Factor Varieties and Symbolic Computation

Antonino Salibra
Università Ca'Foscari,
Venezia
salibra@dsi.unive.it

Giulio Manzonetto
Université Paris 13, LIPN,
CNRS UMR 7030, France
giulio.manzonetto@lipn.fr

Giordano Favro
Università Ca'Foscari,
Venezia
favro@dsi.unive.it

January 18, 2016

Abstract

We propose an algebraization of classical and non-classical logics, based on factor varieties and decomposition operators. In particular, we provide a new method for determining whether a propositional formula is a tautology or a contradiction. This method can be automatized by defining a term rewriting system that enjoys confluence and strong normalization. This also suggests an original notion of logical gate and circuit, where propositional variables becomes logical gates and logical operations are implemented by substitution. Concerning formulas with quantifiers, we present a simple algorithm based on factor varieties for reducing first-order classical logic to equational logic. We achieve a completeness result for first-order classical logic without requiring any additional structure.

Introduction

Algebraic logic investigates the connections between a logic and algebraic properties of its corresponding class of algebras. The origin of modern algebraic logic goes back to Tarski's 1935 paper [24], where he introduced the Tarski-Lindenbaum algebra as a tool for establishing the correspondence between classical propositional logic and Boolean algebras. In this context the tautologies coincide with those formulas equivalent to the truth value "true". Subsequently, a number of different propositional logics were algebraized in this way, the most important being the intuitionistic logic and the multi-valued logics of Post and of Łukasiewicz. The problem of formulating the notion of an algebraizable logic in full generality has been addressed by Blok and Pigozzi in [3], where they showed that, if a logic $\mathcal{L}$ is algebraizable, then there exists a unique quasi-variety $\mathcal{K}$ of algebras which coincides with the equivalent algebraic semantics of $\mathcal{L}$. This means that the consequence relation $\vdash_{\mathcal{L}}$ over $\mathcal{L}$ and the equational consequence relation $\models_{\mathcal{K}}$ over $\mathcal{K}$ are interpretable in one another in a certain (strong) sense (see [3, Def. 2.8 and Thm 2.15]).

The problem of algebraizing predicate logics is much more complicated because of the variable binding properties of the quantifiers. On the one hand, the algebraization of classical predicate logic led Tarski to the definition of cylindric algebras [13] and Halmos to the notion of polyadic Boolean algebras [12]. In practice these algebras are difficult to manipulate because they are endowed with operators representing the quantifiers in the algebraic structure and this complicates their theory.

On the other hand, much work in computer science has been focused on reducing first-order logic to equational logic and, more recently, to term rewriting systems. In [16] McKenzie proved that for every sentence Φ in first-order classical logic there is an equation Φ' in a suitable algebraic language such that Φ has non-trivial models of a given cardinality κ exactly when Φ' does. In his 1992 paper [5], Burris made a substantial advance by using discriminator varieties [27]. A discriminator variety V is characterized by a quaternary term s that realizes the switching function on any subdirectly irreducible member of V [6, Def. 7.3]:

$$\mathsf{s}(a,b,c,d) = \begin{cases} c & \text{if } a = b, \\ d & \text{otherwise.} \end{cases}$$

Thanks to this switching function, Burris has shown that discriminator varieties have unitary unification, which is at the basis of resolution theorem provers and of the Knuth-Bendix method for finding rewriting systems. He was also able to combine McKenzie's analysis of satisfiability with a standard reduction of $\Psi_1,\dots,\Psi_n \models \Phi$ to a set of unsatisfiable sentences in prenex normal form. Indeed, given a formula Φ and a finite set T of formulas, one can prove that $T \vdash \Phi$ holds by showing $T \models \Phi$ which is, in turn, equivalent to showing that $\Sigma := T \cup \{\neg\Phi\}$ has no models. In [5], Burris shows how to define a set E of equations in the equational logic of a given discriminator variety such that Σ has no models of cardinality greater than 1 exactly when E has no non-trivial models. To show that E has no non-trivial models it is enough to derive the identity $x = y$ from E. This approach is however not applicable to propositional logic and the process of deriving $x = y$ is not easily automatable because of the complexity of the axioms in the system.

In this paper we provide a new method for extracting the logical content of a formula: in particular, it allows to determine whether a propositional formula is a tautology or a contradiction. This method is general enough to be applied to any finite multi-valued matrix logic, and we feel that it can be extended to infinite logics, like fuzzy logic [11] and probabilistic logic [19]. In our approach, rather than using the switching function of discriminator varieties, we use the decomposition operators characterizing the factor varieties. Indeed, the very definition of the switching function s suggests a natural move. One could meaningfully wonder what happens if the set $\{\mathsf{t},\mathsf{f}\}$ of classical truth values is substituted by an arbitrary set $V = \{v_1, \dots, v_p\}$ and the role of the equality in the definition of s is played by a generic (multi-valued) relation $R : A^n \to V$.

In other words, we could define an *R-factor function* on a set A as a function $f_R : A^{n+p} \to A$ such that:

$$\forall b_1...b_p.f_R\,(a_1, \dots, a_n, b_1, \dots, b_p) = b_i \text{ iff } R(a_1, \dots, a_n) = v_i.$$

These R-factor functions are at the core of our definition of factor variety, which generalizes not only the notion of discriminator variety, but also the one of factor variety as it was introduced in [22]. Indeed, in that paper Salibra *et al.* only consider the R-factor function corresponding to an arbitrary, but fixed, binary relation R.

Given a relational type ν, we define a *factor variety* as a variety V having an *R-factor term* $f_R(x_1, \dots, x_n, \xi_1, \dots, \xi_p)$ for each n-ary relation symbol $R \in \nu$, that is a term such that the p-ary function $f_R^{\mathbf{A}}(\vec{a}, -_1, \dots, -_p)$ is a decomposition operation (see Definition 2) for all $\mathbf{A} \in \mathsf{V}$ and all $\vec{a} \in A^n$. The factor variety V is generated by the class V_{fa} of all *factor algebras* $\mathbf{A}$, that are algebras such that, for every $R \in \nu$, the decomposition operator $f_R^{\mathbf{A}}\,(\vec{a}, -_1, \dots, -_p)$ is trivial (i.e. it is an R-factor function on A). The class V_{fa} is in bijective correspondence with the class of (proper) ν-structures. Once associated a factor algebra with every structure, we translate the formula Φ into an algebraic term Φ^*. Under this translation, each truth value v_i becomes a fresh variable ξ_i, each relation R is sent to the corresponding R-factor term and logical connectives are translated via suitable substitutions. If v_p represents the truth value "true", we then characterize the logical truth of a universal formula Φ through the equation $\Phi^* = \xi_p$ in the factor variety.

Concerning formulas with quantifiers, we present a new method, simplifying Burris-McKenzie's one, for reducing first-order classical logic to equational logic. This approach allows to achieve a completeness result without requiring any additional structure. However, it cannot be generalized further since it relies on specific properties of classical logic, namely the fact that all formulas can be written in prenex normal form, Skolemization and logical completeness.

Since the axioms of a factor variety are very simple, the process of checking whether $\Phi^* = \xi_p$ holds in such a variety can be automatized in the propositional case by defining a confluent and terminating term rewriting system. The problem of showing $\Phi^* = \xi_p$ is then reduced to the problem of checking whether the normal form of Φ^* is ξ_p. The analysis of the computational complexity of this system is left for future works.

Our algebraic framework also suggests a new notion of logic circuits, that we call *factor circuit*s and are based on components that we call *D-gates*. Rather than implementing a logical connective, a D-gate represents a decomposition operator of some algebra $\mathbf{A}$ belonging to a factor variety. A propositional D-gate has a selector switch and p input ports. When the selector switch is connected to a propositional variable P, the D-gate implements the operator f_P and its input ports correspond to the variables $\vec{\xi}$ in $f_P(\vec{\xi})$. So, the wires that are used for connecting D-gates do not carry signals representing truth values, but rather elements of the algebra $\mathbf{A}$.

We believe that these applications are promising as McKenzie's and Burris's works appear to have been largely overlooked by the communities working on proof assistants. This might be due to the fact that it is not readily apparent how to manipulate the axioms of a discriminator variety. In future works, we plan to investigate unitary unification for factor varieties and extensions of our rewriting system to relational types. It would be interesting to combine our rewriting system with the results of Section 7 for reducing first-order logic inference to a rewriting process. The integration of our methods in theorem provers also deserves to be investigated.

Outline. Section 1 contains some preliminary notions of universal algebra and logics. In Section 2 we discuss classical logic as a motivating example. Factor algebras and factor varieties are introduced in Section 3. Section 4 is devoted to present our algebraization of multi-valued logics. In Section 5 we show how this method can be automatized via a suitable term rewriting system. In Section 6 we introduce the factor circuits and we compare them with the usual boolean circuits. Finally, in Section 7 we explain the new algorithm for reducing first-order classical logic to equational logic, and we prove a completeness theorem.

1 Preliminaries

We refer to [6] for universal algebra and to [3] for logics.

1.1 Algebras, Varieties and Factor Congruences

Let ν be a relational type, that is a family of function/relation symbols with arity. We denote by ν_n the set of symbols in ν having arity n. Function symbols will be denoted by lower case letters f, g, h, while relation symbols by capital letters $R, R_1, R_2, \dots$ Relation symbols of arity 0 are called *propositional variables* and are denoted by P, Q. We write $f \in \nu$ (resp. $R \in \nu$) to indicate that f is a function symbol (resp. R is a relation symbol) of type ν.

An algebraic type is a relational type without relation symbols. If τ is an algebraic type, an algebra $\mathbf{A}$ of type τ is called *a τ-algebra*. $\mathrm{Con}(\mathbf{A})$ is the lattice of all congruences on $\mathbf{A}$. The trivial congruences $\Delta = \{(x, x) : x \in A\}$ and $\nabla = A \times A$ constitute the bottom and the top elements of $\mathrm{Con}(\mathbf{A})$, respectively. Given $a, b \in A$, we write $\vartheta(a, b)$ for the *principal congruence generated by a and b*, that is for the smallest congruence relating them.

Definition 1. *A family $(\varphi_i)_{i\in I}$ of congruences on $\mathbf{A}$ is a* family of complementary factor congruences *if the function*

$$f : \mathbf{A} \to \textstyle\prod_{i\in I}(\mathbf{A}/\varphi_i)$$

defined by $f(a) = (a/\varphi_i)_{i\in I}$ is an isomorphism. When $|I| = 2$, we say that (φ_1, φ_2) is a pair of complementary factor congruences.

A *factor congruence* is any congruence which belongs to a family of complementary factor congruences.

Proposition 1. *A family $(\varphi_i)_{i\in I}$ of congruences on $\mathbf{A}$ is a family of complementary factor congruences exactly when:*

1. *$\bigcap_{i\in I} \varphi_i = \Delta$;*
2. *$\forall a \in A^I$, there is $u \in A$ such that $a_i \, \varphi_i \, u$, for all $i \in I$.*

Therefore (φ_1, φ_2) is a pair of complementary factor congruences if and only if $\varphi_1 \cap \varphi_2 = \Delta$ and $\varphi_1 \circ \varphi_2 = \nabla$. The pair (Δ, ∇) corresponds to the product $\mathbf{A} \cong \mathbf{A} \times \mathbf{1}$, where $\mathbf{1}$ is the singleton algebra; obviously $\mathbf{1} \cong \mathbf{A}/\nabla$ and $\mathbf{A} \cong \mathbf{A}/\Delta$. The set of factor congruences of $\mathbf{A}$ is not, in general, a sublattice of $\mathrm{Con}(\mathbf{A})$.

We say that an algebra $\mathbf{A}$ is: (i) *subdirectly irreducible* if the lattice $\mathrm{Con}(\mathbf{A})$ has a unique atom; (ii) *simple* if $\mathrm{Con}(\mathbf{A}) = \{\Delta, \nabla\}$; (iii) *directly indecomposable* if it admits only the two trivial factor congruences. Any simple algebra is subdirectly irreducible and any subdirectly irreducible algebra is directly indecomposable.

A class V of τ-algebras is a *variety* if it is closed under subalgebras, direct product and homomorphic images. By Birkhoff theorem a class of algebras is variety if and only if it is an equational class.

Factor congruences can be characterized in terms of certain algebra homomorphisms called *decomposition operators* and acting on sequences (see [17, Def. 4.32] for more details).

