Available online at www.sciencedirect.com
SCIENCE DIRECT®

Electronic Notes in Theoretical Computer Science

ELSEVIER Electronic Notes in Theoretical Computer Science 117 (2005) 249–261
www.elsevier.com/locate/entcs

A Computational Definition of the Notion of Vectorial Space

Pablo Arrighi[a] and Gilles Dowek[b]

[a] *Institut Gaspard Monge, 5 Bd Descartes, Champs-sur-Marne, 77454 Marne-la-Vallée Cedex 2, France*
arrighi@univ-mlv.fr

[b] *École polytechnique and INRIA*
LIX, École polytechnique, 91128 Palaiseau Cedex, France
Gilles.Dowek@polytechnique.fr http://www.lix.polytechnique.fr/~dowek/

Abstract

We usually define an algebra by a set, some operations defined on this set and some propositions that the algebra must validate. In some cases, we can replace these propositions by an algorithm on terms constructed upon these operations that the algebra must validate. We show in this note that this is the case for the notion of vectorial space and bilinear function.

Keywords: Rewrite system, vectorial space, bilinear function, tensorial product

One way to prove the equality of two vectors expressed by terms such as $2.\mathbf{x} + \mathbf{y} + 3.\mathbf{x}$ and $5.(\mathbf{x} + \mathbf{y}) + (-4).\mathbf{y}$ is to transform these terms into linear combinations of the unknowns and check that the terms obtained this way are the same. This algorithm transforming a term expressing a vector into a linear combination of the unknowns is also useful to express the operational semantic of programming languages for quantum computing [1], because in such languages a program and its input value form a term expressing a vector whose value, the output, is a linear combination constants. More generally, several algorithms used in linear algebra, such as matrix multiplication algorithms, transform a term expressing a vector with various constructs into a linear combination of constants.

The algorithm transforming a term expressing a vector into a linear combination of the unknowns is valid in all vectorial spaces. The goal of this note

doi:10.1016/j.entcs.2004.06.013

is to show that, moreover it completely defines the notion of vectorial space. This computational definition of the notion of vectorial space can be extended to define other algebraic notions such as bilinearity.

1 Algorithms and models

Definition 1.1 (Rewriting) Let $\mathcal{L}$ be a first-order language and R be a rewrite system on $\mathcal{L}$. We say that a term t *R-rewrites* in one step to a term u if and only if there is an occurrence α in the term t, a rewrite rule $l \longrightarrow r$ in R, and a substitution σ such that $t_{|\alpha} = \sigma l$ and $u = t[\sigma r]_\alpha$.

Definition 1.2 (Associative-Commutative Rewriting) Let $\mathcal{L}$ be a first-order language containing binary function symbols $f_1, ..., f_n$ and R be a rewrite system on $\mathcal{L}$. We say that a term t *$R/AC(f_1, ..., f_n)$-rewrites* in one step to a term u if and only if there is term t', an occurrence α in the term t', a rewrite rule $l \longrightarrow r$ in R, and a substitution σ such that $t' =_{AC} t$, $t'_{|\alpha} = \sigma l$ and $u =_{AC} t'[\sigma r]_\alpha$.

Remark 1.3 This notion must be distinguished from that of *R,AC-rewriting* where a term t rewrites to a term u only when it has a subterm AC-equivalent to an instance of the left hand side of a rewrite rule. For instance with the rule $x+x \longrightarrow 2.x$ the term $t+(u+t)$ R/AC-rewrites to $2.t+u$ but is R, AC-normal.

Definition 1.4 (Algebra) Let $\mathcal{L}$ be a first-order language. An *$\mathcal{L}$-algebra* is a family formed by a set M and for each symbol f of $\mathcal{L}$ of arity n, a function $\hat{f}$ from M^n to M. The denotation $[\![t]\!]_\phi$ of a term t for an assignment ϕ is defined as usual.

