv S \in A$ whenever $R, S \in RelDes(P)$,
2. $t_1 R t_2 \in A$ whenever $t_1, t_2 \in IndTerm(C,F)^\star$ and $R \in RelDes(P)$.

From the atomic formulas, compound formulas are built up as in first-order logic, with quantifiers applied to individual variables only. Notice that once the sets C, F and P are fixed, a unique set of formulas is characterized. We will denote this set by $ForETFR(C,F,P)$.

Definition 3.7. Given a set of constant symbols C, a set of function symbols F and a set of relation constant symbols P, we define the formalism ETFR(C,F,P) as follows:

Formulas: We choose the set $ForETFR(C,F,P)$.
Inference rules: Same as in ETBR.
Axioms: Extend the axioms of ETBR by adding formulas (6) and (7).

Much the same as Tarski defined his calculus of relation from the elementary theory of binary relations, we will define a calculus of fork relations (CFR) from the elementary theory of fork relations.

Definition 3.8. Given a set of relation constant symbols P, we define the formalism CFR(P) as follows:

Formulas: Those formulas from ETFR in which neither individual variables nor constant symbols occur.
Inference rules: Same as in CR.
Axioms: Extend the axioms of CR by adding formulas (8)–(10) from Def. 2.8.

In this section we will analyze the expressive power of CFR. When using fork algebras as a formalism for program construction we will use FOLE and ETFR as program specification languages, and CFR as a calculus for program construction and design. For this to be possible, it is required that CFR be at least as expressive as FOLE and ETFR. In Subsection 3.1 we will show how to carry this program on. As a consequence of these results, in Subsection 3.2 we will show how to algebraize non-classical logics.

For this and the remaining sections, given sets C,F and P we will denote by FOLE(C, F, P) the first-order logic on the language with set of constant symbols C, set of function symbols F and set of predicate symbols P. By $\equiv$ we will denote the equality predicate of FOLE.

An example of a formula from ETFR($\{4\}, \{\sqrt{}, Plus\}, \{\leq\}$) is the formula

$$\forall x \exists y \left(\sqrt{Plus(x, 4)} \leq y \right).$$

The algebraization of logics is a field of extensive and active work. In the remaining part of this section we will show how fork algebras can be used for algebraizing classical first-order logic, as well as many non classical logics. The reader interested in the algebraization of logics should consider reading the books [27] (in particular Section 4.3, which studies the connections between cylindric algebras and logic, and Ch. 5, in which other algebraizations are presented), and [25]. Also fundamental are the works of the Budapest school, specially the papers [1, 2, 3, 39]. Finally, the work of Blok and Pigozzi (see [5] and the references therein) is a very valuable source of results in algebraic logic.

3.1. Algebraization of Classical First-Order Logic

In order to fulfill the task of algebraizing FOLE, we will perform an intermediate step. We will first show how to interpret FOLE in ETFR, and after doing this we will show how to interpret ETFR in CFR.

Notice that in FOLE(C, F, P) there is a standard notion of *arity* for function and predicate symbols. We will also assume that function and constant relation symbols from ETFR(C, F, P) have an arity associated. Arity of functions is defined as usual. For constant relation symbols the arity is defined as a pair $\langle a_1, a_2 \rangle$ where $a_1, a_2 \in A^\star$. a_1 is called the *input arity* and a_2 is called the *output arity*. The reason to do this is that later on we will convert first-order predicates into input-output binary relations. To this syntactic definition of arity corresponds also a semantic notion.

Definition 3.9. Let $\mathfrak{A} \in$ PFAU. Given $e \in U$ and an arity a, we say that e has arity a (denoted $arity(e) = a$), whenever:

1. $e \in Urel$ and $a = u$,
2. $e = *(e_1, e_2)$ and $a = \star(a_1, a_2)$ and $arity(e_1) = a_1$ and $arity(e_2) = a_2$.

Definition 3.10. Given a PFAU $\mathfrak{A}$, a binary relation $R \in \mathfrak{A}$ has arity $\langle a_1, a_2 \rangle$ ($a_1, a_2 \in A^\star$) if

$$R \subseteq \{ \langle x, y \rangle : x, y \in U \text{ and } arity(x) = a_1 \text{ and } arity(y) = a_2 \}.$$

Notation 3.11. Given a finite set $A = \{ a_1, \ldots, a_k \}$, by A' we denote the set $\{ a_1', \ldots, a_k' \}$.
Given a finite set of symbols $A = \{ a_1, \ldots, a_k \}$ and a structure $\mathfrak{A}$, by A we denote the set of symbols $\{ a_1, \ldots, a_k \}$.

We will give semantics for $\mathsf{ETFR}(C, F, P)$ in terms of SPFAU. The definition is as follows.

Definition 3.12. An adequate structure for $\mathsf{ETFR}(C, F, P)$ is a structure $\mathfrak{A} = \langle A, C, F, P \rangle$ such that:

1. $A \in \mathsf{SPFAU}$,
2. For each $c \in C$, $c \in Urel_A$,
3. For each $f \in F$ of arity k, $f : {Urel_A}^k \to Urel_A$,
4. For each $p \in P$ of arity $\langle a_1, a_2 \rangle$ $(a_1, a_2 \in A^\star)$, $p \in A$ has arity $\langle a_1, a_2 \rangle$.

Given a PFAU $\mathfrak{A}$, a mapping $\nu : IndVar \to Urel$ is called a *valuation of individual variables*. If $x \in IndVar$, by $\nu[x/a]$ we denote the valuation defined by:

$$\nu[x/a](y) = \begin{cases} \nu(y) & \text{if } y \neq x, \\ a & \text{if } y = x. \end{cases}$$

Definition 3.13. Given an adequate structure $\mathfrak{A} = \langle A, C, F, P \rangle$ for the calculus $\mathsf{ETFR}(C, F, P)$ and a valuation of individual variables ν we define the mapping V_ν giving semantics to terms of $IndTerm(C, F)^\star$ as follows:

1. $V_\nu(x) = \nu(x)$, for all individual variables x,
2. $V_\nu(c) = c$, for all $c \in C$,
3. $V_\nu(f(t_1, \ldots, t_k)) = f\ (V_\nu(t_1), \ldots, V_\nu(t_k))$ for all $f \in F$,
4. $V_\nu(*(t, t')) = *(V_\nu(t), V_\nu(t'))$.

Definition 3.14. Let $\mathfrak{A} = \langle A, C, F, P \rangle$ be an adequate structure for the formalism $\mathsf{ETFR}(C, F, P)$, and let $m : Rel\_Var \to A$. The pair $\langle \mathfrak{A}, m \rangle$ is called a *model*.

Definition 3.15. Given $\varphi \in ForETFR(C, F, P)$ and a valuation for the individual variables ν, we say that ν satisfies the formula φ in the model $\mathfrak{M} = \langle \langle A, C, F, P \rangle, m \rangle$ (denoted by $\mathfrak{M}, \nu \models_{\mathsf{ETFR}} \varphi$) whenever:

1. If $\varphi = t_1 p t_2$ with $p \in RelDes(P)$ and $t_1, t_2 \in IndTerm(C, F)^\star$, then
$$\mathfrak{M}, \nu \models_{\mathsf{ETFR}} t_1 p t_2 \text{ iff } \langle V_\nu(t_1), V_\nu(t_2) \rangle \in m(p).$$
2. If $\varphi = \neg\alpha$, $\mathfrak{M}, \nu \models_{\mathsf{ETFR}} \neg\alpha$ iff $\mathfrak{M}, \nu \not\models_{\mathsf{ETFR}} \alpha$,
3. If $\varphi = \alpha \vee \beta$, $\mathfrak{M}, \nu \models_{\mathsf{ETFR}} \alpha \vee \beta$ iff $\mathfrak{M}, \nu \models_{\mathsf{ETFR}} \alpha$ or $\mathfrak{M}, \nu \models_{\mathsf{ETFR}} \beta$,
4. If $\varphi = \alpha \wedge \beta$, $\mathfrak{M}, \nu \models_{\mathsf{ETFR}} \alpha \wedge \beta$ iff $\mathfrak{M}, \nu \models_{\mathsf{ETFR}} \alpha$ and $\mathfrak{M}, \nu \models_{\mathsf{ETFR}} \beta$,
5. If $\varphi = \exists x \alpha$, $\mathfrak{M}, \nu \models_{\mathsf{ETFR}} \exists x \alpha$ iff
$$\text{there exists } a \in Urel \text{ such that } \mathfrak{M}, \nu[x/a] \models_{\mathsf{ETFR}} \alpha,$$
6. If $\varphi = \forall x \alpha$, $\mathfrak{M}, \nu \models_{\mathsf{ETFR}} \forall x \alpha$ iff
$$\text{for every } a \in Urel,\ \mathfrak{M}, \nu[x/a] \models_{\mathsf{ETFR}(C,F,P)} \alpha.$$

The following mapping translates formulas from $\mathsf{FOLE}(C, F, P)$ into formulas of $\mathsf{ETFR}(C, F, P')$. We will denote by $ForFOLE(C, F, P)$ the set of formulas of $\mathsf{FOLE}(C, F, P)$.