Given a set A and a set of indices I we define an *I-sequence $\vec{x}$ on A* as a function $\vec{x} : I \to A$. For every index $i \in I$ and element $a \in A$ we denote by $\vec{x}[a/i]$ the I-sequence which coincides with $\vec{x}$, except on i, where it takes the value a. Given $a \in A$ we let a^I denote the constant sequence taking value a for all indices $i \in I$.

Definition 2. *A* decomposition operator *on an algebra $\mathbf{A}$ is a function $f : A^I \to A$ satisfying the following conditions:*

(D1) $f(a^I) = a$, for all $a \in A$;

(D2) $f(f(a_{ij})_{j\in I})_{i\in I} = f(a_{ii})_{i\in I}$;

(D3) f is an algebra homomorphism from $\mathbf{A}^I$ to $\mathbf{A}$.

If I is finite, the axioms (D1)-(D3) can be equationally expressed.

There is a bijective correspondence between families of complementary factor congruences and decomposition operators, and thus, between decomposition operators and factorizations.

Proposition 2. *Any decomposition operator $f : \mathbf{A}^I \to \mathbf{A}$ on an algebra $\mathbf{A}$ induces a family of complementary factor congruences $(\varphi_i)_{i\in I}$ where each $\varphi_i \subseteq A \times A$ is defined by:*

$$a \ \varphi_i \ b \ \text{ if and only if } \ f(a^I[b/i]) = a.$$

Conversely, any family $(\varphi_i)_{i\in I}$ of complementary factor congruences induces a decomposition operator f on $\mathbf{A}$:

$$f(\vec{x}) = u \ \text{ if and only if } \ x_i \, \varphi_i \, u, \ \text{for all } i \in I.$$

Indeed, it is possible to prove that such an element u is unique.

1.2 Matrix Logics

A matrix logic $\mathcal{L}$ is defined by specifying the logical connectives, the set of truth values, among which there is a "designated value" representing the traditional truth value "*verum*", and the truth functions that interpret the logical connectives.

We start by taking an algebraic type τ that represents the set of logical connectives together with their arity.

Definition 3. *A* logical τ-matrix *is a pair* $(\mathbf{V}, \mathtt{t})$ *where* $\mathbf{V}$ *is a finite* τ*-algebra and* $\mathtt{t}$ *is an element of* V*.*

When τ is clear from the context, we just speak of a *logical matrix.* The elements of the universe V are called *truth values* and are denoted by $v_1, \dots, v_p$, while $\mathtt{t}$ is called the *designated element.*

We write Pvar for the set of propositional variables. *Propositional formulas* ϕ *of type* τ are defined by induction as follows:

$$\phi, \psi \ ::= \ P \mid o(\phi_1, \dots, \phi_n) \text{ where } P \in \text{Pvar and } o \in \tau_n.$$

A *truth assignment* is any function $\mathcal{I} : \text{Pvar} \to V$. Given a propositional formula ϕ, its *interpretation in* $\mathbf{V}$ *w.r.t.* $\mathcal{I}$ is the element $[\![\phi]\!]^{\mathcal{I}}$ inductively defined by (for $P \in \text{Pvar}, o \in \tau_n$):

1. $[\![P]\!]^{\mathcal{I}} = \mathcal{I}(P)$;
2. $[\![o(\phi_1, \dots, \phi_n)]\!]^{\mathcal{I}} = o^{\mathbf{V}}([\![\phi_1]\!]^{\mathcal{I}}, \dots, [\![\phi_n]\!]^{\mathcal{I}})$.

We say that a propositional formula ϕ is a *tautology* (resp. a *contradiction*) whenever $[\![\phi]\!]^{\mathcal{I}} = \mathtt{t}$ (resp. $[\![\phi]\!]^{\mathcal{I}} \neq \mathtt{t}$) for all truth assignments $\mathcal{I}$.

Definition 4. *The* propositional matrix logic $\mathcal{L}$ *induced by a logical* τ*-matrix* $(\mathbf{V}, \mathtt{t})$ *is the logic whose semantics is defined as follows:* $\psi_1, \dots, \psi_n \models_{\mathcal{L}} \phi$ *if and only if, for every truth assignment* $\mathcal{I}$*,* $[\![\phi]\!]^{\mathcal{I}} = \mathtt{t}$ *whenever* $[\![\psi_i]\!]^{\mathcal{I}} = \mathtt{t}$ *for all* i*.*

Example 1. *We provide some examples of matrix logics.*

1. ***Classical Logic*** $\mathcal{C}$**.** *The type of logical connectives is* $\tau = \{\wedge, \vee, \neg, \mathtt{f}, \mathtt{t}\}$*, the logical matrix is* $(\mathbf{2}, \mathtt{t})$ *where* $\mathbf{2}$ *is the two elements boolean algebra of truth values* $\mathtt{f}, \mathtt{t}$ *and* $\mathtt{t}$ *is the designated element. As usual, we consider* $\mathtt{f} < \mathtt{t}$.

2. ***The*** n***-valued logics*** *under consideration (Łukasiewicz, Gödel and Post Logics) have a totally ordered set* $0 < \frac{1}{n-1} < \frac{2}{n-1} < \cdots < \frac{n-2}{n-1} < 1$ *of truth values,* 1 *as designated element, and join and meet are defined by* $a \vee b = \max\{a, b\}$ *and* $a \wedge b = \min\{a, b\}$*. These logics only differ for the definition of negation and implication, which is not present in Post Logic.*

- ***Łukasiewicz Logic*** $\mathcal{L}\!\!-_n$*:*

$$\neg a = 1 - a; \qquad a \to b = \min(1, 1 - a + b).$$

- ***Gödel Logic*** $\mathcal{G}_n$*:*

$$a \to b = \begin{cases} 1 & \textit{if } a \leq b \\ b & \textit{if } a > b \end{cases} \qquad \neg a = \begin{cases} 1 & \textit{if } a = 0 \\ 0 & \textit{if } a \neq 0. \end{cases}$$

- ***Post Logic*** $\mathcal{P}_n$*:*

$$\neg a = \begin{cases} a - \frac{1}{n-1} & \textit{if } a \neq 0 \\ 1 & \textit{if } a = 0. \end{cases}$$

The n-valued Gödel logics are *superintuitionistic logics*, which means they are logics between intuitionistic and classical logics. Superintuitionistic logics form a complete lattice whose unique coatom is the 3-valued Gödel Logic $\mathcal{G}_3$. As shown by Gödel in [10], the intuitionistic logic is not definable by a finite logical matrix.

Quantified Matrix Logics. In the rest of the section, we consider fixed a countably infinite set Var of individual variables (that will be denoted by x, y, z, w), an algebraic type τ of logical connectives, a logical τ-matrix $(\mathbf{V}, \mathtt{t})$ and a relational type ν containing both function and relation symbols with arity.

Terms of type ν, or ν*-terms*, are defined as usual from individual variables in Var and function symbols in ν. The set of all ν-terms will be denoted by $\mathfrak{T}_\nu$ and its elements by $t, t_1, t_2, \dots$

Well formed formulas are defined by the following grammar, where $R \in \nu_m$ is a relation symbol, $o \in \tau_n$ is a logical connective and $t_1, \dots, t_m$ are ν-terms:

$$\Phi, \Psi ::= R(t_1, \dots, t_m) \mid o(\Phi_1, \dots, \Phi_n) \mid \forall x.\Phi \mid \exists x.\Phi$$

We say that a formula Φ is: (i) a *sentence* if it has no free variables; (ii) *open* if it is quantifier-free; (iii) *in prenex form* if it has the shape $Q_1 x_1 \dots Q_n x_n.\Psi$ where $Q_i \in \{\forall, \exists\}$ and Ψ is an open formula (called *the matrix of* Φ); (iv) *universal* if it is in prenex form and all its quantifiers Q_i are universal.

Definition 5. *A* ν-structure $\mathcal{S}$ on V *is given by* $(S, g^{\mathcal{S}}, R^{\mathcal{S}})_{g,R \in \nu}$ *where* S *is a set,* $g^{\mathcal{S}} : S^k \to S$ *is a* k*-ary operation for any function symbol* $g \in \nu_k$ *and* $R^{\mathcal{S}} : S^n \to V$ *is a function for any relation symbol* $R \in \nu_n$*. We say that* $\mathcal{S}$ *is* proper *whenever* $|S| > 1$.

We let $\mathsf{Str}^*_{\nu,V}$ be the class of all proper ν-structures on V.

Given a ν-structure $\mathcal{S}$ on V as above, an *environment* is a function $\rho : \mathrm{Var} \to S$. The interpretation $[\![t]\!]^{\mathcal{S}}_{\rho}$ of a term t is defined as usual. To interpret the quantifiers we assume the set V of truth values to be a finite lattice, whose top element is the designated element t. The *interpretation of a formula* Φ *in* $\mathcal{S}$ *w.r.t.* ρ is then defined inductively as follows (for $R \in \nu_m$, $\vec{t} \in \mathcal{T}^m_{\nu}$ and $o \in \tau_n$):

1. $[\![R(t_1,\dots,t_m)]\!]^{\mathcal{S}}_{\rho} = R^{\mathcal{S}}([\![t_1]\!]^{\mathcal{S}}_{\rho},\dots,[\![t_m]\!]^{\mathcal{S}}_{\rho})$;
2. $[\![o(\Phi_1,\dots,\Phi_n)]\!]^{\mathcal{S}}_{\rho} = o^{\mathbf{V}}([\![\Phi_1]\!]^{\mathcal{S}}_{\rho},\dots,[\![\Phi_n]\!]^{\mathcal{S}}_{\rho})$;
3. $[\![\forall x.\Phi]\!]^{\mathcal{S}}_{\rho} = \bigwedge_{a\in S}[\![\Phi]\!]_{\rho[a/x]}$;
4. $[\![\exists x.\Phi]\!]^{\mathcal{S}}_{\rho} = \bigvee_{a\in S}[\![\Phi]\!]_{\rho[a/x]}$.

We write $\mathcal{S} \models_{\rho} \Phi$ whenever $[\![\Phi]\!]^{\mathcal{S}}_{\rho} = \mathsf{t}$. We say that a formula Φ is a *logical truth* if $\mathcal{S} \models_{\rho} \Phi$ for every structure $\mathcal{S}$ and environment ρ.

A class S of ν-structures is called *universal* if it can be axiomatized by universal formulas.

Definition 6. *The* quantified matrix logic $\mathcal{QL}$, *induced by a logical* τ*-matrix* $(\mathbf{V},\mathsf{t})$ *and a relational type* ν, *is the logic whose semantics is defined as:* $\Psi_1,\dots,\Psi_n \models_{\mathcal{QL}} \Phi$ *if and only if, for every structure* $\mathcal{S}$ *and environment* ρ, $[\![\Phi]\!]^{\mathcal{S}}_{\rho} = \mathsf{t}$ *whenever* $[\![\Psi_k]\!]^{\mathcal{S}}_{\rho} = \mathsf{t}$ *for all* k.

The *propositional translation* of a formula Φ is the propositional formula Φ^{p} defined as:

- $R(t_1,\dots,t_m)^{\mathrm{p}} = P_R$, where $P_R \in \mathrm{Pvar}$;
- $o(\Phi_1,\dots,\Phi_n)^{\mathrm{p}} = o(\Phi_1^{\mathrm{p}},\dots,\Phi_n^{\mathrm{p}})$;
- $(\forall x.\Phi)^{\mathrm{p}} = (\exists x.\Phi)^{\mathrm{p}} = \Phi^{\mathrm{p}}$.

In classical logic with equality, there exists an equality symbol which is propositionally translated by setting $(t_1 = t_2)^{\mathrm{p}} = \mathsf{t}$.

Lemma 1. *A formula* Φ *is true in all singleton structures if and only if its propositional translation* Φ^{p} *is a tautology.*

Proof. Let $\mathcal{S}$ be a structure over $\{s\}$, $\rho : \mathrm{Var} \to \{s\}$ be its unique environment and $\mathcal{I} : \mathrm{Pvar} \to V$ be a truth assignment such that $\mathcal{I}(P_R) = v_i$ if and only if $R^{\mathcal{S}}(s,\dots,s) = v_i$. It is possible to prove that $[\![\Phi]\!]^{\mathcal{S}}_{\rho} = [\![\Phi^{\mathrm{p}}]\!]^{\mathcal{I}}$ by induction on the complexity of Φ. $\square$

2 The Motivating Example

In the following sections we provide a new method for extracting the logical content of a propositional formula ϕ and determine whether ϕ is a tautology or a contradiction.