Definition 1.5 (Model of a rewrite system) Let $\mathcal{L}$ be a first-order language and R an algorithm defined by a rewrite system on terms of the language $\mathcal{L}$. An *$\mathcal{L}$-algebra* $\mathcal{M}$ is a *model* of the algorithm R, or the algorithm R is *valid* in the model $\mathcal{M}$, ($\mathcal{M} \models R$) if for all rewrite rules $l \longrightarrow r$ of the rewrite system and valuations ϕ, $[\![l]\!]_\phi = [\![r]\!]_\phi$.

Example 1.6 Consider the language $\mathcal{L}$ formed by two binary symbols $+$ and $\times$ and the algorithm R defined by the rules

$$(x + y) \times z \longrightarrow (x \times z) + (y \times z)$$
$$x \times (y + z) \longrightarrow (x \times y) + (x \times z)$$

transforming for instance, the term $(a + a) \times a$ to the term $a \times a + a \times a$. The algebra $\langle \{0, 1\}, \mathit{min}, \mathit{max} \rangle$ is a model of this algorithm.

Remark 1.7 This definition of the validity of an algorithm in a model extends some definitions of the semantics of a programming language where a semantic is defined by a set M, a function $[\,]$ mapping values of the language to elements of M and n-ary programs to functions from M^n to M, such that the program P taking the values $v_1, ..., v_n$ as input produces the value w as output if and only if $[w] = [P]([v_1], ..., [v_n])$.

Indeed, let us consider a programming language where the set of values is defined by a first-order language, whose symbols are called *constructors*. Consider an extension of this language with a function symbol p and possibly other function symbols. A program P in this language is given by a terminating and confluent rewrite system on the extended language, such that for any n-uple of values $v_1, ..., v_n$ the program P taking the values $v_1, ..., v_n$ as input produces the value w as output if and only if the normal form of the term $p(v_1, ..., v_n)$ is w. Then, a model of this rewrite system is formed by a set M, for each constructor c of arity m, a function $\hat{c}$ from M^m to M, a function $\hat{p}$ from M^n to M, and possibly other functions, such that for all rules $l \longrightarrow r$ of the rewrite system and valuations ϕ, $[\![l]\!]_\phi = [\![r]\!]_\phi$.

The denotations of the constructors define the function $[\,]$ above mapping values to elements of M and the function $\hat{p}$ is the function $[P]$. For any n-uple of values $v_1, ..., v_n$, if the normal form of the term $p(v_1, ..., v_n)$ is the value w then $[\![w]\!] = \hat{p}([\![v_1]\!], ..., [\![v_n]\!])$ and thus $[w] = [P]([v_1], ..., [v_n])$.

Definition 1.8 (Model of an AC-rewrite system) Let $\mathcal{L}$ be a first-order language containing binary function symbols $f_1, ..., f_n$, and R an algorithm defined by an $AC(f_1, ..., f_n)$-rewrite system on terms of the language $\mathcal{L}$. An *$\mathcal{L}$-algebra* $\mathcal{M}$ is a *model* of the algorithm R ($\mathcal{M} \models R$) if

- for all rewrite rules $l \longrightarrow r$ of R and valuations ϕ, $[\![l]\!]_\phi = [\![r]\!]_\phi$,
- for all valuations ϕ and indices i

$$[\![f_i(x, f_i(y, z))]\!]_\phi = [\![f_i(f_i(x, y), z)]\!]_\phi$$
$$[\![f_i(x, y)]\!]_\phi = [\![f_i(y, x)]\!]_\phi$$

2 Vectorial spaces

2.1 An algorithm

Let $\mathcal{L}$ be a 2-sorted language with a sort K for scalars and a sort E for vectors containing two binary symbols $+$ and $\times$ of rank $\langle K, K, K\rangle$, two constants 0 and 1 of sort K, a binary symbol, also written $+$, of rank $\langle E, E, E\rangle$, a binary symbol . of rank $\langle K, E, E\rangle$ and a constant $\mathbf{0}$ of sort E.