In order to simplify the presentation, from here on given a first-order predicate symbol $p \in P$ with arity k, we will assume that the arity of $p' \in P'$ is $\langle n, m \rangle$ (n, m numbers) with $n + m = k$. Moreover we will assume that the first n parameters from p will be input parameters of p' and the last m parameters from p will be output parameters of p'.

Definition 3.16. The mapping $T_\nabla : ForFOLE(C,F,P) \to ForETFR(C,F,P')$ is defined inductively as follows.

1. $T_\nabla(p(t_1,\ldots,t_n,t'_1,\ldots,t'_m)) = t_1 \star \cdots \star t_n p' t'_1 \star \cdots \star t'_m$, if $p \in P$ and p' has arity $\langle n,m\rangle$,
2. $T_\nabla(t_1 \equiv t_2) = t_1\, 1'\, t_2$,
3. $T_\nabla(\neg\alpha) = \neg T_\nabla(\alpha)$,
4. $T_\nabla(\alpha \vee \beta) = T_\nabla(\alpha) \vee T_\nabla(\beta)$,
5. $T_\nabla(\alpha \wedge \beta) = T_\nabla(\alpha) \wedge T_\nabla(\beta)$,
6. $T_\nabla(\exists x\alpha) = \exists x T_\nabla(\alpha)$,
7. $T_\nabla(\forall x\alpha) = \forall x T_\nabla(\alpha)$.

Given a first-order term t and a valuation of variables ν into a FOLE model $\mathfrak{A}$, by $\mathbf{V}_\nu(t)$ we denote the value of t under the valuation ν in the model $\mathfrak{A}$.

Theorem 3.17. Let $\phi \in ForFOLE(C,F,P)$. Given a FOLE(C,F,P) model $\mathfrak{A}$ with domain A there exists a ETFR(C,F,P') model $\mathfrak{B}$ such that for every valuation for individual variables ν,

$$\mathfrak{A},\nu \models_{\mathsf{FOLE}} \phi \qquad \Longleftrightarrow \qquad \mathfrak{B},\nu \models_{\mathsf{ETFR}} T_\nabla(\phi).$$

Proof:
Let $\mathfrak{G} = \langle A', * \rangle$ be the free groupoid with set of free generators A. Notice that $*$ is injective. Let B be the FullPFAU with base A' and injective mapping $*$. Define, for $c \in C$ and $f \in F$, $c^{\mathfrak{B}} = c$ and $f^{\mathfrak{B}} = f$. Define, for $p \in P$ of arity n and $p' \in P'$ of arity $\langle r,s\rangle$ with $r+s=n$, $p'^{\mathfrak{B}} = \{\langle a_1 * \cdots * a_r, b_1 * \cdots * b_s\rangle : p(a_1,\ldots,a_r,b_1,\ldots,b_s)\}$. Let $m : RelVar \to B$ be arbitrary. Then, we define $\mathfrak{B} = \left\langle \left\langle B, C^{\mathfrak{B}}, F^{\mathfrak{B}}, P'^{\mathfrak{B}} \right\rangle, m \right\rangle$.

It is clear that

$$\forall t \in IndTerm(C,F),\ V_\nu(t) = \mathbf{V}_\nu(t). \tag{9}$$

Given $p \in P$ of arity k, $p' \in P'$ of arity $\langle r,s\rangle$ and $t_1,\ldots,t_r,t'_1,\ldots,t'_s \in IndTerm(C,F)$,

$$\begin{aligned}
&\mathfrak{A} \models_{\mathsf{FOLE}} p(t_1,\ldots,t_r,t'_1,\ldots,t'_s)\\
&\quad\Longleftrightarrow \langle \mathbf{V}_\nu(t_1),\ldots,\mathbf{V}_\nu(t_r),\mathbf{V}_\nu(t'_1),\ldots,\mathbf{V}_\nu(t'_s)\rangle \in p && \text{(Def. } \models_{\mathsf{FOLE}})\\
&\quad\Longleftrightarrow \langle \mathbf{V}_\nu(t_1) * \cdots * \mathbf{V}_\nu(t_r),\mathbf{V}_\nu(t'_1) * \cdots * \mathbf{V}_\nu(t'_s)\rangle \in p' && \text{(Def. } p'^{\mathfrak{B}})\\
&\quad\Longleftrightarrow \langle V_\nu(t_1) * \cdots * V_\nu(t_r),V_\nu(t'_1) * \cdots * V_\nu(t'_s)\rangle \in p' && \text{(by (9))}\\
&\quad\Longleftrightarrow \mathfrak{B},\nu \models_{\mathsf{ETFR}} t_1 \star \cdots \star t_r p' t'_1 \star \cdots \star t'_s && \text{(Def. } \models_{\mathsf{ETFR}})\\
&\quad\Longleftrightarrow \mathfrak{B},\nu \models_{\mathsf{ETFR}} T_\nabla(p(t_1,\ldots,t_r,t'_1,\ldots,t'_s)). && \text{(Def. } T_\nabla)
\end{aligned}$$

The remaining part of the proof is easy and is left to the reader.

Theorem 3.18. Let $\phi \in ForFOLE(C,F,P)$. Let $\mathfrak{A} = \left\langle \left\langle A, C^{\mathfrak{A}}, F^{\mathfrak{A}}, P'^{\mathfrak{A}} \right\rangle, m \right\rangle$ be a ETFR(C,F,P') model. Then there exists a FOLE(C,F,P) model $\mathfrak{B} = \langle B, C^{\mathfrak{B}}, F^{\mathfrak{B}}, P^{\mathfrak{B}} \rangle$ such that for every valuation for individual variables ν,

$$\mathfrak{A},\nu \models_{\mathsf{ETFR}} T_\nabla(\phi) \qquad \Longleftrightarrow \qquad \mathfrak{B},\nu \models_{\mathsf{FOLE}} \phi.$$

Proof:
Let $B = Urel_A$. For each $c \in C$ and $f \in F$, we define $c^{\mathfrak{B}} = c$ and $f^{\mathfrak{B}} = f$. For each $p' \in P'$ of arity $\langle r,s\rangle$ we define

$$p^{\mathfrak{B}} = \left\{ \langle a_1,\ldots,a_r,b_1,\ldots,b_s\rangle : \langle a_1 * \cdots * a_r, b_1 * \cdots * b_s\rangle \in p'^{\mathfrak{A}} \right\}.$$

If $\phi = p(t_1, \ldots, t_r, t'_1, \ldots, t'_s)$ with $p \in P$, then

$$\begin{aligned}
&\mathfrak{A},\nu \models_{\mathsf{ETFR}} T_\nabla(p(t_1, \ldots, t_r, t'_1, \ldots, t'_s)) \\
&\quad \Longleftrightarrow \mathfrak{A}, \nu \models_{\mathsf{ETFR}} t_1 \star \cdots \star t_r p' t'_1 \star \cdots \star t'_s && \text{(Def. } T_\nabla\text{)} \\
&\quad \Longleftrightarrow \langle V_\nu(t_1) * \cdots * V_\nu(t_r), V_\nu(t'_1) * \cdots * V_\nu(t'_r)\rangle \in p' && \text{(Def. } \models_{\mathsf{ETFR}}\text{)} \\
&\quad \Longleftrightarrow \langle V_\nu(t_1), \ldots, V_\nu(t_r), V_\nu(t'_1), \ldots, V_\nu(t'_r)\rangle \in p && \text{(Def. } p \text{)} \\
&\quad \Longleftrightarrow \langle \mathbf{V}_\nu(t_1), \ldots, \mathbf{V}_\nu(t_r), \mathbf{V}_\nu(t'_1), \ldots, \mathbf{V}_\nu(t'_r)\rangle \in p && \text{(by (9))} \\
&\quad \Longleftrightarrow \mathfrak{B}, \nu \models_{\mathsf{FOLE}} p(t_1, \ldots, t_r, t'_1, \ldots, t'_s). && \text{(Def. } \models_{\mathsf{FOLE}}\text{)}
\end{aligned}$$

The remaining part of the proof is easy and is left to the reader.