As a motivating example, we consider the classical logic $\mathcal{C}$ as defined in Example 1.1. Our approach consists of two steps.

Step 1. The first step consists in defining a translation $(\cdot)^*$ sending propositional formulas into algebraic terms. Under this translation, the truth values f,t become new algebraic variables $\xi_{\mathsf{f}},\xi_{\mathsf{t}}$. A propositional variable P becomes a binary operator $P(-,-)$. A propositional formulas ϕ is translated inductively into an algebraic term ϕ^* on the variables $\xi_{\mathsf{f}},\xi_{\mathsf{t}}$. To simplify the notation, we will write $\phi^*(t_0,t_1)$ for the substitution $\phi^*\{t_0/\xi_{\mathsf{f}}, t_1/\xi_{\mathsf{t}}\}$.

$$
\begin{array}{rcl}
P^* & = & P(\xi_{\mathsf{f}},\xi_{\mathsf{t}});\\
(\neg\phi)^* & = & \phi^*(\xi_{\neg\mathsf{f}},\xi_{\neg\mathsf{t}}) = \phi^*(\xi_{\mathsf{t}},\xi_{\mathsf{f}});\\
(\phi\wedge\psi)^* & = & \psi^*(\phi^*(\xi_{\mathsf{f}\wedge\mathsf{f}},\xi_{\mathsf{f}\wedge\mathsf{t}}),\phi^*(\xi_{\mathsf{t}\wedge\mathsf{f}},\xi_{\mathsf{t}\wedge\mathsf{t}}));\\
 & = & \psi^*(\phi^*(\xi_{\mathsf{f}},\xi_{\mathsf{f}}),\phi^*(\xi_{\mathsf{f}},\xi_{\mathsf{t}}));\\
(\phi\vee\psi)^* & = & \psi^*(\phi^*(\xi_{\mathsf{f}\vee\mathsf{f}},\xi_{\mathsf{f}\vee\mathsf{t}}),\phi^*(\xi_{\mathsf{t}\vee\mathsf{f}},\xi_{\mathsf{t}\vee\mathsf{t}}));\\
 & = & \psi^*(\phi^*(\xi_{\mathsf{f}},\xi_{\mathsf{t}}),\phi^*(\xi_{\mathsf{t}},\xi_{\mathsf{t}}));\\
(\phi\to\psi)^* & = & (\neg\phi\vee\psi)^* = \psi^*(\phi^*(\xi_{\mathsf{t}},\xi_{\mathsf{f}}),\phi^*(\xi_{\mathsf{t}},\xi_{\mathsf{t}})).
\end{array}
$$

Connectives are therefore implemented through substitutions and Boolean operations on the indices of $\xi_{\mathsf{f}},\xi_{\mathsf{t}}$. The above translation determines a congruence $\sim^*$ on the set of propositional formulas by setting $\phi \sim^* \psi$ if and only if $\phi^* = \psi^*$. For instance, we have $\neg\neg\phi \sim^* \phi$ and $(\phi_1\vee\phi_2)\vee\phi_3 \sim^* \phi_1\vee(\phi_2\vee\phi_3)$, but $\phi_1\vee\phi_2 \not\sim^* \phi_2\vee\phi_1$. This defines a non-commutative intermediate logic $\mathcal{C}_{\mathrm{int}}$ which is strictly weaker than classical logic.

For example, we have $(\neg P\vee P)^* = P(P(\xi_{\mathsf{t}},\xi_{\mathsf{f}}),P(\xi_{\mathsf{t}},\xi_{\mathsf{t}}))$ and $(P\vee\neg P)^* = P(P(\xi_{\mathsf{t}},\xi_{\mathsf{t}}),P(\xi_{\mathsf{f}},\xi_{\mathsf{t}}))$, hence $\neg P\vee P \not\sim^* P\vee\neg P$.

Step 2. To retrieve classical logic, we need to give each P the operational behavior of a binary decomposition operator:

(D1) $P(x,x) = x$;

(D2) $P(P(x,y),P(w,z)) = P(x,z)$;

(D3) $P(Q(x,y),Q(w,z)) = Q(P(x,w),P(y,z))$, for every propositional variable $Q \in \mathrm{Pvar}$.

Both truth values and propositional variables, that are static objects in the logic $\mathcal{C}$, become dynamic entities after the translation: indeed variables $\xi_\mathsf{f}, \xi_\mathsf{t}$ can receive substitutions and operators $P(-,-)$ induce decompositions. We prove that the formula ϕ is a tautology (resp. a contradiction) if and only if $\phi^* = \xi_\mathsf{t}$ (resp. $\phi^* = \xi_\mathsf{f}$) is provable using the axioms $(D1)$-$(D3)$ above, see Corollary 1.

For example, the formula $\neg P \vee P$ is a tautology since

$$(\neg P \vee P)^* = P(P(\xi_\mathsf{t},\xi_\mathsf{f}),P(\xi_\mathsf{t},\xi_\mathsf{t})) =_{\mathrm{D2}} P(\xi_\mathsf{t},\xi_\mathsf{t}) =_{\mathrm{D1}} \xi_\mathsf{t}.$$

In Section 5, we give this process a computational flavor by showing that, by orienting the equations $(D1)$-$(D3)$ from left to right, we obtain a confluent term rewriting system. Moreover, by well-ordering the propositional variables we can prevent $(D3)$ from looping and ensure termination. This approach also suggests a new notion of circuit, described in Section 6, which is based on components that we call "decomposition gates" and behave like the decomposition operators of an algebra in a factor variety.

The translation above can be also generalized to first-order formulas by transforming an n-ary relation symbol R into an operator $R(-_1,\dots,-_{n+2})$ of arity $n+2$ (since there are two truth values), which is a decomposition operator in the last two coordinates. Open formulas can be therefore inductively translated, as in Step 1, into algebraic terms on the variables $\mathrm{Var} \cup \{\xi_\mathsf{f},\xi_\mathsf{t}\}$, assuming the following translation of atomic formulas:

$$R(t_1,\dots,t_n)^* = R(t_1,\dots,t_n,\xi_\mathsf{f},\xi_\mathsf{t}).$$

Such a translation provides a bijective correspondence between first-order theories axiomatized by universal sentences without equality and varieties of factor algebras axiomatized by identities such as $\Phi^* = \xi_\mathsf{t}$. In presence of equality, the situation becomes more subtle. Intuitively, the problem is that factor algebras can only capture correctly *proper* structures. In other words, a formula like $\forall x \exists y.\neg(x=y)$, which is true in all proper structures, but fails in any singleton structure, will be seen as a logical truth in any factor algebra. Hence, to see whether the formula Φ is actually a logical truth, one also need to verify that its propositional translation Φ^{P} is a tautology and apply Lemma 1.

3 Factor Algebras and Factor Varieties

In this section we are going to introduce factor algebras and factor varieties. We consider fixed a relational type ν and a logical τ-matrix $(\mathbf{V},\mathsf{t})$ where $V = \{v_1,\dots,v_p\}$. We write $\hat\nu$ for the smallest algebraic type containing: a function symbol $g \in \hat\nu_k$ for each function symbol $g \in \nu_k$; a function symbol $f_R \in \hat\nu_{n+p}$ for each relation symbol $R \in \nu_n$. Remark that a relation R of arity n is transformed into a function f_R having p additional arguments.

Definition 7. *A $\hat\nu$-factor algebra $\mathbf{A} = (A, g^{\mathbf{A}}, f_R^{\mathbf{A}})_{g,R\in\nu}$ is a $\hat\nu$-algebra such that, for all $f_R \in \hat\nu_{n+p}$ and $\vec a \in A^n$ there exists an index $i \in [1..p]$ such that:*

$$\forall \xi_1 \dots \xi_p . f_R(\vec a, \xi_1,\dots,\xi_p) = \xi_i. \tag{3.1}$$

The class $\mathsf{FA}_{\hat\nu}$ of all $\hat\nu$-factor algebras is a universal class, i.e. it is closed under subalgebras and ultraproducts. We write $\mathsf{FA}^*_{\hat\nu}$ for the class of *proper* factor algebras (where proper means that $|A|>1$).

Given a $\hat\nu$-factor algebra $\mathbf{A}$, the *algebraic reduct* of $\mathbf{A}$ is the algebra $\mathrm{Alg}(\mathbf{A}) = (A, g^{\mathbf{A}})_{g\in\nu}$.

Definition 8. *We associate with every proper factor algebra $\mathbf{A}$ a proper structure $\mathrm{Str}(\mathbf{A})$ having the same algebraic reduct, and relations defined by (for all $f_R \in \hat\nu_{n+p}$ and $\vec a \in A^n$):*

$$R^{\mathrm{Str}(\mathbf{A})}(\vec a) = v_k \text{ iff } \forall \xi_1,\dots,\xi_p . f_R^{\mathbf{A}}(\vec a,\xi_1,\dots,\xi_p) = \xi_k.$$

Conversely, we associate with every proper structure $\mathcal{S}$ a proper factor algebra $\mathrm{Fa}(\mathcal{S})$ having the same algebraic reduct as $\mathcal{S}$ and whose functions f_R ($R \in \nu_n$) are defined as follows:

$$f_R^{\mathrm{Fa}(\mathcal{S})}(\vec a,\xi_1,\dots,\xi_p) = \xi_k \text{ iff } R^{\mathcal{S}}(\vec a) = v_k.$$

In particular, we have $\mathrm{Str}(\mathrm{Fa}(\mathcal{S})) = \mathcal{S}$ and $\mathrm{Fa}(\mathrm{Str}(\mathbf{A})) = \mathbf{A}$.

Note that the above correspondence fails on singleton structures. Let $\mathcal{S},\mathcal{T}$ be two structures over $\{*\}$ with a relation symbol R such that $R^{\mathcal{S}}(*) = \mathsf{t}$ but $R^{\mathcal{T}}(*) \neq \mathsf{t}$. The structures $\mathcal{S}$ and $\mathcal{T}$ are not isomorphic, but correspond to the same trivial factor algebra.

3.1 Congruences of Factor Algebras

This technical section, that can be skipped on a first reading, is devoted to analyze some properties of the congruences on factor algebras. Let us consider a relational type ν and a $\hat{\nu}$-factor algebra $\mathbf{A}$. Remember that p is the cardinality of the set V of truth values.

Definition 9. *We say that a pair of elements $(b, c) \in A \times A$* splits $\mathbf{A}$ *if there exist $f_R \in \hat{\nu}_{n+p}$, $\vec{a} \in A^n$ and an index $i \in [1..n]$ such that (for all $\vec{\xi} \in A^p$):*

$$f_R(\vec{a}[b/i], \vec{\xi}) = \xi_k, \qquad f_R(\vec{a}[c/i], \vec{\xi}) = \xi_j, \quad \textit{for } k \neq j.$$

A pair is called unsplitting *if it does not split $\mathbf{A}$. We denote by $\Upsilon_{\mathbf{A}}$ the set of all unsplitting pairs of $\mathbf{A}$.*

From the point of view of the structure $\mathrm{Str}(\mathbf{A})$, a pair (b, c) is unsplitting if the elements b and c are indistinguishable, which means that for all $R \in \nu_n$, $\vec{a} \in A^n$ and index $i \in [1..n]$ we have:

$$\begin{aligned} &R^{\mathrm{Str}(\mathbf{A})}(a_1, \dots, a_{i-1}, b, a_{i+1}, \dots, a_n) = \\ &R^{\mathrm{Str}(\mathbf{A})}(a_1, \dots, a_{i-1}, c, a_{i+1}, \dots, a_n). \end{aligned} \tag{3.2}$$

Lemma 2. *Let $f_R \in \hat{\nu}_{n+p}$ and $\vec{a}, \vec{b} \in A^n$ be two sequences. If there exists $\vec{\xi} \in A^p$ such that $f_R(\vec{a}, \vec{\xi}) \neq f_R(\vec{b}, \vec{\xi})$, then there exists an index $k \in [1..n]$ such that (a_k, b_k) splits $\mathbf{A}$.*

Proof. The proof is by induction over the cardinality of the set $\{j : a_j \neq b_j\}$. Let i be the least index such that $a_i \neq b_i$. If the pair (a_i, b_i) splits $\mathbf{A}$ then we have the conclusion. Otherwise, by defining $\vec{c} = a[b_i/i]$, we have $f_R(\vec{c}, \vec{\xi}) = f_R(\vec{a}, \vec{\xi}) \neq f_R(\vec{b}, \vec{\xi})$. Now, if $\vec{c} = \vec{b}$ we have a contradiction. If $\vec{c} \neq \vec{b}$ then the conclusion follows by the induction hypothesis. □

Definition 9 extends to sets $S \subseteq A \times A$ by saying that S *splits* $\mathbf{A}$ if there exists a pair $(c, d) \in S$ splitting $\mathbf{A}$ (i.e. $S \not\subseteq \Upsilon_{\mathbf{A}}$).