To transform a term of sort E into a linear combination of the unknows, we want to develop sums of vectors

$$\lambda.(\mathbf{u}+\mathbf{v}) \longrightarrow \lambda.\mathbf{u}+\lambda.\mathbf{v}$$

but factor sums of scalars and nested products

$$\lambda.\mathbf{u}+\mu.\mathbf{u} \longrightarrow (\lambda+\mu).\mathbf{u}$$
$$\lambda.(\mu.\mathbf{u}) \longrightarrow (\lambda\times\mu).\mathbf{u}$$

we also need the trivial rules

$$\mathbf{u}+\mathbf{0} \longrightarrow \mathbf{u}$$
$$0.\mathbf{u} \longrightarrow \mathbf{0}$$
$$1.\mathbf{u} \longrightarrow \mathbf{u}$$

and, finally, three more rules for confluence

$$\lambda.\mathbf{0} \longrightarrow \mathbf{0}$$
$$\lambda.\mathbf{u}+\mathbf{u} \longrightarrow (\lambda+1).\mathbf{u}$$
$$\mathbf{u}+\mathbf{u} \longrightarrow (1+1).\mathbf{u}$$

As we want to be able to apply the factorization rule to a term of the form $(3.\mathbf{x}+4.\mathbf{y})+2.\mathbf{x}$, reductions in the above rewrite system must be defined modulo the associativity and commutativity of $+$. This leads to the following definition.

Definition 2.1 (The rewrite system R) The rewrite system R is the AC(+)-rewrite system

$$\mathbf{u}+\mathbf{0} \longrightarrow \mathbf{u}$$
$$0.\mathbf{u} \longrightarrow \mathbf{0}$$
$$1.\mathbf{u} \longrightarrow \mathbf{u}$$
$$\lambda.\mathbf{0} \longrightarrow \mathbf{0}$$
$$\lambda.(\mu.\mathbf{u}) \longrightarrow (\lambda.\mu).\mathbf{u}$$
$$\lambda.\mathbf{u}+\mu.\mathbf{u} \longrightarrow (\lambda+\mu).\mathbf{u}$$
$$\lambda.\mathbf{u}+\mathbf{u} \longrightarrow (\lambda+1).\mathbf{u}$$
$$\mathbf{u}+\mathbf{u} \longrightarrow (1+1).\mathbf{u}$$
$$\lambda.(\mathbf{u}+\mathbf{v}) \longrightarrow \lambda.\mathbf{u}+\lambda.\mathbf{v}$$

To be complete, we should also transform the axioms of the theory of fields into a rewrite system, which is known to be difficult. However, there are many fields, for instance the field $\mathbb{Q}$ of rational numbers, whose addition and multiplication can be presented by a terminating and ground confluent rewrite system. Thus, we shall not consider an arbitrary vectorial space over an arbitrary field. But, we consider a given field $\mathcal{K}$ defined by a terminating and ground confluent rewrite system S and focus on $\mathcal{K}$-vectorial spaces.

Definition 2.2 (Scalar rewrite system) A *scalar rewrite system* is a rewrite system on a language containing at least the symbols $+$, $\times$, 0 and 1 such that:

- S is terminating and ground confluent,
- for all closed terms λ, μ and ν, the pair of terms
  · $0 + \lambda$ and λ,
  · $0 \times \lambda$ and 0,
  · $1 \times \lambda$ and λ,
  · $\lambda \times (\mu + \nu)$ and $(\lambda \times \mu) + (\lambda \times \nu)$,
  · $(\lambda + \mu) + \nu$ and $\lambda + (\mu + \nu)$,
  · $\lambda + \mu$ and $\mu + \lambda$,
  · $(\lambda \times \mu) \times \nu$ and $\lambda \times (\mu \times \nu)$,
  · $\lambda \times \mu$ and $\mu \times \lambda$
  have the same normal forms,
- 0 and 1 are normal terms.