By $\models_{\mathsf{FOLE}} \phi$ we will denote the fact that formula ϕ is *valid* in FOLE. In a similar way we say that a formula ϕ from ETFR is *valid* in ETFR if for all ETFR model $\mathfrak{A}$ and every valuation of individual variables ν we have $\mathfrak{A}, \nu \models_{\mathsf{ETFR}} \phi$.

By using the mapping T_∇ we obtain the following result on the interpretability of FOLE in ETFR.

Theorem 3.19. Let ϕ be a FOLE(C, F, P) formula. Then

$$\models_{\mathsf{FOLE}} \phi \quad \Longleftrightarrow \quad \models_{\mathsf{ETFR}} T_\nabla(\phi).$$

Proof:

$\Rightarrow$) If $\not\models_{\mathsf{ETFR}} T_\nabla(\phi)$, then there exists an ETFR(C, F, P') model $\mathfrak{A}$ and a valuation of individual variables ν such that $\mathfrak{A}, \nu \not\models_{\mathsf{ETFR}} T_\nabla(\phi)$. Then by Thm. 3.18 there exists a FOLE(C, F, P) model $\mathfrak{B}$ such that $\mathfrak{B}, \nu \not\models_{\mathsf{FOLE}} \phi$. Then $\not\models_{\mathsf{FOLE}} \phi$.
$\Leftarrow$) If $\not\models_{\mathsf{FOLE}} \phi$, then there exists a FOLE(C, F, P) model $\mathfrak{A}$ and a valuation of individual variables ν such that $\mathfrak{A}, \nu \not\models_{\mathsf{FOLE}} \phi$. By Thm. 3.17 there exists a ETFR(C, F, P') structure $\mathfrak{B}$ such that $\mathfrak{B}, \nu \not\models_{\mathsf{ETFR}} T_\nabla(\phi)$. Then $\not\models_{\mathsf{ETFR}} T_\nabla(\phi)$.

In the remaining part of this subsection we will show that ETFR(C, F, P) can be interpreted into CFR(A) for a suitable set of constant relation symbols A. Finally, by exploiting the relationship existing between CFR(A) and AFA we will show how to reason algebraically in order to prove logical properties from FOLE and ETFR.

Given sets C, F and P consisting of constant, function and relation symbols respectively, by K we will denote the set $C' \cup F' \cup P'$. By CFR$^+(K)$ we will denote the extension of CFR obtained by adding the following axioms.

1. The formula
$$1;1'_{\mathsf{U}};1 = 1,$$
which implies that models of CFR$^+$ are abstract fork algebras with urelements.
2. for each $c \in C$, we add the following equations stating that c' is a *constant* relation having an urelement in its range:
$$\begin{aligned}
&\breve{c'};c' + 1'_{\mathsf{U}} = 1'_{\mathsf{U}} \ (c' \text{ is functional}), \\
&1;c' = c' \ (c' \text{ is left-ideal}) , \\
&c';1 = 1 \ (c' \text{ is nonempty}) .
\end{aligned}$$
3. for each $f \in F$ with arity k, the following equations stating that f' is a functional relation that takes k urelements as input and produces urelements as output:
$$\begin{aligned}
&\breve{f'};f' + 1'_{\mathsf{U}} = 1'_{\mathsf{U}}, \\
&\underbrace{1'_{\mathsf{U}} \otimes \cdots \otimes 1'_{\mathsf{U}}}_{k \text{ times}};f' = f'.
\end{aligned}$$

4. for each $p \in P$ with arity $\langle m, n\rangle$, the following equations stating that p' is a binary relation expecting m urelements as input and n urelements as output:

$$\underbrace{1'_U \otimes \cdots \otimes 1'_U}_{m \text{ times}} ; p' ; \underbrace{1'_U \otimes \cdots \otimes 1'_U}_{n \text{ times}} = p'.$$

In items 3 and 4 we are assuming that f' has input arity k and that p' has arity $\langle m, n\rangle$. We can generalize to arbitrary arities by rearranging parenthesis in a convenient way. For example, if p' has arity $\langle (u \star u) \star (u \star u), (u \star u) \star u\rangle$, then we would impose the condition

$$(1'_U \otimes 1'_U) \otimes (1'_U \otimes 1'_U) p' (1'_U \otimes 1'_U) \otimes 1'_U = p'.$$

Notice that given a finite set K with constant, function and relation symbols, only finitely many equations are introduced in items 1 to 4 above.

In what follows, $t^{;n}$ is an abbreviation for $\overbrace{t; \cdots ; t}^{n \text{ times}}$. For the sake of completeness, $t^{;0}$ is defined as 1'.

The following mapping translates $\mathsf{ETFR}(C, F, P)$ formulas into $\mathsf{CFR}^+(K)$ formulas. The definition proceeds in two steps, since also terms (not only formulas) need to be translated.

For the following definitions, σ will be a sequence of numbers increasingly ordered. Intuitively, the sequence will contain the indices of those individual variables that appear free in the formula (or term) being translated. By $Ord(n, \sigma)$ 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.

Definition 3.20. The mapping $\delta_\sigma : IndTerm(C, F) \to RelDes(C' \cup F')$, translating individual terms into relation designations, is defined inductively by the conditions:

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

Before defining the mapping T_σ translating ETFR formulas, we need to define some auxiliary terms. Given a sequence σ such that $Length(\sigma) = l$, we define the term $\Delta_{\sigma,n}$ $(n < \omega)$ by the condition

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

The term $\Delta_{\sigma,n}$ can be understood as a cylindrification [26, 27] in the k-th. coordinate of a l-dimensional space.

For the next mapping to be correctly defined, we assume that atomic formulas of the form $R \equiv S$ do not occur in the scope of a quantifier over individual variables. This is a reasonable assumption because, since atomic formulas of this form do not contain any individual variables, they can be promoted outside the scope of quantifiers.

Definition 3.21. The mapping T_σ, translating $\mathsf{ETFR}(C, F, P)$ formulas into $\mathsf{CFR}^+(K)$ formulas, is defined inductively as follows.

1. $T_\sigma(R \equiv S) = R \equiv S$ $(R, S \in RelDes(P))$,

2. $T_\sigma(t_1 \star \cdots \star t_r R t'_1 \star \cdots \star t'_s) = T'_\sigma(t_1 \star \cdots \star t_r R t'_1 \star \cdots \star t'_s) \equiv 1'_{\cup^k};1$, $(k = Length(\sigma)$, $t_i, t'_j \in IndTerm(C,F)$ for all i,j, $1 \leq i \leq r$, $1 \leq j \leq s$ and $R \in RelDes(P))$,
3. $T_\sigma(\neg\alpha) = \neg T_\sigma(\alpha)$,
4. $T_\sigma(\alpha \vee \beta) = T_\sigma(\alpha) \vee T_\sigma(\beta)$,
5. $T_\sigma(\alpha \wedge \beta) = T_\sigma(\alpha) \wedge T_\sigma(\beta)$,
6. $T_\sigma(\exists v_n \alpha) = T'_\sigma(\exists v_n \alpha) \equiv 1'_{\cup^k};1$, $(k = Length(\sigma))$,
7. $T_\sigma(\forall v_n \alpha) = T'_\sigma(\forall v_n \alpha) \equiv 1'_{\cup^k};1$, $(k = Length(\sigma))$,
8. $T'_\sigma(t_1 \star \cdots \star t_r R t'_1 \star \cdots \star t'_s) =$

$$\Big((\delta_\sigma(t_1) \nabla \cdots \nabla \delta_\sigma(t_r)) \;\nabla\; (\delta_\sigma(t'_1) \nabla \cdots \nabla \delta_\sigma(t'_s))\,;\breve{R}\Big)\,;\breve{2};1, \tag{10}$$

9. $T'_\sigma(\neg\alpha) = \overline{T'_\sigma(\alpha)}$,
10. $T'_\sigma(\alpha \vee \beta) = T'_\sigma(\alpha) + T'_\sigma(\beta)$,
11. $T'_\sigma(\alpha \wedge \beta) = T'_\sigma(\alpha) \cdot T'_\sigma(\beta)$,
12. $T'_\sigma(\exists v_n(\alpha)) = \Delta_{\sigma,n};T'_{[\sigma \oplus n]}(\alpha)$,
13. $T'_\sigma(\forall v_n(\alpha)) = T'_\sigma(\neg\exists v_n \neg\alpha)$.