Lemma 3. *Let b, c be two distinct elements of $\mathbf{A}$ and $\mathbf{B} = \mathrm{Alg}(\mathbf{A})$ be the algebraic reduct of $\mathbf{A}$. The principal congruence $\vartheta^{\mathbf{A}}(b, c) \in \mathrm{Con}(\mathbf{A})$ generated by b, c satisfies the following conditions:*

- $\vartheta^{\mathbf{B}}(b, c) \subseteq \vartheta^{\mathbf{A}}(b, c)$;
- $\vartheta^{\mathbf{A}}(b, c) = \begin{cases} \nabla^{\mathbf{A}} & \textit{if } \vartheta^{\mathbf{B}}(b, c) \textit{ splits } \mathbf{A}; \\ \vartheta^{\mathbf{B}}(b, c) & \textit{otherwise.} \end{cases}$

Proof. If $(d, e) \in \vartheta^{\mathbf{B}}(b, c)$ splits $\mathbf{A}$, then for some $f_R, \vec{a}$ and i, $f_R(\vec{a}[d/i])$ and $f_R(\vec{a}[e/i])$ project on different coordinates, say j and k. Thus $\xi_j = f_R(\vec{a}[d/i], \vec{\xi})\ \vartheta^{\mathbf{A}}(b, c)\ f_R(\vec{a}[e/i], \vec{\xi}) = \xi_k$, so $(\xi_j, \xi_k) \in \vartheta^{\mathbf{A}}(b, c)$. As ξ_j, ξ_k are arbitrary $\vartheta^{\mathbf{A}}(b, c) = \nabla^{\mathbf{A}}$. Otherwise, since the operations $f_R(\vec{a}, -, \dots, -)$ are projections, the relation $\vartheta^{\mathbf{B}}(b, c)$ is a congruence on $\mathbf{A}$. □

By Lemmas 2 and 3, any proper congruence is contained in $\Upsilon_{\mathbf{A}}$.

Definition 10. *A factor algebra $\mathbf{A}$ is* rigid *whenever $\Upsilon_{\mathbf{A}} = A \times A$.*

In other words, the factor algebra $\mathbf{A}$ is rigid exactly when the interpretation $R^{\mathrm{Str}(\mathbf{A})}$ of a relation symbol R is a constant function.

Proposition 3. *If $\mathbf{A}$ is directly decomposable then $\mathbf{A}$ is rigid.*

Proof. Let $\mathbf{A}$ be directly decomposable. Then there is a pair $(\varphi, \bar{\varphi})$ of non-trivial complementary factor congruences. By Lemma 3 and the fact that $\varphi, \bar{\varphi} \neq \nabla$, we have $\varphi \cup \bar{\varphi} \subseteq \Upsilon_{\mathbf{A}}$. Since $\Upsilon_{\mathbf{A}}$ is an equivalence relation, we have $\nabla = \varphi \circ \bar{\varphi} = \Upsilon_{\mathbf{A}}$, so $\mathbf{A}$ is rigid. □

We now characterize simple and directly indecomposable factor algebras in terms of properties of their congruences.

Proposition 4. *Let $\mathbf{A}$ be a proper factor algebra.*

1) *$\mathbf{A}$ is simple iff every proper congruence $\vartheta^{\mathrm{Alg}(\mathbf{A})}(b, c)$ splits $\mathbf{A}$.*
2) *$\mathbf{A}$ is directly indecomposable iff one of the following conditions is satisfied: (i) $\mathbf{A}$ is not rigid; (ii) $\mathbf{A}$ is rigid and the algebraic reduct $\mathrm{Alg}(\mathbf{A})$ of $\mathbf{A}$ is directly indecomposable.*

Proof. Trivial by Lemma 3 and Proposition 3. □

3.2 Factor Varieties

A variety V generated by a class of $\hat{\nu}$-factor algebras is called a *factor variety*. If V is a factor variety then V_{fa} denotes the class of $\hat{\nu}$-factor algebras belonging to V.

Proposition 5. *The variety $\mathsf{V}_{\hat{\nu}}$ generated by the class of all $\hat{\nu}$-factor algebras is axiomatized by (for $f_R \in \hat{\nu}_{n+p}$):*

(F1) $f_R(\vec{x}, \xi, \dots, \xi) = \xi$;

(F2) $f_R(\vec{x}, f_R(\vec{x}, \xi_{11}, \dots, \xi_{1p}), \dots, f_R(\vec{x}, \xi_{p1}, \dots, \xi_{pp})) =$
$f_R(\vec{x}, \xi_{11}, \dots, \xi_{pp})$;

(F3) $f_R(\vec{x}, h(\xi_{11}, \dots, \xi_{1k}), \dots, h(\xi_{p1}, \dots, \xi_{pk})) =$
$h(f_R(\vec{x}, \xi_{11}, \dots, \xi_{p1}), \dots, f_R(\vec{x}, \xi_{1k}, \dots, \xi_{pk}))$, *where $h \in \hat{\nu}_k$ is an arbitrary element of $\hat{\nu}$.*

Let $\mathbf{A} \in \mathsf{V}_{\hat{\nu}}$. For every $f_R \in \hat{\nu}_{n+p}$ and $\vec{a} \in A^n$, the p-ary map $f_R(\vec{a}, -, \dots, -)$ is a decomposition operator on $\mathbf{A}$. By $(F3)$, the decomposition operators f_R $(R \in \nu)$ are closed under composition.

By Definition 7 and by [6, Ch. 5, Thm. 2.20], the following proposition holds.

Proposition 6. *Given a factor variety V, the class V_{fa} is a universal class, so that it is closed under subalgebras and ultraproducts.*

Proposition 7. *Given a factor variety V, every directly indecomposable algebra $\mathbf{A} \in \mathsf{V}$ is a factor algebra.*

Proof. In any directly indecomposable algebra $\mathbf{A} \in \mathsf{V}$, every map $f_R(\vec{a}, -, \dots, -)$ is a trivial decomposition operator. So there must be $i \in [1..p]$ such that $\mathbf{A} \models \forall \xi_1 \dots \xi_n . f_R(\vec{a}, \xi_1, \dots, \xi_n) = \xi_i$. □

Example 2. *Let P, Q, R be propositional variables.*

In this example we will write: (i) $x \cdot y$, or just xy, for $f_P(x, y)$; (ii) $x + y$ for $f_Q(x, y)$; (iii) $\langle x, y, z\rangle$ for $f_R(x, y, z)$.

- ***Two-valued logic with a unique propositional variable.*** *The factor variety of all algebras $\mathbf{A} = (A, \cdot^{\mathbf{A}})$, where the binary operation $\cdot^{\mathbf{A}}$ is a decomposition operator on $\mathbf{A}$, is the variety of* rectangular bands *(see [14]), i.e., idempotent semigroups satisfying $xyz = xz$. The factor algebras in this variety are the left-zero bands (satisfying $xy = x$) and the right-zero bands (satisfying $xy = y$).*
- ***Two-valued logic with two propositional variables.*** *The factor variety of all algebras $\mathbf{A} = (A, \cdot^{\mathbf{A}}, +^{\mathbf{A}})$, where the binary operations $\cdot^{\mathbf{A}}$ and $+^{\mathbf{A}}$ are commuting decomposition operators on $\mathbf{A}$, is the variety of* distributive rectangular double bands. *Every algebra $\mathbf{A}$ in this variety is such that $(A, \cdot^{\mathbf{A}})$ and $(A, +^{\mathbf{A}})$ are rectangular bands, where the operations $\cdot^{\mathbf{A}}$ and $+^{\mathbf{A}}$ distribute over each other. We have four kinds of factor algebras: (1) ll-zero double bands: $xy = x = x + y$; (2) rr-zero double bands: $xy = y = x + y$; (3) lr-zero double bands: $xy = x = y + x$; (4) rl-zero double bands: $xy = y = y + x$.*
- ***Two-valued logic with two propositional variables P, Q such that $P \leftrightarrow \neg Q$.*** *The factor subvariety of the variety of distributive rectangular double bands generated by the rl-zero and lr-zero double bands constitutes the variety of* rectangular skew lattices. *Skew lattices, whose study began with the 1989 paper of Leech [15], represent the most studied class of non-commutative lattices. The importance of skew lattices lies in the structural role they play in the study of discriminator varieties.*
- ***Three-valued logic with a unique propositional variable*** *corresponds to the factor variety axiomatized by:*

  (i) $\langle x, x, x\rangle = x$;

  (ii) $\langle\langle x, y, z\rangle, \langle a, b, c\rangle, \langle m, n, p\rangle\rangle = \langle x, b, p\rangle$.

4 Algebraization of Multi-Valued Logics

In this section we consider fixed a relational type ν and a logical τ-matrix $(\mathbf{V}, \mathsf{t})$, where $V = \{v_1, \dots, v_p\}$. As announced in Section 2, we define a translation $(\cdot)^*$ from open ν-formulas into suitable terms of type $\hat{\nu}$, that we call *logical terms*.

4.1 Logical terms

First, let us fix a set $\Xi = \{\xi_1, \dots, \xi_p\}$ of fresh algebraic variables (one for each truth value), called *logical variables*. Recall that $\mathcal{T}_\nu$ stands for the set of all ν-terms (denoted by t, t_i) over the set Var. The set $\mathcal{LT}_{\hat{\nu}}$ of *logical terms* of type $\hat{\nu}$ (denoted by s, u) is generated by this grammar (for $\xi_i \in \Xi$, $f_R \in \hat{\nu}$ and $\vec{t} \in \mathcal{T}_\nu^n$):

$$s, u ::= \ \xi_i \mid f_R(\vec{t}, u_1, \dots, u_p)$$

Note that $\mathcal{LT}_{\hat{\nu}} \not\subseteq \mathcal{T}_\nu$ since $\nu \neq \hat{\nu}$ and neither the logical variables ξ_i nor the function symbols f_R can occur in t. Let $s, u_1, \dots, u_p \in \mathcal{LT}_{\hat{\nu}}$, we write $s\{u_1/\xi_1, \dots, u_p/\xi_p\}$ for the logical term obtained by substituting simultaneously u_i for each occurrence of ξ_i in s.

Lemma 4. *Given a factor algebra* $\mathbf{A}$*, an environment* $\rho : \mathrm{Var} \to A$ *and a logical term* u*, there exists a* $k \in [1..p]$ *such that* $\mathbf{A} \models_\rho \forall \xi_1 \ldots \xi_p . u = \xi_k$*.*

Proof. By induction on the size of the logical term u. □

4.2 From Open Formulas to Logical Terms through Substitutions

The translation $(\cdot)^*$ given in Section 2 for classical logic, can be easily generalized to an arbitrary p-valued matrix logic $\mathcal{L}$. Since the result of the translation is very verbose, we first introduce some clever notation based on (hyper)matrices.

Tabular notation. We consider hypermatrices of dimension $n_1 \times \cdots \times n_k$ over the set $\mathcal{LT}_{\hat{\nu}}$ of logical terms, that is functions $M : n_1 \times \cdots \times n_k \to \mathcal{LT}_{\hat{\nu}}$. Given a hypermatrix M as above, we write $M_{i_1 \ldots i_k}$ for the logical term $M(i_1, \ldots, i_k)$. A hypermatrix M of dimension p^k is called *cubical.* A *vector* v is any hypermatrix of dimension $p \times 1$ (or $1 \times p$) and its *transpose* is denoted by v^T.