In item 8 we are assuming that R has arity $\langle r, s\rangle$ (r, s numbers). If we allow for arbitrary arities, then the parenthesis in formula (10) should be rearranged. For example, if R has arity $\langle (u \star u) \star (u \star u), (u \star u) \star u\rangle$, then

$$T'_\sigma((t_1 \star t_2) \star (t_3 \star t_4)\, R\, (t_5 \star t_6) \star t_7)$$
$$= \Big(((\delta_\sigma(t_1) \nabla \delta_\sigma(t_2)) \nabla (\delta_\sigma(t_3) \nabla \delta_\sigma(t_4))) \nabla ((\delta_\sigma(t_5) \nabla \delta_\sigma(t_6)) \nabla \delta_\sigma(t_7))\,;\breve{R}\Big)\,;\breve{2};1.$$

Notation 3.22. 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.

Theorem 3.23. Let $\alpha \in \mathsf{ETFR}(C,F,P)$ and let $\sigma = \sigma_\alpha$. Then,

$$\models_{\mathsf{ETFR}} \alpha \quad \Longleftrightarrow \quad \models_{\mathsf{CFR}^+} T_\sigma(\alpha).$$

Theorem 3.23 shows that reasoning in ETFR can be replaced by reasoning in CFR^+. The reason why this is considered an algebraization of ETFR is because validity in CFR^+ means that some formula holds in the class of algebras SAFAU. In order to obtain an algebraization of FOLE it suffices to compose the mappings T_∇ and T'_σ. We then obtain the following mapping $T_{\nabla,\sigma} : ForFOLE(C,F,P) \to ForCFR(C \cup F \cup P)$. If we recall that in order to apply the mapping T_∇ there is to divide arguments of predicate symbols between input and output arguments, we will assume that all arguments are to be considered as input arguments.

1. $T_{\nabla,\sigma}(p(t_1,\ldots,t_k)) = (\delta_\sigma(t_1) \nabla \cdots \nabla \delta_\sigma(t_k))\,;p';1$,
2. $T_{\nabla,\sigma}(\neg\alpha) = \overline{T_{\nabla,\sigma}(\alpha)}$,
3. $T_{\nabla,\sigma}(\alpha \vee \beta) = T_{\nabla,\sigma}(\alpha) + T_{\nabla,\sigma}(\beta)$,
4. $T_{\nabla,\sigma}(\exists v_n \alpha) = \Delta_{\sigma,n};T_{\nabla,[\sigma \oplus n]}(\alpha)$.

By using the translation $T_{\nabla,\sigma}$ we can prove the following theorem.

Theorem 3.24. Let $\alpha \in ForFOLE(C,F,P)$ and let $\sigma = \sigma_\alpha$. Assume also that $Length(\sigma) = k$,

$$\models_{\mathsf{FOLE}} \alpha \quad \Longleftrightarrow \quad \models_{\mathsf{CFR}^+} T_{\nabla,\sigma}(\alpha) \equiv 1'_{\cup^k};1.$$

From CFR^+ we define the formalism CFREQ^+ as a restriction of CFR^+. CFREQ^+ is defined as follows.

Formulas: Equations from $ForCFR^+$.

Inference Rules: Those of equational logic.
Axioms: Set of equations characterizing the class AFAU.

Notice that in the axiomatization of CFREQ^+ we dropped the axiom requiring models to be simple.

Theorem 3.25. Let e be an equation. Then,

$$\models_{\mathsf{CFR}^+} e \quad \Longleftrightarrow \quad \vdash_{\mathsf{CFREQ}^+} e.$$

Proof:
By definition of the formalism CFR^+,

$$\models_{\mathsf{CFR}^+} e \text{ iff for all } \mathfrak{A} \in \mathsf{SAFAU},\ \mathfrak{A} \models e. \tag{11}$$

Since the variety generated by SAFAU is AFAU, SAFAU and AFAU share the same equational theory. Thus,

$$\text{for all } \mathfrak{A} \in \mathsf{SAFAU},\ \mathfrak{A} \models e \text{ iff for all } \mathfrak{A} \in \mathsf{AFAU},\ \mathfrak{A} \models e. \tag{12}$$

Then, by (11) and (12),

$$\models_{\mathsf{CFR}^+} e \text{ iff for all } \mathfrak{A} \in \mathsf{AFAU},\ \mathfrak{A} \models e. \tag{13}$$

By definition of CFREQ^+ and (13),

$$\models_{\mathsf{CFR}^+} e \quad \Longleftrightarrow \quad \models_{\mathsf{CFREQ}^+} e. \tag{14}$$

Since equational logic is complete,

$$\models_{\mathsf{CFREQ}^+} e \quad \Longleftrightarrow \quad \vdash_{\mathsf{CFREQ}^+} e. \tag{15}$$

Finally, joining (14) and (15),

$$\models_{\mathsf{CFR}^+} e \quad \Longleftrightarrow \quad \vdash_{\mathsf{CFREQ}^+} e.$$

Corollary 3.26. Let $\alpha \in ForFOLE(C, F, P)$ and let $\sigma = \sigma_\alpha$. Assume also that $Length(\sigma) = k$,

$$\models_{\mathsf{FOLE}} \alpha \quad \Longleftrightarrow \quad \vdash_{\mathsf{CFREQ}^+} T_{\nabla,\sigma}(\alpha) \equiv 1'_{\mathsf{U}}{}^{k};1.$$

Proof:
By Thm. 3.24,

$$\models_{\mathsf{FOLE}} \alpha \quad \Longleftrightarrow \quad \models_{\mathsf{CFR}^+} T_{\nabla,\sigma}(\alpha) \equiv 1'_{\mathsf{U}}{}^{k};1. \tag{16}$$

By Thm. 3.25,

$$\models_{\mathsf{CFR}^+} T_{\nabla,\sigma}(\alpha) \equiv 1'_{\mathsf{U}}{}^{k};1 \quad \Longleftrightarrow \quad \vdash_{\mathsf{CFREQ}^+} T_{\nabla,\sigma}(\alpha) \equiv 1'_{\mathsf{U}}{}^{k};1. \tag{17}$$

Thus, by (16) and (17),

$$\models_{\mathsf{FOLE}} \alpha \quad \Longleftrightarrow \quad \vdash_{\mathsf{CFREQ}^+} T_{\nabla,\sigma}(\alpha) \equiv 1'_{\mathsf{U}}{}^{k};1.$$

In [27], FOLE is algebraized using cylindric algebras. Cylindric algebras are a very natural algebraic counterpart of FOLE. The fact they have a Boolean algebra reduct allows to algebraize the propositional part of FOLE. Also, for each quantifier $\exists v_i$, a new operator c_i (called the i-th cylindrification) is defined. The axioms for the cylindrifications are natural translations of valid properties for the existential quantifiers. For example, the property $\exists v_i \exists v_j \alpha \Leftrightarrow \exists v_j \exists v_i \alpha$ corresponds to the cylindric algebra axiom $c_i c_j x = c_j c_i x$, for all pair of indices $\langle i, j \rangle$. Finally, for each pair of indices $\langle i, j \rangle$, a constant element d_{ij} (called the i,j-diagonal element) is distinguished. Intuitively, d_{ij} characterizes algebraically the predicate $v_i = v_j$. Notice that infinitely many axioms are required for axiomatizing the infinitely many cylindrifications and diagonal elements. The fact fork algebras have only finitely many operators and are axiomatized by a finite set of equations makes fork algebras more attractive in computer science, where this finiteness plays an essential part in the implementability of a calculus for program construction based on fork algebras.

3.2. Algebraization of Non-Classical Logics

In this subsection we will show how to algebraize many non-classical logics (modal logics, dynamic logic, relevant logics, intuitionistic logic, intermediate logics) with the help of fork algebras. The results that will be presented in this part were published in [17, 18] and were obtained as joint work with Ewa Orlowska.

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 [20], Gries [19]).

In this subsection we will develop an equational formalism that is capable of modeling a great variety of applied nonclassical logics and of simulating nonclassical means of reasoning. The formalism is based on fork algebras. The idea of relational formalization of logical systems has been originated in Orlowska ([43]) and developed further in Orlowska ([44, 45, 46]).

Examples of relational formalisms for applied logics can also be found in Buszkowski and Orlowska ([7]), Demri and Orlowska ([10]), Demri et al. ([11]), Herment and Orlowska ([28]). The main idea of all these relational formalisms is to represent formulas of a nonclassical logic as relations from a suitable algebra of binary relations, and to develop a relational proof system for this logic. However, since the class of algebras of binary relations (representable relation algebras) is not finitely axiomatizable, and the finitely axiomatizable class of relation algebras is not representable (it is not the case that every relation algebra is isomorphic to an algebra of binary relations), the existing relational framework for nonclassical logics suffers several disadvantages. The class of fork algebras is both finitely axiomatizable and representable, and hence a fork algebra formalism seems to be an appropriate candidate for relational formalization of nonclassical logics. Here we will prove interpretability of several nonclassical logics in CFR^+.

The interpretability of a nonclassical logic in the logic CFR^+ is established by means of a provability preserving translation of formulas of the logic into formulas of CFR^+. Under that translation both formulas, formerly understood as sets of states, and accessibility relations receive a uniform representation as relations. The propositional connectives are transformed into relational operations. The constraints on accessibility relations are translated into relational equations. The major advantage of relational formalization is that it provides a uniform framework for representation of a broad class of applied logics and enables us to apply an equational proof theory to these logics.

3.2.1. Interpretability of Modal Logics in CFR^+

Definition 3.27. A frame is a relational system $\langle W, R \rangle$ where W is a nonempty set of possible worlds and $R \subseteq W \times W$ is a binary accessibility relation between worlds.

Definition 3.28. Given a frame $\langle W, R \rangle$, a Kripke model is a triple $\langle W, R, m \rangle$ where m is a meaning function that assigns subsets of W to propositional variables.

Definition 3.29. The inductive definition of satisfiability describes the truth conditions depending on the complexity of formulas. For atomic formulas (i.e., propositional variables) we have:

(at) $M, w \models p$ iff $w \in m(p)$ for any propositional variable p.

For formulas built with extensional operators such as classical negation, disjunction, conjunction or implication, their satisfiability at a possible world is completely determined by satisfiability of their subformulas at that world.

($\neg$) $M, w \models \neg\alpha$ iff not $M, w \models \alpha$
($\vee$) $M, w \models \alpha \vee \beta$ iff $M, w \models \alpha$ or $M, w \models \beta$
($\wedge$) $M, w \models \alpha \wedge \beta$ iff $M, w \models \alpha$ and $M, w \models \beta$
($\rightarrow$) $M, w \models \alpha \rightarrow \beta$ iff $M, w \models \neg\alpha \vee \beta$.

For formulas defined from modal operators, as $[R]$ (necessity) and $\langle R \rangle$ (possibility), we have

($[R]$) $M, w \models [R]\alpha$ iff for all $u \in W$, $\langle w, u \rangle \in R$ implies $M, u \models A$
($\langle R \rangle$) $M, w \models \langle R \rangle\, \alpha$ iff there is an $u \in W$ s.t. $\langle w, u \rangle \in R$ and $M, u \models A$.

For the sake of simplicity, we use the same symbol for the relational constant R that appears in modal operators and the accessibility relation which is denoted by this constant.

In various modal logics, the accessibility relation is assumed to satisfy certain conditions. If we call FRM (Conditions) the class of all those frames in which the accessibility relation satisfies the "Conditions", then the set of all the formulas valid in that class is called the logic L(Conditions). For example: K (no restriction on accessibility), T (reflexive), KB (symmetric), B (reflexive, symmetric), $K4$ (transitive), $KB4$ (symmetric, transitive), $S4$ (reflexive, transitive), $S5$ (equivalence), $S4.1$ (reflexive, symmetric, atomic), KD (serial), KDB (symmetric, serial), $KD4$ (transitive, serial), $S4.3.1$ (reflexive, transitive, euclidean, discrete), G (transitive, well–capped).

Definition 3.30. Given a Kripke model $M = \langle W, R, m \rangle$ and a modal formula α, α is said to be *true* in M if for all $w \in W$, $M, w \models \alpha$.

Definition 3.31. A modal formula α is called *valid* if it is true in all models.

In the following definition we introduce a mapping T_M translating modal formulas into CFR^+ equations.

Definition 3.32. Before defining the mapping T_M, we define the mapping T'_M by:

(var) $T'_M(p_i) = P_i$, where p_i is a propositional variable and $P_i \in RelVar$,
(neg) $T'_M(\neg\alpha) = 1'_\cup ; \overline{T'_M(\alpha)}$,
(and) $T'_M(\alpha \wedge \beta) = T'_M(\alpha) \cdot T'_M(\beta)$,
(or) $T'_M(\alpha \vee \beta) = T'_M(\alpha) + T'_M(\beta)$,
($\langle R \rangle$) $T'_M(\langle R \rangle\, \alpha) = R ; T'_M(\alpha)$.
($[R]$) $T'_M([R]\alpha) = T'_M(\neg \langle R \rangle \neg\alpha)$.

For the sake of simplicity we assume that the constant R from the modal language is translated into a constant from the language of CFR^+ that is denoted by the same symbol.

We finally define the mapping T_M by $T_M(\alpha) = T'_M(\alpha) + \overline{1_\cup} \equiv 1$.

The next theorem (whose complete proof is given in [17]) shows the interpretability of any modal logic $L(C)$ (C a set of FOLE sentences) into CFR^+. We will denote by $\langle\rangle$ the empty sequence of indices. Also, given a set of formulas Ψ, by $T_M(\Psi)$ we denote the set $\{ T_M(\psi) : \psi \in \Psi \}$, and by $T_{\nabla,\sigma}(\Psi)$ we denote the set $\{ T_{\nabla,\sigma}(\psi) : \psi \in \Psi \}$.

Theorem 3.33. Given a modal logic $L(C)$, where C is a set of FOLE sentences, a set of modal formulas Ψ and a modal formula φ,

$$\Psi \models_{L(C)} \varphi \qquad \Longleftrightarrow \qquad T_{\nabla,\langle\rangle}(C) \cup T_M(\Psi) \models_{\mathsf{CFR}^+} T_M(\varphi).$$

Notice that the use of the relational composition to algebraize the modalities is a standard thing to do (see for example [6] Chapters 5 and 6). The result Thm. 3.33 provides, is that *any* modal logic whose accessibility relation is characterized by a set of FOLE sentences has a relational algebraization. Much the same as the naive translation of modal logics into FOLE allows to use the tools of first-order logic to study modal logics, Thm. 3.33 allows to use relational tools.

3.2.2. Interpretability of Dynamic Logic in CFR

In [24], the syntax and semantics of propositional dynamic logic are introduced. Dynamic logic is considered as a programming logic, i.e., a logic suitable for asserting and proving properties of programs. Dynamic logic is a modal logic whose modal operators are determined by programs understood as binary relations in a set of computation states. The following definitions provide a formal description of propositional dynamic logic.

Definition 3.34. Let us consider a set P_0 of so called atomic programs, and a set F_0 of so called atomic dynamic formulas. From these sets we will construct the sets F of dynamic formulas and P of compound programs.

F and P are the smallest sets satisfying the following conditions:

1. $true \in F$; $false \in F$; $F_0 \subseteq F$,
2. if $p \in F$ and $q \in F$ then $\neg p \in F$ and $(p \vee q) \in F$,
3. if $p \in F$ and $\alpha \in P$ then $\langle \alpha \rangle p \in F$,
4. $P_0 \subseteq P$,
5. if $\alpha \in P$ and $\beta \in P$ then $(\alpha \cup \beta) \in P$, $(\alpha ; \beta) \in P$ and $\alpha^* \in P$,
6. if $p \in F$ then $p? \in P$.