Given a logical term s, we write $\mathsf{v}(s)$ for the constant vector $[s, \ldots, s]^T$, thus of dimension $p \times 1$.

Let M be a cubical hypermatrix of dimension p^k such that $M_{i_1 \ldots i_k} \in \mathcal{LT}_{\hat{\nu}}$ and let s be a logical term possibly containing $\xi_1, \ldots, \xi_p$ as variables. The matrix multiplication $M\mathsf{v}(s)$ is a hypermatrix of dimension p^{k-1} defined as follows:

$$(M\mathsf{v}(s))_{i_1 \ldots i_{k-1}} = s\{M_{i_1 \ldots i_{k-1},1}/\xi_1, \ldots, M_{i_1 \ldots i_{k-1},p}/\xi_p\}.$$

As an example, the product between a $p \times p$-matrix and $\mathsf{v}(s)$ is:

$$\begin{bmatrix} u_{11} & \cdots & u_{1p} \\ \vdots & \ddots & \vdots \\ u_{p1} & \cdots & u_{pp} \end{bmatrix} \begin{bmatrix} s \\ \vdots \\ s \end{bmatrix} = \begin{bmatrix} s\{u_{11}/\xi_1, \ldots, u_{1p}/\xi_p\} \\ \vdots \\ s\{u_{p1}/\xi_1, \ldots, u_{pp}/\xi_p\} \end{bmatrix}$$

Hereafter, we will write $M\mathsf{v}_1 \cdots \mathsf{v}_k$ for $((\cdots(M\mathsf{v}_1)\cdots)\mathsf{v}_k)$.

The translation. We translate inductively an open formula Φ of a matrix logic $\mathcal{L}$ into a logical term Φ^* as follows:

- $v_i^* = \xi_i$;
- $R(\vec{t}\,)^* = f_R(\vec{t}, \xi_1, \ldots, \xi_p)$;
- $o(\Psi_1, \ldots, \Psi_n)^* = \big(M^o \mathsf{v}(\Psi_1^*) \cdots \mathsf{v}(\Psi_{n-1}^*)\big)^T \mathsf{v}(\Psi_n^*)$, where M^o is the cubical hypermatrix of dimension p^n defined by: $M^o_{i_1 i_2 \ldots i_n} = \xi_k$ if and only if $o^{\mathbf{V}}(v_{i_n}, \ldots, v_{i_2}, v_{i_1}) = v_k$.

In particular, the translation of $P \in \mathrm{Pvar}$ is $P^* = f_P(\xi_1, \ldots, \xi_p)$.

Note that, in the definition above, M^o has dimension p^n and each $\mathsf{v}(\psi_i^*)$ has dimension $p \times 1$. So $M^o \mathsf{v}(\Psi_1^*) \cdots \mathsf{v}(\Psi_{n-1}^*)$ is a $p \times 1$-matrix and its transposed a $1 \times p$-matrix $[u_1, \ldots, u_p]$. By multiplying it by $\mathsf{v}(\Psi_n^*)$ we get a 1×1-matrix, that is a term. Moreover, we have:

$$o(\Psi_1, \ldots, \Psi_n)^* = \Psi_n^*\{u_1/\xi_1, \ldots, u_p/\xi_p\}. \tag{4.1}$$

It is easy to check by a straightforward induction on the open formula Φ that its translation Φ^* is actually a logical term. Note that, in the propositional case, such a translation induces a congruence $\sim^*$ on the set of formulas: two formulas ϕ and ψ are $\sim^*$-equivalent whenever they have the same translation $\phi^* = \psi^*$. Interestingly enough, this defines a non-commutative logic $\mathcal{L}'$ which is strictly weaker than the logic $\mathcal{L}$ we started from. The precise relationship between the logics $\mathcal{L}$ and $\mathcal{L}'$ will be investigated in further works.

Theorem 1. *Let* $\mathcal{S}$ *be a proper structure and* $\rho : \mathrm{Var} \to S$ *be an environment. Then* $[\![\Phi]\!]^{\mathcal{S}}_\rho = v_k$ *iff* $\mathrm{Fa}(\mathcal{S}) \models_\rho \forall \xi_1 \ldots \xi_p . \Phi^* = \xi_k$*.*

Proof. The proof is by induction over the complexity of the open formula Φ, using equation (4.1) and Lemma 4. □

Recall that Φ^{p} denotes the propositional translation of Φ (see Section 1). From Theorem 1 and Lemma 1, we obtain this corollary.

Corollary 1. *A universal* ν*-sentence* Φ *is a logical truth if and only if* $\mathsf{V}_{\hat{\nu}} \models \forall \xi_1 \ldots \xi_p . \Phi^* = \xi_{\mathsf{t}}$ *and* Φ^{p} *is a tautology.*

When the logic under consideration is without equality, a sentence Φ fails in a singleton structure if and only if it fails in some proper structure. Therefore, in this case it is possible to omit "and Φ^{p} is a tautology" in the statement of Corollary 1.

4.3 The algebraization of propositional logics

Propositional logic is a particular instance of quantified logic. Indeed, the set Pvar of propositional variables can be considered as a relational type, where every $P \in \mathrm{Pvar}$ is a relation symbol of arity 0. According to Definition 5, a structure $\mathcal{S}$ of type Pvar, hereafter called a *propositional structure*, is a pair $(S, P^{\mathcal{S}})_{P\in\mathrm{Pvar}}$ such that $P^{\mathcal{S}} \in V$ for every $P \in \mathrm{Pvar}$. The propositional structure $\mathcal{S}$ determines the truth assignment $\mathcal{I}_{\mathcal{S}} : \mathrm{Pvar} \to V$ defined by $\mathcal{I}_{\mathcal{S}}(P) = P^{\mathcal{S}}$. Conversely, every truth assignment $\mathcal{I} : \mathrm{Pvar} \to V$ determines, for each set S, a propositional structure $\mathcal{S}_{\mathcal{I}} = (S, P^{\mathcal{S}_{\mathcal{I}}})_{P\in\mathrm{Pvar}}$ where $P^{\mathcal{S}_{\mathcal{I}}} = \mathcal{I}(P)$. The interpretation of a propositional formula ϕ in a propositional structure $\mathcal{S}$ coincides with its propositional interpretation w.r.t. the truth assignment $\mathcal{I}_{\mathcal{S}}$: in other words, $[\![\phi]\!]^{\mathcal{S}}_{\rho} = [\![\phi]\!]^{\mathcal{I}_{\mathcal{S}}}$ for every environment $\rho : \mathrm{Var} \to S$.

We call p-factor algebra every factor algebra associated with a propositional structure according to Definition 8. Every p-factor algebra $\mathbf{A}$ is rigid and $\mathrm{Con}(\mathbf{A})$ coincides with the lattice of equivalence relations on A. So, a p-factor algebra $\mathbf{A}$ is directly indecomposable exactly when $\mathbf{A}$ is finite of prime cardinality. We denote by $\mathsf{V}_{\mathrm{prop}}$ the factor variety generated by all p-factor algebras.

Corollary 2. *Let* Pvar *be the type of propositional variables. A propositional formula ϕ is a tautology iff* $\mathsf{V}_{\mathrm{prop}} \models \forall\xi_1 \dots \xi_p.\phi^* = \xi_{\mathsf{t}}$.

We now apply our translation to propositional formulas of the logics in Example 1. To simplify the notations we confuse P with f_P, and i with ξ_i. We also perform some on-the-flight application of (F1) and directly write u rather than $s\{u/\xi_1, \dots, u/\xi_p\}$.

Example 3. (3-valued Logics with $0 < \frac{1}{2} < 1$) *The translation of some basic formulas:*

- $Ł_3\mathcal{G}_3\mathcal{P}_3$: $(P \wedge Q)^* = Q(0, P(0, \frac{1}{2}, \frac{1}{2}), P(0, \frac{1}{2}, 1))$
- $Ł_3\mathcal{G}_3\mathcal{P}_3$: $(P \vee Q)^* = Q(P(0, \frac{1}{2}, 1), P(\frac{1}{2}, \frac{1}{2}, 1), 1)$
- $Ł_3$: $(\neg P)^* = P(1, \frac{1}{2}, 0)$
- $\mathcal{G}_3$: $(\neg P)^* = P(1, 0, 0)$
- $\mathcal{P}_3$: $(\neg P)^* = P(1, 0, \frac{1}{2})$
- $Ł_3$: $(P \to Q)^* = Q(P(1, \frac{1}{2}, 0), P(1, 1, \frac{1}{2}), 1)$
- $\mathcal{G}_3$: $(P \to Q)^* = Q(P(1, 0, 0), P(1, 1, \frac{1}{2}), 1)$.

Example 4. *The translation of $P \vee \neg P$ in three-valued logics:*

- $Ł_3$: $P(1, P(\frac{1}{2}, \frac{1}{2}, 1), P(0, \frac{1}{2}, 1))$
- $\mathcal{G}_3$: $P(1, P(0, \frac{1}{2}, 1), P(0, \frac{1}{2}, 1))$
- $\mathcal{P}_3$: $P(1, P(0, \frac{1}{2}, 1), P(\frac{1}{2}, \frac{1}{2}, 1))$.

Example 5. *The Pierce law $((P \to Q) \to P) \to P$ translated in classical logic and in some three-valued logics:*

- $\mathcal{C}$: $P(P(Q(P(\mathsf{t}, \mathsf{f}), \mathsf{t}), \mathsf{f}), \mathsf{t})$
- $Ł_3$: $P(P(\alpha_1, \alpha_2, 0), P(\beta_1, \beta_2, \frac{1}{2}), 1)$ *where*
  - $\alpha_1 = Q(P(1, \frac{1}{2}, 0), P(1, 1, \frac{1}{2}), 1)$
  - $\alpha_2 = Q(P(\frac{1}{2}, 0, 0), P(\frac{1}{2}, \frac{1}{2}, 0), \frac{1}{2})$
  - $\beta_1 = Q(P(1, 1, \frac{1}{2}), 1, 1)$
  - $\beta_2 = Q(P(1, \frac{1}{2}, \frac{1}{2}), P(1, 1, \frac{1}{2}), 1)$
- $\mathcal{G}_3$: $P(P(\gamma_1, 0, 0), P(\delta_1, \delta_2, \frac{1}{2}), 1)$ *where*
  - $\gamma_1 = Q(P(1, 0, 0), 1, 1)$
  - $\delta_1 = Q(P(1, \frac{1}{2}, \frac{1}{2}), 1, 1)$
  - $\delta_2 = Q(P(1, \frac{1}{2}, \frac{1}{2}), P(1, 1, \frac{1}{2}), 1)$.

4.4 The Treatment of Equality in Classical Logic

Classical logic with equality has a binary relation symbol E as a primitive logical symbol which is always interpreted as the actual equality relation between members of the domain of discourse.

If $\mathcal{S}$ is a structure with equality on $V = \{\xi_{\mathsf{f}}, \xi_{\mathsf{t}}\}$, then the factor algebra $\mathrm{Fa}(\mathcal{S})$ has the following *switching function* f_{E} defined on S:

$$f_{\mathsf{E}}(x, y, w, z) = \begin{cases} z & \text{if } x = y; \\ w & \text{if } x \neq y. \end{cases}$$

As mentioned in the introduction, a variety of algebras generated by a class of algebras with a common switching term operation is called a *discriminator variety* [6, §9]. Discriminator varieties [27] are referred by Burris and

Sankappanavar in [6, p. 186] as "the most successful generalization of Boolean algebras to date, successful because we obtain Boolean product representations (which can be used to provide a deep insight into algebraic and logical properties)".

If $\nu_{\rm eq}$ is a relational type with equality, then the factor variety $\mathcal{V}^{\mathsf{E}}_{\nu_{\rm eq}}$ generated by all $\hat{\nu}_{\rm eq}$-factor algebras, where f_{E} is the switching function, is a discriminator variety. Notice that $\mathcal{V}^{\mathsf{E}}_{\nu_{\rm eq}}$ is a proper subvariety of the variety generated by all $\hat{\nu}_{\rm eq}$-factor algebras.

Following Vaggione [26], we have that $\mathcal{V}^{\mathsf{E}}_{\nu_{\rm eq}}$ is axiomatized by the axioms (F1)-(F3) and the identities $f_{\mathsf{E}}(x,x,\xi_{\mathsf{f}},\xi_{\mathsf{t}}) = \xi_{\mathsf{t}}$ (the reflexive property of E) and $f_{\mathsf{E}}(x,y,x,y) = x$. This last identity expresses the implication $\mathsf{E}(x,y) \to x = y$.