Definition 3.35. A structure is a triple $D = \langle W, \tau, \delta \rangle$ where W is a set of *states*, τ assigns subsets of W to atomic formulas, and δ assigns subsets of $W \times W$ to atomic programs. The mappings τ and δ are extended inductively to determine the meaning of compound formulas and programs as follows:

$\tau(true) = W$,
$\tau(false) = \emptyset$,
$\tau(\neg p) = \overline{\tau(p)}$,
$\tau(p \vee q) = \tau(p) \cup \tau(q)$,
$\tau(\langle \alpha \rangle p) = \{ s \in W : \exists t \, (\langle s, t \rangle \in \delta(\alpha) \wedge t \in \tau(p)) \}$,
$\delta(\alpha ; \beta) = \{ \langle s, t \rangle : \exists u \, (\langle s, u \rangle \in \delta(\alpha) \wedge \langle u, t \rangle \in \delta(\beta)) \}$,
$\delta(\alpha \cup \beta) = \delta(\alpha) \cup \delta(\beta)$,
$\delta(p?) = \{ \langle s, s \rangle : s \in \tau(p) \}$,
$\delta(\alpha^*) = \delta(\alpha)^* = \{ \langle s, t \rangle : \exists k, s_0, s_1, ..., s_k (s_0 = s \wedge s_k = t \wedge (\forall i, 1 \leq i \leq k) \langle s_{i-1}, s_i \rangle \in \delta(\alpha)) \}$.

Propositional dynamic logic is know to have a complete Hilbert–style calculus. The calculus is given in the following definition. The proof of the completeness of the calculus is given in [24], Thm. 2.11, p.515.

Definition 3.36. The calculus for propositional dynamic logic is given by the following axiom schemas and inference rules:

(A1) all instances of tautologies of the propositional calculus.
(A2) $\langle \alpha \rangle (p \vee q) \leftrightarrow (\langle \alpha \rangle p \vee \langle \alpha \rangle q)$
(A3) $\langle \alpha ; \beta \rangle p \leftrightarrow \langle \alpha \rangle \langle \beta \rangle p$
(A4) $\langle \alpha \cup \beta \rangle p \leftrightarrow (\langle \alpha \rangle p \vee \langle \beta \rangle p)$
(A5) $\langle \alpha^* \rangle p \leftrightarrow (p \vee \langle \alpha \rangle \langle \alpha^* \rangle p)$
(A6) $\langle q? \rangle p \leftrightarrow p \wedge q$
(A7) $[\alpha^*](p \rightarrow [\alpha]p) \rightarrow (p \rightarrow [\alpha^*]p)$
(A8) $[\alpha](p \rightarrow q) \rightarrow ([\alpha]p \rightarrow [\alpha]q)$.

The inference rules for the calculus are, as in the modal logic K, modus ponens and generalization.

The presence of the Kleene star operator in the language of dynamic logic requires a slight generalization of fork algebras in order to obtain the interpretability result.

Definition 3.37. A *closure* AFAU (CAFAU for short), is a structure $\langle A, * \rangle$ such that A is an AFAU, and $*$ satisfies the equations

$$R^* = 1' + R;R^*, \tag{18}$$

$$R^*;S;1 \leq S;1 + R^*;(\overline{S;1} \cdot R;S;1). \tag{19}$$

From the class CAFAU of closure fork algebras with urelements we will define as an extension of CFR^+ the formalism CFR^*. This formalism is obtained by modifying Def. 3.2 (that defines the set of relation designations *RelDes*) by adding the condition

$$\text{If } R \in \mathit{RelDes}, \text{ then } R^* \in \mathit{RelDes},$$

and we add formulas (18) and (19) as axioms.

The following mapping will be used in order to prove the interpretability of propositional dynamic logic in the formalism CFR^*.

Definition 3.38. In order to define mappings T_{DL} and T_P from dynamic logic formulas and compound programs, respectively, onto CFR^* we first define recursively the mappings T'_{DL} and T_P by:

$T'_{DL}(p_i) = P_i$, (p_i an atomic formula.)
$T'_{DL}(\mathit{true}) = \cup 1$,
$T'_{DL}(\mathit{false}) = 0$,
$T'_{DL}(\neg p) = 1' \cup ; \overline{T'_{DL}(p)}$,
$T'_{DL}(p \vee q) = T'_{DL}(p) + T'_{DL}(q)$,
$T'_{DL}(\langle R \rangle p) = T_P(R);T'_{DL}(p)$,
$T_P(R_i) = R_i$, (R_i an atomic program.)
$T_P(R \cup S) = T_P(R) + T_P(S)$,
$T_P(R; S) = T_P(R);T_P(S)$,
$T_P(R^*) = T_P(R)^*$,
$T_P(p?) = T'_{DL}(p) \cdot 1' \cup$.

Next, we define the mapping T_{DL} by $T_{DL}(\alpha) = T'_{DL}(\alpha) + \overline{\cup 1} \equiv 1$.

The following theorem (whose complete proof is given in [17]), presents the result on the interpretability of propositional dynamic logic into CFR^*.

Theorem 3.39. Given a dynamic formula φ,

$$\vdash_{DL} \varphi \qquad \Longleftrightarrow \qquad \models_{\mathsf{CFR}^*} T_{DL}(\varphi).$$

We obtain a fork-algebraic formalization of relevant logics (see [44] for an introduction to relevant logics related to this work) in much the same way as that of modal logics. Although models of relevant logics are based on ternary accessibility relations, there is an easy way of representing these relations as binary ones. In general, given a ternary relation R, we will codify it by the binary relation $\{\langle x, *(y,z)\rangle : R(x,y,z)\}$.

The main result concerning the algebraization of relevant logics is given by the following theorem whose complete proof appears in [17].

Theorem 3.40. There is a recursive mapping T_R translating relevant formulas into ForCFR^+ such that for every relevant formula α,

$$\vdash_{RL} \alpha \qquad \Longleftrightarrow \qquad \models_{\mathsf{CFR}^+} T_R(\alpha).$$

3.2.3. A Rasiowa–Sikorski Calculus for CFREQ$^+$

The original Rasiowa–Sikorski proof system presented in [48] refers to the classical predicate logic. The system is designed for verification of validity of formulas from this logic. It consists of a pair of rules for each propositional connective and each quantifier. Every pair of rules, in turn, consists of a "positive" rule and a "negative" rule. A positive (resp. negative) rule exhibits the logical behavior of the underlying connective or quantifier (negated connective or negated quantifier). For example, the rules for conjunction are the following

$$\frac{\Gamma, \alpha \wedge \beta, \Delta}{\Gamma, \alpha, \Delta \quad \Gamma, \beta, \Delta} \ (P\wedge) \qquad \frac{\Gamma, \neg(\alpha \wedge \beta), \Delta}{\Gamma, \neg\alpha, \neg\beta, \Delta} \ (N\wedge)$$

The system operates in a top-down manner. Application of a rule results in a decomposition of a given formula into the formulas that are the arguments of a respective connective or quantifier. In general, the rules apply to finite sequences of formulas. To apply a rule we choose a formula in a sequence that is to be decomposed and we replace it by its components, thus obtaining either a single new sequence (for 'or'-like connectives) or a pair of sequences (for 'and'-like connectives). In the process of decomposition we form a tree whose nodes consist of finite sequences of formulas. We stop applying rules to the formulas in a node after obtaining an axiom sequence (appropriately defined) or when none of the rules is applicable to the formulas in this node. If a decomposition tree of a given formula is finite, then its validity can be syntactically recognized from the form of the sequences appearing in the leaves of the tree. The system we present is an extension of the proof system presented in Orlowska [43, 46]. The system consists of a positive and a negative decomposition rule for each relational operation from the language of CFREQ$^+$, and moreover of the specific rules that reflect properties of the function $\star$ and the relational constant 1'.

In this part we will assume that $IndVar = UreVar \cup CompVar$, with $UreVar$ and $CompVar$ two disjoint sets.

Next we will present the rules of the sequent calculus FLC for CFREQ$^+$. Since we are dealing with fork algebras with urelements (required in order to interpret first-order theories), the calculus we present is more involved than a calculus for fork algebras when no assumption is done on the existence of urelements.

$$\frac{\Gamma, xR+Sy, \Delta}{\Gamma, xRy, xSy, \Delta} \ (P+) \qquad \frac{\Gamma, x\overline{R+S}y, \Delta}{\Gamma, x\overline{R}y, \Delta \quad \Gamma, x\overline{S}y, \Delta} \ (N+)$$

$$\frac{\Gamma, xR\cdot Sy, \Delta}{\Gamma, xRy, \Delta \quad \Gamma, xSy, \Delta} \ (P\cdot) \qquad \frac{\Gamma, x\overline{R\cdot S}y, \Delta}{\Gamma, x\overline{R}y, x\overline{S}y, \Delta} \ (N\cdot)$$

$$\frac{\Gamma, xR;Sy, \Delta}{\Gamma, xRz, \Delta, xR;Sy \quad \Gamma, zSy, \Delta, xR;Sy} \ (P;) \qquad \frac{\Gamma, x\overline{R;S}y, \Delta}{\Gamma, x\overline{R}z_1, z_1\overline{S}y, \Delta \quad \Gamma, x\overline{R}z_2, z_2\overline{S}y, \Delta} \ (N;)$$

$$\frac{\Gamma, x\overline{\overline{R}}y, \Delta}{\Gamma, xRy, \Delta} \ (N^-)$$

$$\frac{\Gamma, x\breve{R}y, \Delta}{\Gamma, yRx, \Delta} \ (P^\smile) \qquad \frac{\Gamma, x\overline{\breve{R}}y, \Delta}{\Gamma, y\overline{R}x, \Delta} \ (N^\smile)$$

$$\frac{\Gamma, xR\nabla Sy, \Delta}{\Gamma, y1'u \star v, \Delta, xR\nabla Sy \quad \Gamma, xRu, \Delta, xR\nabla Sy \quad \Gamma, xSv, \Delta, xR\nabla Sy} \ (P\nabla)$$

$$\frac{\Gamma, x\overline{R\nabla S}y, \Delta}{\begin{array}{l}\Gamma, y0'u_1 \star v_1, x\overline{R}u_1, x\overline{S}v_1, \Delta \quad \Gamma, y0'u_2 \star v_2, x\overline{R}u_2, x\overline{S}v_2, \Delta \\ \Gamma, y0'u_3 \star v_3, x\overline{R}u_3, x\overline{S}v_3, \Delta \quad \Gamma, y0'u_4 \star v_4, x\overline{R}u_4, x\overline{S}v_4, \Delta\end{array}} \ (N\nabla)$$

$$\frac{\Gamma, x_1 \star x_2 1' y_1 \star y_2, \Delta}{\Gamma, x_1 1' y_1, \Delta \quad \Gamma, x_2 1' y_2, \Delta} \ (P1') \qquad \frac{\Gamma, x_1 \star x_2 0' y_1 \star y_2, \Delta}{\Gamma, x_1 0' y_1, x_2 0' y_2, \Delta} \ (N1')$$

$$\frac{\Gamma, xRy, \Delta}{\Gamma, x1'z, xRy, \Delta \quad \Gamma, zRy, xRy, \Delta} \ (1'_a)$$

$$\frac{\Gamma, xRy, \Delta}{\Gamma, xRz, xRy, \Delta \quad \Gamma, z1'y, xRy, \Delta} \ (1'_b)$$

$$\frac{\Gamma, x1'y, \Delta}{\Gamma, y1'x, \Delta, x1'y} \ (Sym)$$

$$\frac{\Gamma, x1'y, \Delta}{\Gamma, x1'z, \Delta, x1'y \quad \Gamma, z1'y, \Delta, x1'y} \ (Trans)$$

$$\frac{\Gamma, x1'y, \Delta}{\Gamma, x \star u1'y \star v, \Delta, x1'y \quad \Gamma, x \star u0'y \star v, \Delta, x1'y} \ (Cut)$$

$$\frac{\Gamma}{\Gamma, x1'y} \ (U)$$

In rule $(P;)$, $z \in IndTerm$ is arbitrary. In rule $(N;)$, $z_1 \in UreVar$ and $z_2 \in CompVar$. In rule $(P\nabla)$, $u, v \in IndTerm$ are arbitrary. In rule $(N\nabla)$, $u_1, u_2, v_1, v_3 \in UreVar$ and $u_3, u_4 v_2, v_4 \in CompVar$. In rules $(1'_a)$ and $(1'_b)$, $z \in IndVar$ is arbitrary. In rule $(Trans)$, $z \in IndTerm$ is arbitrary. In rule (Cut), $u, v \in IndTerm$ are arbitrary. Finally, in rule (U), $x \in UreVar$ and $y \in IndTerm \setminus UreVar$.

Definition 3.41. A formula $t_1 R t_2$ is called *indecomposable* if it satisfies either of the following conditions.

1. $R \in RelVar \cup RelConst$,
2. $R = \overline{S}$ and $S \in RelVar \cup RelConst$,
3. $R \in \{1', 0'\}$.

Definition 3.42. A sequence of formulas Γ is called *indecomposable* if all the formulas in Γ are indecomposable.

Definition 3.43. A sequence of formulas Γ is called *fundamental* if either of the following is true.

1. Γ contains simultaneously the formulas $t_1 R t_2$ and $t_1 \overline{R} t_2$, for some $t_1, t_2 \in IndTerm$ and $R \in RelTerm$.
2. Γ contains the formula $t 1' t$ for some $t \in IndTerm$.

Definition 3.44. Let T be a tree satisfying:

1. Each node contains a finite sequence of fork formulas.
2. If the sequences of fork formulas $\Delta_1, \ldots, \Delta_k$ are the immediate successors of the sequence of fork formulas Γ, then there exists an instance of a rule from FLC of form

$$\frac{\Gamma}{\Delta_1 \quad \Delta_2 \quad \cdots \quad \Delta_k}.$$

Then T is a *proof tree.*

A branch in a proof tree is called *closed* if it ends in a fundamental sequence.

Definition 3.45. A formula $t_1 R t_2$ is *provable* in the calculus FLC iff there exists a proof tree T satisfying:

1. T is finite,
2. $t_1 R t_2$ is the root of T,
3. Each leaf of T contains a fundamental sequence.

Theorem 3.46. The calculus FLC is sound and complete w.r.t. CFREQ^+, i.e., given an equation $R \equiv S$,

$$\models_{\mathsf{CFREQ}^+} R \equiv S \iff \vdash_{FLC} x\,(\overline{R}+S)\cdot(R+\overline{S})\,y, \text{ with } x, y \in CompVar.$$

The calculus FLC can be used in order to prove properties from all the logics studied before, namely, FOLE, modal logics, dynamic logic, and relevant logics. Besides these logics, the calculus FLC has been used in [18] for reasoning in intuitionistic logic, minimal intuitionistic logic, and many intermediate logics.

4. Further Work

As it was mentioned in the abstract, in a future paper we will present a methodology for program construction based on fork algebras. We will use both FOLE and ETFR as specification frameworks, and CFR^+ and CFREQ^+ as our formalism for program construction. We will also incorporate program design decisions in our framework.

5. Conclusions

We presented a study of fork algebras from the algebraic and logical points of view, aiming to settle the basis for the study of program construction within fork algebras. We presented the representation theorem and also how to algebraize classical and non-classical logics using a formalism based on fork algebras.

6. Acknowledgements

Most of this paper was written while Marcelo Frias was visiting the LIFIA, University of La Plata, Argentina. For all the friendship and support he found there, he is deeply grateful.

The authors also wish to thank the anonymous referees, whose advise has certainly contributed to improve this paper.

References

[1] Andréka, H. and Németi, I., *General algebraic logic: A perspective on 'What is logic'*, in D. M. Gabbay (Ed.), *What is a Logical System?*, Vol. 4 of Studies in Logic and Computation, pp. 393–443, Oxford, Clarendon Press, 1994.

[2] Andréka, H. and Németi, I., Sain, I., *Applying algebraic logic to logic*, in Proceedings of AMAST'93, Springer, 1993. Updated version available from `ftp.math-inst.hu/pub/algebraic-logic/meth.dvi.ps`.

[3] Andréka, H. and Sain, I., *Connections between algebraic logic and initial algebra semantics of CF languages.* In B. Dömölki and T. Gergely (Eds.), *Mathematical Logic in Computer Science (Proc. Coll. Salgótarján, 1978)*, Vol. 26 of Colloq. Math. Soc. J. Bolyai, pp. 25–83. Amsterdam, 1981.