We now introduce a general method to express *some* properties of relations involving equality, such as the anti-symmetric property of a binary relation, without introducing an operation symbol f_{E} for equality in the algebraic type. Let Φ be a formula without equality, whose free variables include x and y, and let $\Phi \to x = y$ be an implication. The logical term Φ^*, which is the translation of the formula Φ, depends on $\xi_{\mathsf{f}}, \xi_{\mathsf{t}}$. If $\mathcal{S}$ is a proper structure and $\rho : \mathrm{Var} \to S$ is an environment, then by Theorem 1 we have $\mathcal{S} \models_\rho \Phi$ if and only if $\mathrm{Fa}(\mathcal{S}) \models_\rho \forall \xi_{\mathsf{f}}\xi_{\mathsf{t}}.\Phi^* = \xi_{\mathsf{t}}$.

Lemma 5. *Given a proper structure $\mathcal{S}$ and a formula Φ without equality, we have that $\mathcal{S} \models \Phi \to x = y$ holds if and only if $\mathrm{Fa}(\mathcal{S}) \models \forall \xi_{\mathsf{f}}.\Phi^*\{x/\xi_{\mathsf{t}}\} = \Phi^*\{y/\xi_{\mathsf{t}}\}$ holds.*

Notice that Vaggione's axiom $f_{\mathsf{E}}(x,y,x,y) = x$ (that expresses the implication $\mathsf{E}(x,y) \to x = y$) can be rewritten as follows $f_{\mathsf{E}}(x,y,\xi_{\mathsf{f}},x) = f_{\mathsf{E}}(x,y,\xi_{\mathsf{f}},y)$, while the anti-symmetric property $(xRy \wedge yRx \to x = y)$ can be translated by $f_R(y,x,\xi_{\mathsf{f}},f_R(x,y,\xi_{\mathsf{f}},x)) = f_R(y,x,\xi_{\mathsf{f}},f_R(x,y,\xi_{\mathsf{f}},y))$.

In the next example we explain how ordered algebras, introduced by Bloom in [4], can be developed as pure algebraic structures.

Example 6. (Ordered Algebras = Classical logic with a binary relation defining a compatible partial ordering) *An ordered algebra is an algebra endowed with a compatible partial order $\leq$. The* factor variety *corresponding to ordered algebras is the variety axiomatized by (F1)-(F3) and the following identities:*

(O$_1$) $f_{\leq}(x,x,\xi_{\mathsf{f}},\xi_{\mathsf{t}}) = \xi_{\mathsf{t}}$ *(Reflexivity);*

(O$_2$) $f_{\leq}(x,z,f_{\leq}(y,z,\xi_{\mathsf{t}},f_{\leq}(x,y,\xi_{\mathsf{t}},\xi_{\mathsf{f}})),\xi_{\mathsf{t}}) = \xi_{\mathsf{t}}$ *(Transitivity);*

(O$_3$) $f_{\leq}(y,x,\xi_{\mathsf{f}},f_{\leq}(x,y,\xi_{\mathsf{f}},x))) = f_{\leq}(y,x,\xi_{\mathsf{f}},f_{\leq}(x,y,\xi_{\mathsf{f}},y))$ *(Antisymmetry);*

(O$_4$) $f_{\leq}(g(\vec{z}\ [x/z_i]),g(\vec{z}\ [y/z_i]),f_{\leq}(x,y,\xi_{\mathsf{t}},\xi_{\mathsf{f}}),\xi_{\mathsf{t}}) = \xi_{\mathsf{t}}$ *for every function symbol g (Monotonicity wrt coordinate i).*

Every factor algebra $\mathbf{A}$ in this variety is a simple algebra, because every pair (a,b) (with $a \neq b$) splits $\mathbf{A}$ (see Section 3.1).

The remaining examples are devoted to show that some universal theories can be represented by well-known varieties of algebras.

Example 7. (Right-handed Skew Boolean Algebras = Classical logic with a unary relation R satisfying $R(0) \wedge \forall x(R(x) \to x = 0)$) *Let R be a unary relation and 0 be a constant. Following Cvetko-Vah and Salibra [8], the factor variety axiomatized by $f_R(0,\xi_{\mathsf{f}},\xi_{\mathsf{t}}) = \xi_{\mathsf{t}}$ and $f_R(x,\xi_{\mathsf{f}},0) = f_R(x,\xi_{\mathsf{f}},x)$, is term equivalent to the variety of right-handed skew Boolean algebras. A factor algebra $\mathbf{A}$ in this variety satisfies $f_R(0,\xi_{\mathsf{f}},\xi_{\mathsf{t}}) = \xi_{\mathsf{t}}$ and $f_R(x,\xi_{\mathsf{f}},\xi_{\mathsf{t}}) = \xi_{\mathsf{f}}$ for all $x \in A \setminus \{0\}$. Skew Boolean algebras, introduced by Cornish in [7], are non-commutative one-pointed generalizations of Boolean algebras, and occur naturally in rings, where they can be defined on certain sets of idempotents, and in particular in rings whose full set of idempotents is closed under multiplication.*

Example 8. (Boolean Algebras = Classical logic with a unary relation R satisfying $\neg R(0) \wedge R(1) \wedge \forall x(\neg R(x) \to x = 0) \wedge \forall x(R(x) \to x = 1)$) *Following Salibra* et al. *[23], the factor variety axiomatized by $f_R(0,\xi_{\mathsf{f}},\xi_{\mathsf{t}}) = \xi_{\mathsf{f}}$, $f_R(1,\xi_{\mathsf{f}},\xi_{\mathsf{t}}) = \xi_{\mathsf{t}}$, $f_R(x,\xi_{\mathsf{f}},1) = f_R(x,\xi_{\mathsf{f}},x)$ and $f_R(x,0,\xi_{\mathsf{t}}) = f_R(x,x,\xi_{\mathsf{t}})$, is term equivalent to the variety of Boolean algebras. Up to isomorphism, we have only one factor algebra which corresponds to the Boolean algebra of truth values* **2**.

5 Term Rewriting System for Factor Axioms

We now show how to turn the equations (F1)-(F3) axiomatizing the factor variety $\mathsf{V}_{\hat{\nu}}$ into rewriting rules. The term rewriting system (TRS, for short) that we obtain enjoys confluence and strong normalization. Therefore, in order to check whether $\mathsf{V}_{\hat{\nu}} \models \Phi^* = \xi_k$ holds it is enough to see whether the normal form of Φ^* is ξ_k.

For the sake of simplicity, we consider a propositional matrix logic $\mathcal{L}$ with two truth values t,f (so $\Xi = \{\xi_{\mathsf{t}},\xi_{\mathsf{f}}\}$). All definitions and results extend easily to all p-valued propositional matrix logics. We feel that this method is generalizable to arbitrary quantified matrix logics, but the actual generalization is left for future works.

We then consider a relational type ν only containing (countably many) propositional variables. Let us fix an enumeration $(P_i)_{i\in\mathbb{N}}$ of all the propositional variables in ν. Intuitively, this associates a priority $i \in \mathbb{N}$ with each propositional variable. To simplify the notation, we will still denote by P_i the binary operator $f_{P_i} \in \hat{\nu}$.

Definition 11. *The rewriting rules $\mathcal{R}$ on $\mathcal{LT}_{\hat{\nu}}$ are (for $i \in \mathbb{N}$):*

(F_1) $P_i(x, x) \rightarrowtail x$;

(F_2^ℓ) $P_i(P_i(x, y), z) \rightarrowtail P_i(x, z)$;

(F_2^r) $P_i(x, P_i(y, z)) \rightarrowtail P_i(x, z)$;

(F_3) $P_i(P_j(x, y), P_j(w, z)) \rightarrowtail P_j(P_i(x, w), P_i(y, z))$;

(F_3^ℓ) $P_i(P_j(x, y), z) \rightarrowtail P_j(P_i(x, z), P_i(y, z))$;

(F_3^r) $P_i(x, P_j(y, z)) \rightarrowtail P_j(P_i(x, y), P_i(x, z))$;

where the rules (F_3), (F_3^ℓ) and (F_3^r) only apply when $i > j$.

The TRS $\mathcal{R}$ is rather standard, except for the fact that it has infinitely many function symbols, a property that we need to handle carefully when proving termination. Note that equation ($F2$) of Proposition 5 is recovered in two steps: $P_i(P_i(x, y), P_i(w, z)) \rightarrowtail_{F_2^\ell} P_i(x, P_i(w, z)) \rightarrowtail_{F_2^r} P_i(x, z)$. Analogously, ($F3$) corresponds to ($F_3^\ell$) and ($F_3^r$), but we keep the redundant rule (F_3) to avoid an unnecessary growth of the size of the terms during the reduction.

We prove that $\mathcal{R}$ is locally confluent and terminating, so we conclude that it is confluent by Newman's lemma [2, Thm. 1.2.1].

Proposition 8. *The TRS $\mathcal{R}$ is locally confluent.*

Proof. By [2, Lemma 2.7.15], as all critical pairs are convergent. □

The fact that $\mathcal{R}$ is terminating is non-trivial because the duplication in the rules (F_3^*) may increase substantially the size of the term. Thanks to the condition "$i > j$" these rules push the symbols with small indices towards the root and those with big indices toward the leaves. Thus, two terms should be compared by first comparing their root symbols, and then recursively comparing their immediate subterms. In other words, we need a lexicographic path order (lpo).

Definition 12. *The* lexicographic path order $>_{\text{lpo}}$ *on terms is defined as follows: $s >_{\text{lpo}} u$ if and only if*

(LPO_1) $u \in \text{Var} \cup \Xi$, *$u$ occurs in s and $s \neq u$, or*

(LPO_2) $s = P_i(s_1, s_2)$, $u = P_j(u_1, u_2)$ *and one of the following conditions holds:*
- *(a)* $\exists k \in [1, 2],\ s_k \geq_{\text{lpo}} u$,
- *(b)* $i > j$, *and* $\forall k \in [1, 2], s >_{\text{lpo}} u_k$,
- *(c)* $i = j$, $(s_1, s_2) >_{\text{lpo}}^{\text{lex}} (u_1, u_2)$ *and* $\forall k \in [1, 2], s >_{\text{lpo}} u_k$,

where $>_{\text{lpo}}^{\text{lex}}$ stands for the lexicographic lpo-*order on pairs.*

By [2, Prop. 6.4.25], the relation $>_{\text{lpo}}$ is a *simplification order*, which means that it is an order closed under contexts, under substitutions, and possesses the subterm property.

Let us denote by $\text{Fun}(u)$ the set of function symbols occurring in u. The TRS $\mathcal{R}$ satisfies the following properties:

Lemma 6. *For all rewriting rules $s \rightarrowtail u \in \mathcal{R}$ we have:*
(*i*) $s >_{\text{lpo}} u$,
(*ii*) $\text{Fun}(u) \setminus \text{Fun}(s) = \emptyset$.

Proof. By a straightforward case analysis. □

Condition (*i*) amounts to saying that the TRS is *simplifying*, that is compatible with a simplification order. In the case of finite TRS, this is enough to conclude termination. As shown in [20], for infinite TRS one also need to check that the rules only introduce finitely many function symbols (in our case none, see condition (*ii*)).

Theorem 2. *The TRS $\mathcal{R}$ is confluent and terminating.*

Proof. By Lemma 6 and [20, Thm. 4.13] $\mathcal{R}$ is terminating, therefore by Proposition 8 and Newman's lemma it is confluent. □

We denote by $\text{nf}(u)$ the (unique) normal form of u w.r.t. $\mathcal{R}$.

Corollary 3. *A propositional formula ϕ is a tautology iff $\text{nf}(\phi^*) = \xi_{\text{t}}$.*

As an example, we apply the TRS to show that the law of Pierce $((P \to Q) \to P) \to P$ holds in classical logic $\mathcal{C}$, but not in Gödel's logic $\mathcal{G}_3$. We recall that both translations are given in Example 5. Without loss of generality, we assume that the priority of P is smaller than the priority of Q. As in Example 5, we will just write i for ξ_i.