[4] Baum, G.A., Frias, M.F., Haeberer, A.M. and Martínez López, P.E., *From Specifications to Programs: A Fork-algebraic Approach to Bridge the Gap*, in Proceedings of MFCS'96, LNCS 1113, Springer-Verlag, pp. 180–191, 1996.
[5] Blok, W., and Pigozzi, D., *Algebraizable Logics*, Memoirs of the A. M. S., Vol 77, 1989.
[6] Brink, C., Kahl, W. and Schmidt, G. (Eds.), *Relational Methods in Computer Science*, Springer, Wien New York, 1997.
[7] Buszkowski, W. and Orlowska, E., *Relational formalization of dependencies in information systems*. In: Orlowska, E. (ed), Reasoning with Incomplete Information. The Rough Set Approach. To appear.
[8] Chin, L. H. and Tarski, A., *Distributive and Modular Laws in the Arithmetic of Relation Algebras*, in University of California Publications in Mathematics. University of California, pp. 341–384, 1951.
[9] De Morgan, A., *On the Syllogism, and Other Logical Writings*, Yale University Press, 1966.
[10] Demri, S. and Orlowska, E., *Logical analysis of demonic nondeterministic programs*, Theoretical Computer Science, to appear.
[11] Demri, S., Orlowska, E. and Rewitzky, I., *Towards reasoning about Hoare relations*. Annals of Mathematics and Artificial Intelligence Vol. 12, pp. 265–289, 1994.
[12] Ebbinghaus H. D., Flum J. and Thomas W., *Mathematical Logic* (Second Edition), Undergraduate texts in mathematics, Springer–Verlag, 1994.
[13] Frias, M. F., Baum, G. A. and Haeberer, A. M., *Representability and Program Construction within Fork Algebras*, to appear in Journal of the IGPL.
[14] Frias, M. F., Baum, G. A., Haeberer, A. M. and Veloso, P. A. S., *A Representation Theorem for Fork Algebras*, (Technical Report) MCC. 29/93, PUC-RJ, August 1993.
[15] Frias, M. F., Baum, G. A., Haeberer, A. M. and Veloso, P. A. S., *Fork Algebras are Representable*, in Bulletin of the Section of Logic, University of Łódź, Vol. 24, No. 2, pp.64–75, 1995.
[16] Frias, M. F., Haeberer, A. M. and Veloso, P. A. S., *A Finite Axiomatization for Fork Algebras*, Journal of the IGPL, Vol. 5, No. 3, pp. 311–319, 1997.
[17] Frias, M. F. and Orlowska E., *Equational Reasoning in Non-Classical Logics*, Journal of Applied Non Classical Logic, to appear, 1997.
[18] 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, to appear, 1997.
[19] Gries, D., *Equational logic as a tool*, LNCS 936, Springer–Verlag, pp. 1–17, 1995.
[20] Gries, D. and Schneider, F. B., *A Logical Approach to Discrete Math.*, Springer–Verlag, 1993.
[21] Gyuris, V., *A Short Proof for Representability of Fork Algebras*, Journal of the IGPL, Vol. 3, No. 5, pp.791–796, 1995.
[22] Haeberer, A.M., Baum, G.A. and Schmidt G., *On the Smooth Calculation of Relational Recursive Expressions out of First-Order Non-Constructive Specifications Involving Quantifiers*, in Proceedings of the International Conference on Formal Methods in Programming and Their Applications, LNCS 735, Springer–Verlag, pp. 281–298, 1993.
[23] Haeberer, A.M. and Veloso, P.A.S., *Partial Relations for Program Derivation: Adequacy, Inevitability and Expressiveness*, in Constructing Programs from Specifications – Proceedings of the IFIP TC2 Working Conference on Constructing Programs from Specifications. North Holland., pp. 319–371, 1991.
[24] Harel, D., *Dynamic Logic*, Handbook of Philosophical Logic, Vol. II, D. Reidel Publishing Company, pp.497–604, 1984.
[25] Halmos, P., *Algebraic Logic*, Chelsea, 1962.
[26] Henkin, L., Monk, D. and Tarski, A., *Cylindric Algebras Part I*, Studies in Logic and the Foundations of Mathematics, vol.64, North–Holland, 1971.
[27] Henkin, L., Monk, D. and Tarski, A., *Cylindric Algebras Part II*, Studies in Logic and the Foundations of Mathematics, vol.115, North–Holland, 1985.

[28] Herment, M. and Orlowska, E., *Handling information logics in a graphical proof editor*, Computational Intelligence, Vol. 11, No. 2, pp. 297-322, 1995.
[29] Jónsson, B., Andréka, H. and Németi, I., *Free Algebras in Discriminator Varieties*, Algebra Universalis, Vol. 28, pp. 401–447, 1991.
[30] Jónsson, B. and Tarski, A. *Boolean Algebras with Operators PART II*, American Journal of Mathematics, Vol. 74, pp. 127–162, 1952.
[31] Kurucz, Á. and Németi, I., *Representability of Pairing Algebras depends on your Ontology*, Journal of Symbolic Logic, to appear.
[32] Löwenheim, L., *Uber Möglichkeiten im Relativkalkul*, Mathematische Annalen, Vol. 76, pp. 447–470, 1915.
[33] Lyndon, R., *The Representation of Relational Algebras*, Annals of Mathematics, (Series 2) Vol. 51, pp. 707–729, 1950.
[34] Lyndon, R., *The Representation of Relational Algebras, II*, Annals of Mathematics, (Series 2), Vol. 63, pp. 294–307, 1956.
[35] Maddux, R., *Topics in Relation Algebras*, Ph.D. Thesis, University of California at Berkeley, 1978.
[36] McKenzie, R. N. W., *The Representation of Integral Relation Algebras*, Michigan Mathematical Journal. Vol. 17, pp. 279–287, 1970.
[37] Mikulás, S., Sain, I. and Simon, A., *Complexity of the Equational Theory of Relational Algebras with Projection Elements.* Bulletin of the Section of Logic, University of Lódź, Vol. 21, No. 3, pp. 103–111, 1992.
[38] Németi, I., *Free Algebras and Decidability in Algebraic Logic*, D. Sc. dissertation (in hungarian), Math. Inst. Hungar. Acad. Sci., 1986.
[39] Németi, I., *Algebraizations of quantifier logics.* Studia Logica, Vol. 50, pp. 485–569, 1991. Updated version available from `ftp.math-inst.hu/pub/algebraic-logic/survey.ps`.
[40] Németi, I., *Ontology can turn negative results to positive*, Bulletin of the Section of Logic, University of Lódź, Vol. 25, No. 1, pp.29–40, 1996.
[41] Németi, I., *Strong Representability of Fork Algebras, a Set Theoretic Foundation*, Journal of the IGPL, Vol. 5, No. 1, pp. 3–23, 1997.
[42] Sain, I. and Németi, I., *Fork Algebras in Usual and in Non-well founded Set Theories*, Studia Logica library, a special volume dedicated to the memory of Helena Rasiowa, to appear.
[43] Orlowska, E., *Relational interpretation of modal logics.* In: Andreka,H., Monk,D. and Nemeti,I. (eds) Algebraic Logic. Colloquia Mathematica Societatis Janos Bolyai vol 54, North Holland, pp. 443–471, 1988.
[44] Orlowska, E., *Relational proof system for relevant logics.* Journal of Symbolic Logic, Vol. 57, pp. 1425–1440, 1992.
[45] Orlowska,E., *Relational semantics for nonclassical logics: Formulas are relations.* In: Wolenski,J. (ed) Philosophical Logic in Poland. Kluwer, pp. 167–186, 1994.
[46] Orlowska, E., *Relational proof systems for modal logics.* In: Wansing,H. (ed) Proof Theory of Modal Logics. Kluwer, pp. 55–77, 1996.
[47] Peirce, Ch. S., *Collected Papers*, Harvard University Press, Cambridge, 1933.
[48] Rasiowa, H. and Sikorski, R. *The Mathematics of Metamathematics.* Polish Science Publishers, Warsaw, 1963.
[49] Schmidt, G. and Ströhlein, T., *Relations and Graphs*, EATCS Monographs in Theoretical Computer Science, Springer–Verlag, 1993.
[50] Schröder, E. F. W. K., *Vorlesungen über die Algebra der Logik (exacte Logik)*, Volume 3, "Algebra und Logik der Relative", part I, Leipzig, 1895.
[51] Tarski, A., *On the Calculus of Relations*, Journal of Symbolic Logic, Vol. 6, pp. 73–89, 1941.
[52] Tarski, A., *Some metalogical results concerning the calculus of relations*, Journal of Symbolic Logic, Vol. 18, pp. 188–189, 1953.
[53] Tarski, A. and Givant, S.,*A Formalization of Set Theory without Variables*, A. M. S. Coll. Pub., Vol. 41, 1987.