In $\mathcal{C}$ we have the following reduction:

$$P(P(Q(P(\mathsf{t},\mathsf{f}),\mathsf{t}),\mathsf{f}),\mathsf{t}) \rightarrowtail_{F_2^\ell} P(Q(P(\mathsf{t},\mathsf{f}),\mathsf{t}),\mathsf{t}) \rightarrowtail_{F_3^\ell}$$
$$P(P(Q(\mathsf{t},\mathsf{t}),Q(\mathsf{f},\mathsf{t})),\mathsf{t}) \rightarrowtail_{F_2^\ell} P(Q(\mathsf{t},\mathsf{t}),\mathsf{t}) \rightarrowtail_{F_1}$$
$$P(\mathsf{t},\mathsf{t}) \rightarrowtail_{F_1} \mathsf{t}.$$

Since t is designated, the formula is a classical tautology.

To compute the reduction in $\mathcal{G}_3$, we will use the notations γ_1, δ_1, δ_2 introduced in Example 5, and the following facts:

1. $\gamma_1 \stackrel{F_3}{\to} \gamma_1'$, for $\gamma_1' = P(Q(1,1,1),Q(0,1,1),Q(0,1,1))$;
2. $\delta_2 \stackrel{F_3}{\to} \delta_2'$, for $\delta_2' = P(Q(1,1,1),Q(\frac{1}{2},1,1),Q(\frac{1}{2},\frac{1}{2},1))$.

Therefore, in $\mathcal{G}_3$ we have the following reduction:

$$\begin{array}{l} P(P(\gamma_1,0,0),P(\delta_1,\delta_2,\frac{1}{2}),1) \rightarrowtail_{F_2} P(\gamma_1,\delta_2,1) \rightarrowtail_{F_3} \\ P(\gamma_1',\delta_2,1) \rightarrowtail_{F_3} P(\gamma_1',\delta_2',1) \rightarrowtail_{F_2} P(Q(1,1,1),\delta_2',1) \rightarrowtail_{F_2} \\ P(Q(1,1,1),Q(\frac{1}{2},1,1),1) \rightarrowtail_{F_1} P(1,Q(\frac{1}{2},1,1),1) \end{array}$$

Since $P(1,Q(\frac{1}{2},1,1),1)$ is in normal form, we conclude that Pierce law is neither a tautology nor a contradiction in $\mathcal{G}_3$.

We end this section by remarking that the logical terms that appear during the reduction are not necessarily the translation of a logical formula. Henceforth, this process of calculus cannot be simulated within the logic under consideration.

6 Factor Circuits and Applications to Hardware Design

Classical propositional logic is used as a technical tool for the analysis and the synthesis of electrical circuits built up from *switches* with two stable states: the voltage levels. Analogously, p-valued logics correspond to circuits built from similar switches with p stable states, each representing a different truth value. This whole field of application of logic is called many-valued (or fuzzy) switching.

We refer the reader to [9] for a good introduction on this subject.

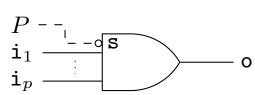
Our algebraic approach to multi-valued logics suggests a new notion of circuit, based on components that we call "decomposition gates" and behave as decomposition operators of an algebra $\mathbf{A}$ belonging to a factor variety $\mathsf{V}_{\hat{p}}$. In this section we consider $\mathbf{A}$ fixed.

We start by presenting the p-valued propositional case, then we instantiate it to propositional classical logic and compare it with the usual boolean circuits, finally we discuss the most general case.

A propositional *decomposition gate* (*D-gate*, for short) has:

- p input ports $\mathtt{i}_1, \dots, \mathtt{i}_p$ (one for each truth value);
- a switch $\mathtt{s}$, called the *selector switch*;
- an output port $\mathtt{o}$.

The graphical representation of a D-gate is the following:



The selector switch has a particular status since it specifies which decomposition operator f_P is implemented by the gate. For instance, when $\mathbf{A}$ is a factor algebra then f_P is a projection π_k^p (that is a trivial decomposition operator) and the selector switch transforms the D-gate into a multiplexer selecting its k-th input (thus $\mathtt{o} := \mathtt{i}_k$).

D-gates can be composed using *wires* by connecting the output port $\mathtt{o}$ of a D-gate with one (or more) input port(s) $\mathtt{i}_k$ of other D-gates. Therefore the wires transport the values of the algebraic variables $\xi_1, \dots, \xi_p$, in other words elements of A.

The circuit obtained by composing several D-gates is called *factor circuit.* Since each D-gate implements a decomposition operator of the algebra $\mathbf{A}$ and by (F3) decomposition operators of $\mathbf{A}$ commute, by [17, Ex. 4.38.15 p. 167] a factor circuit represents itself a decomposition operator on $\mathbf{A}$.

Every logical term u can be easily represented as a factor circuit by following its syntactic tree and drawing a D-gate with selector switch P_i for each function symbol f_{P_i}. A formula ϕ is then transformed into the factor circuit corresponding to the term ϕ^*.

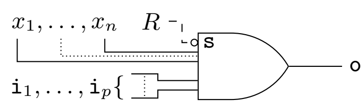
D-gates for propositional classical logic $\mathcal{C}$ are shown in Figure 1(a): to simplify the picture, we omit the selector switch and directly label the gate with the propositional variable P_i, where i represents the priority of P (as in Section 5). A quick comparison between the usual boolean circuits and factor circuits shows the novelty of this approach (cf. Figure 1(b)).

In the boolean circuits, each logical gate implements a logical connective $o \in \tau$ of arity n, so it has n input ports $\mathtt{i}_1, \dots, \mathtt{i}_n$, and its output is obtained by applying such a connective to the inputs: $o(\mathtt{i}_1, \dots, \mathtt{i}_n)$. The logical gates are connected with each others through wires that transport truth values. The remaining input wires are connected with propositional variables that can be seen as switches allowing to choose their truth values. The circuit as a whole corresponds to a boolean expression and can be simplified accordingly. Popular techniques are based, for instance, on Karnaugh maps and the result is a circuit in sum-of-products form.

On the contrary, in factor circuits there is a unique kind of gate, the D-gate, whose behavior depend on its selector switch. Every D-gate implements a decomposition operator f_{P_i}, possesses two input ports because there are two truth values, and its output is $f_{P_i}(\mathtt{i}_1, \mathtt{i}_2)$. D-gates are connected through wires transporting elements of the $\hat{\nu}$-algebra $\mathbf{A}$. The remaining input wires are connected with switches representing algebraic variables ξ_i. Globally, a factor circuit represents a decomposition operator built up from basic decomposition operators (namely, those in $\hat{\nu}$). Factor circuits can be simplified by calculating their normal form using the term rewriting system defined in Section 5 (see Figure 1(c)). Note that a factor circuit in normal form has a particular shape (see Figure 1(d)): it is a binary tree such that all the D-gates $P_{i_1}, \dots, P_{i_k}$ encountered in a root-to-leaf path have strictly increasing priority.

An interesting feature of factor circuits is that it is possible to exclude a sub-circuit by exploiting the algebraic properties of its components. Consider, for instance, the circuit in Figure 1(d) and suppose that we want to give ξ_{f} as first input of P_2 (rather than the result of $P_3(P_4(x, y), P_4(w, z))$). Then it is enough to connect the variable ξ_{f} to all input ports of the gates labelled with P_4 and the dashed subgraph trivializes thanks to axiom $(D1)$.

The D-gates for quantified matrix logics are a straightforward generalization of the propositional ones. Since an arbitrary D-gate represents a decomposition operator of shape $f_R \in \hat{\nu}_{n+p}$, it has n additional input parameters corresponding to the arguments of the relation $R(x_1, \dots, x_n)$, that is it can be drawn as follows:



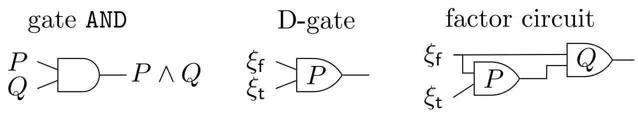
When composing arbitrary D-gates with each other, the new arguments do not play any role. In other words, it is forbidden to connect the output $\mathtt{o}$ with an input x_k. In a factor circuit the wires corresponding to $x_1, \dots, x_n$ will remain as pending input lines.

7 Symbolic Computation

Much work in computer science has been focused on reducing first-order logic to equational logic and term rewriting systems. In Tarski-Givant [25] one has a reduction of first-order Zermelo-Fraenkel set theory to traditional equational logic by using a sophisticated encoding into the equational logic of relation algebras. Burris-McKenzie's reduction of first-order logic *with* equality to equational logic through discriminator varieties uses a technique which is described in [5]. The new technique of reduction introduced in this section is based on factor varieties and can be applied to first-order logic with or without equality.

Let ν be a relational type and let $T \cup \{\Phi\}$ be a finite set of first-order ν-sentences. One of the fundamental achievements of Gödel was to show that the semantic notion $T \models \Phi$ can be captured by a syntactic notion $T \vdash \Phi$. The usual procedure to avoid the manipulation of quantifiers consists in observing that $T \models \Phi$ holds iff $T \cup \{\neg\Phi\}$ is not satisfiable iff the set of sentences in $T \cup \{\neg\Phi\}$ Skolemized is not satisfiable. This reduces the syntactic level to universally quantified sentences. Such sentences are easily expressed as conjunctions of clauses (i.e., universally quantified disjunctions of atomic and/or negated atomic formulas), so we have $T \models \Phi$ iff a suitable set of clauses is not satisfiable. Robinson's resolution rule [21] is complete for unsatisfiable sets of clauses, provided that the equality is not present in the language. In presence of equality, other rules must be introduced like paramodulation [18].

Burris and McKenzie replaces all atomic subformulas of the form $R(t_1, \dots, t_n)$ in the universally quantified sentences obtained after Skolemization, by $f_R(t_1, \dots, t_n) = t_1$, where f_R is a new function symbol corresponding to R (This approach to encoding relations as functions can be found in [1, p. 98]). The switching function of a suitable discriminator variety is used to remove the logical connectives and to derive a set of equations axiomatizing a new discriminator variety, which can be used to analyze $T \models \Phi$ when we are working with a first-order language with equality.

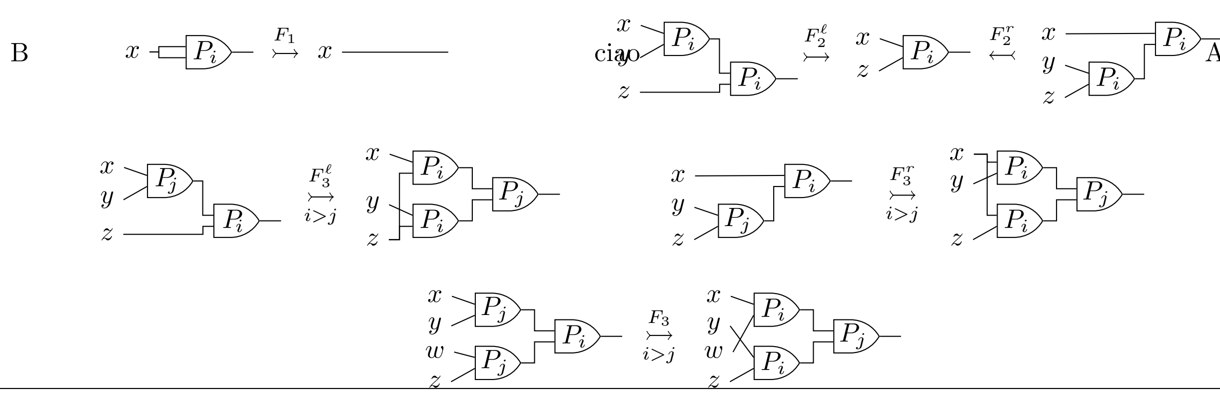


(a) A logic gate AND, a decomposition gate P and a factor circuit implementing the classical logic formula $P \wedge Q$.

	logic gate AND	D-gate
operation	connective $\wedge$	decomposition operator f_P
meaning	static (AND)	dynamic (depends on P)
no. inputs	arity of $\wedge$	$\lvert V \rvert$
input values	prop. variables P, Q	algebraic variables $\xi_{\mathsf{f}}, \xi_{\mathsf{t}}$
signals carried by the wires	truth values	elements of A
output	$P \wedge Q$	$f_P(\xi_{\mathsf{f}}, \xi_{\mathsf{t}})$

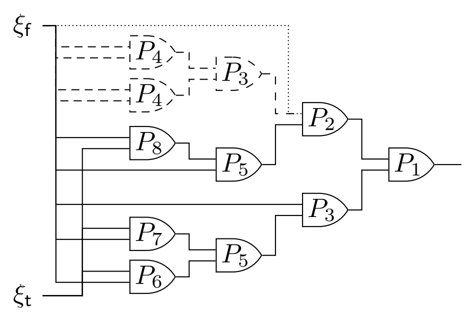
(b) Comparison between a logic gate AND and a D-gate.

Rewriting Rules



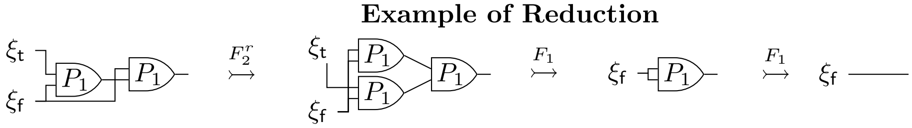
(c) Rewriting System for Factor Circuits.

A Factor Circuit in Normal Form



(d) Example of a factor circuit in normal form. The dashed subtree morally disappears because all input ports receive the same value ξ_{f}.

Example of Reduction



(e) A reduction showing that $P_1 \wedge \neg P_1$ is a contradiction in classical logic.

Figure 1: Factor circuits and decomposition gates.

7.1 Reduction to Equations through Factor Varieties

Let $T = \{\Psi_1, \dots, \Psi_n\}$ be a set of first-order sentences in classical logic and Φ be a sentence. Our goal is to reduce the semantical problem of checking whether $T \models \Phi$ holds to an equational problem in factor varieties. This will be achieved by executing the following reduction procedure, and then applying Theorem 3 below.

Reduction procedure. Consider the set $\Sigma = \{\Psi_1, \dots, \Psi_n, \neg\Phi\}$.

1. Convert all sentences in Σ into prenex normal form.
2. Compute the set $\Sigma^\sigma = \{\Psi_1^\sigma, \dots, \Psi_n^\sigma, (\neg\Phi)^\sigma\}$ by Skolemizing the sentences obtained in Step 1. As it is customary, we omit the universal quantifiers in the Skolemized sentences.
3. Add to the relational type ν the new function symbols introduced by the Skolemization to obtain the new relational type μ.
4. Consider the $\hat{\mu}$-factor variety V_Σ axiomatized by:
   (i) the axioms (F1)-(F3);
   (ii) $(\Psi_1^\sigma)^* = \xi_\mathsf{t}, \dots, (\Psi_n^\sigma)^* = \xi_\mathsf{t}$ and $((\neg\Phi)^\sigma)^* = \xi_\mathsf{t}$;
   (iii) $f_\mathsf{E}(x, x, \xi_\mathsf{f}, \xi_\mathsf{t}) = \xi_\mathsf{t}$ and $f_\mathsf{E}(x, y, x, y) = x$ *only if* the equality symbol E is present in the language.

   Let us denote by $\mathrm{Ax}(\mathsf{V}_\Sigma)$ the set of these axioms.

We denote by $\vdash_{\mathrm{eq}}$ the deducibility in the equational calculus. We have the following completeness theorem.

Theorem 3 (Completeness Theorem)**.** *Let $\Phi, \Psi_1, \dots, \Psi_n$ be first-order sentences in classical logic. Then we have $\Psi_1, \dots, \Psi_n \models \Phi$ if and only if $\mathrm{Ax}(\mathsf{V}_\Sigma) \vdash_{\mathrm{eq}} \forall xy(x = y)$ and the propositional formula $(\Psi_1 \wedge \cdots \wedge \Psi_n \to \Phi)^{\mathrm{p}}$ is a tautology.*

Proof. ($\Rightarrow$) The factor variety V_Σ is generated by the class $(\mathsf{V}_\Sigma)_{\mathrm{fa}}$ of factor algebras. From the hypothesis it follows that $(\mathsf{V}_\Sigma)_{\mathrm{fa}}$ is constituted by the trivial factor algebra. By Lemma 1 and by hypothesis we conclude that $(\Psi_1 \wedge \cdots \wedge \Psi_n \to \Phi)^{\mathrm{p}}$ is a tautology.

($\Leftarrow$) If $\Psi_1, \dots, \Psi_n \not\models \Phi$ then there exist a ν-structure $\mathcal{S}$ and a μ-structure $\mathcal{W}$ such that $\mathcal{S} \models \Sigma$, $|S| = |W|$ and $\mathcal{W} \models \Sigma^\sigma$. If $|W| > 1$, then by Theorem 1 we have $\mathrm{Fa}(\mathcal{W}) \models (\Psi^\sigma)^* = \xi_\mathsf{t}$ for every $\Psi \in \Sigma$, so that $\mathrm{Fa}(\mathcal{W}) \in \mathsf{V}_\Sigma$ and $\mathsf{V}_\Sigma \not\models \forall xy(x = y)$. If $|W| = 1$, then by Lemma 1 and the hypothesis on $\mathcal{W}$ the formula $(\Psi_1 \wedge \cdots \wedge \Psi_n \to \Phi)^{\mathrm{p}}$ is not a tautology. □

The following examples are described in [5, pp. 198-199]. The reader can compare the simplicity of our method with respect to Burris's and McKenzie's reduction procedure.

Example 9. *Let T be empty and $\Phi = \forall x(R(x) \vee \neg R(x))$. Then $\neg\Phi$ is logically equivalent to $\exists x(\neg R(x) \wedge R(x))$. After Skolemization we obtain the formula $\neg R(c) \wedge R(c)$. We consider the factor variety axiomatized by the identity $f_R(c, \xi_\mathsf{f}, f_R(c, \xi_\mathsf{t}, \xi_\mathsf{f})) = \xi_\mathsf{t}$, that implies $\xi_\mathsf{f} = \xi_\mathsf{t}$. Then by Theorem 3 it follows that $\emptyset \models \Phi$.*

Example 10. *Let T be the theory axiomatized by:*

$$\begin{array}{ll} a \neq b, & \\ \forall x(x = a \vee x = b), & \forall xyz(R(x,y) \wedge R(x,z) \to y = z), \\ \forall x \exists y R(x,y), & \forall xyz(R(x,z) \wedge R(y,z) \to x = y). \end{array}$$

Let $\Phi = \forall y \exists x R(x,y)$ and $\Sigma = T \cup \{\neg\Phi\}$. After Skolemization of Σ we get the following Σ^σ:

$$\begin{array}{lll} a \neq b, & x = a \vee x = b, & R(x,y) \wedge R(x,z) \to y = z, \\ R(x, g(x)), & \neg R(x, c), & R(x,z) \wedge R(y,z) \to x = y. \end{array}$$

The factor variety V_Σ is axiomatized by:

$$\begin{array}{c} f_\mathsf{E}(x, x, \xi_\mathsf{f}, \xi_\mathsf{t}) = \xi_\mathsf{t}, \qquad f_\mathsf{E}(x, y, x, y) = x, \\ f_\mathsf{E}(a, b, \xi_\mathsf{t}, \xi_\mathsf{f}) = \xi_\mathsf{t}, \qquad f_\mathsf{E}(x, b, f_\mathsf{E}(x, a, \xi_\mathsf{f}, \xi_\mathsf{t}), \xi_\mathsf{t}) = \xi_\mathsf{t}, \\ f_R(x, z, \xi_\mathsf{f}, f_R(x, y, \xi_\mathsf{f}, y)) = f_R(x, z, \xi_\mathsf{f}, f_R(x, y, \xi_\mathsf{f}, z)), \\ f_R(x, z, \xi_\mathsf{f}, f_R(y, z, \xi_\mathsf{f}, x)) = f_R(x, z, \xi_\mathsf{f}, f_R(y, z, \xi_\mathsf{f}, y)), \\ \\ f_R(x, g(x), \xi_\mathsf{f}, \xi_\mathsf{t}) = \xi_\mathsf{t}, \qquad f_R(x, c, \xi_\mathsf{t}, \xi_\mathsf{f}) = \xi_\mathsf{t}. \end{array}$$

Since T has no singleton models, by Theorem 3 we have that $T \models \Phi$ iff we can equationally prove $\mathrm{Ax}(\mathsf{V}_\Sigma) \vdash_{\mathrm{eq}} a = b$.

Acknowledgments. This work is partly supported by the ANR Project Coquas 12JS0200601. We thank the University of Paris 13 for inviting Salibra at the laboratory LIPN for one month.

References

[1] W. Ackermann. *Solvable Cases of the Decision Problem*. Studies in Logic and the Foundations of Mathematics. North-Holland, 1954.

[2] M. Bezem, J. W. Klop, and R. de Vrijer. *Term Rewriting Systems – TeReSe*. Cambridge University Press, 2003.

[3] W. Blok and D. Pigozzi. Algebraizable logics. *Memories of the American Mathematical Society*, 77(396), 1989.

[4] S. L. Bloom. Varieties of ordered algebras. *Journal of Computer and System Sciences*, 13(2):200 – 212, 1976.

[5] S. Burris. Discriminator varieties and symbolic computation. *Journal of Symbolic Computation*, 13(2):175 – 207, 1992.

[6] S. Burris and H. P. Sankappanavar. *A course in universal algebra*, volume 78 of *Graduate Text in Mathematics*. Springer-Verlag, 1981.

[7] W. Cornish. Boolean skew algebras. *Acta Mathematica Academiae Scientiarum Hungarica*, 36(3-4):281–291, 1980.

[8] K. Cvetko-Vah and A. Salibra. The connection of skew boolean algebras and discriminator varieties to Church algebras. *Algebra Universalis*, 73(3-4):369–390, 2015.

[9] G. Epstein. *Multi-valued Logic Design: an Introduction*. IOP Publishing, London, 1993.

[10] K. Gödel. Zum intuitionistischen aussagenkalkül. In *Anzeiger der Akademie der Wissenschaften in Wien*, volume 69, pages 65–66, 1932.

[11] P. Hájek. *Metamathematics of Fuzzy Logic*. Kluwer, Dordrect, 1998.

[12] P. R. Halmos. Polyadic boolean algebras. *Proceedings of the National Academy of Sciences USA*, 40(5):296–301, 1954.

[13] L. Henkin, J. D. Monk, and A. Tarski. *Cylindric algebras, Part I and II*. North-Holland, 1971, 1985.

[14] J. Howie. *An introduction to semigroup theory*. Academic Press, 1976.

[15] J. E. Leech. Skew lattices in rings. *Algebra Universalis*, 26(1):48–72, 1989.

[16] R. McKenzie. On the spectra, and negative solution of the decision problem for identities having finite nontrivial model. *J. Symbolic Logic*, 40:186–196, 1975.

[17] R. N. McKenzie, G. F. McNulty, and W. F. Taylor. *Algebras, Lattices, Varieties, Volume I*. Wadsworth Brooks, Monterey, California, 1987.

[18] R. Nieuwenhuis and A. Rubio. Paramodulation-based theorem proving. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, volume I, chapter 1, pages 371–443. Elsevier, 1999.

[19] N. J. Nilsson. Probabilistic logic. *Artificial Intelligence*, 28(1):71–88, 1986.

[20] E. Ohlebusch. A note on simple termination of infinite term rewriting systems. Technical report, 1992.

[21] J. A. Robinson. A machine-oriented logic based on the resolution principle. *J. of ACM*, 12(1):23–41, 1965.

[22] A. Salibra, A. Ledda, and F. Paoli. Factor varieties. *Soft Computing*, 2016. To appear.

[23] A. Salibra, A. Ledda, F. Paoli, and T. Kowalski. Boolean-like algebras. *Algebra Universalis*, 69(2):113–138, 2013.

[24] A. Tarski. Grundzüge des systemenkalküls I. *Fundamenta Mathematicae*, 25:503–526, 1935.

[25] A. Tarski and S. Givant. A formalization of set theory without variables. *AMS Colloquium Publications*, 41, 1987.

[26] D. Vaggione. Equational characterization of the quaternary discriminator. *Algebra Universalis*, 43(1):99–100, 2000.

[27] H. Werner. *Discriminator Algebras*. Studien zur Algebra und ihre Anwendungen, Band 6, Akademie-Verlag, Berlin, 1978.