Proposition 2.3 *The system R terminates.*

Proof. Consider the following interpretation (compatible with AC)

$$|\mathbf{u} + \mathbf{v}| = 2 + |\mathbf{u}| + |\mathbf{v}|$$
$$|\lambda.\mathbf{u}| = 1 + 2|\mathbf{u}|$$
$$|\mathbf{0}| = 0$$

Each time a term $\mathbf{t}$ rewrites to a term $\mathbf{t}'$ we have $|\mathbf{t}| > |\mathbf{t}'|$. Hence, the system terminates. □

Proposition 2.4 *For any scalar rewrite system S, the system $R \cup S$ terminates.*

Proof. By definition of the function $|\ |$, if a term $\mathbf{t}$ S-reduces to a term $\mathbf{t}'$ then $|\mathbf{t}| = |\mathbf{t}'|$. Consider a $(S \cup R)$-reduction sequence. At each R-reduction step, the measure of the term strictly decreases and at each S-reduction step it remains the same. Thus there are only a finite number of R-reduction steps in the sequence and, as S terminates, the sequence is finite. □

Definition 2.5 (The rewrite system S_0) The system S_0 is formed by the rules

$$\begin{aligned} 0 + \lambda &\longrightarrow \lambda \\ 0 \times \lambda &\longrightarrow 0 \\ 1 \times \lambda &\longrightarrow \lambda \\ \lambda \times (\mu + \nu) &\longrightarrow (\lambda \times \mu) + (\lambda \times \nu) \end{aligned}$$

where + and × are AC symbols.

Proposition 2.6 *The rewrite system $R \cup S_0$ is confluent.*

Proof. As the system is terminating it is sufficient to prove the all critical pair close. This can be mechanically checked, for instance using the system CIME [1] . □

Definition 2.7 (**Subsumption**) A terminating and confluent relation S *subsumes* a relation S_0 if whenever $t\ S_0\ u$, t and u have the same S-normal form.

Definition 2.8 (**Commutation**) The relation R *commutes* with the relation R', if whenever $\mathbf{t}\ R\ \mathbf{u}_1$ and $\mathbf{t}\ R'\ \mathbf{u}_2$, there exists a term $\mathbf{w}$ such that $\mathbf{u}_1\ R'\ \mathbf{w}$ and $\mathbf{u}_2\ R\ \mathbf{w}$.

Proposition 2.9 *Let R, S and S_0 be three relations defined on a set such that S is terminating and confluent, $R \cup S_0$ is confluent, S subsumes S_0, and the the relation R commutes with the reflexive-transitive closure S^* of S. Then the relation $R \cup S$ is confluent.*

Proof. We write $\mathbf{t}{\downarrow}$ for the S-normal form of $\mathbf{t}$. We define the relation $S^{\downarrow}$ by $\mathbf{t}\ S^{\downarrow}\ \mathbf{u}$ if $\mathbf{u}$ is the S-normal form of $\mathbf{t}$ and the relation $R; S^{\downarrow}$ by $\mathbf{t}\ (R; S^{\downarrow})\ \mathbf{u}$ if there exists a term $\mathbf{v}$ such that $\mathbf{t}\ R\ \mathbf{v}\ S^{\downarrow}\ \mathbf{u}$.

First notice that, if $\mathbf{t}\ R\ \mathbf{u}$ then $\mathbf{t}{\downarrow}\ (R; S^{\downarrow})\ \mathbf{u}{\downarrow}$. Thus if $\mathbf{t}\ (R \cup S)^*\ \mathbf{u}$ then $\mathbf{t}{\downarrow}\ (R; S^{\downarrow})^*\ \mathbf{u}{\downarrow}$ and if $\mathbf{t}\ (R \cup S_0)^*\ \mathbf{u}$ then $\mathbf{t}{\downarrow}\ (R; S^{\downarrow})^*\ \mathbf{u}{\downarrow}$.

We then check that $R; S^{\downarrow}$ is locally confluent. If $\mathbf{t}\ (R; S^{\downarrow})\ \mathbf{v}_1$ and $\mathbf{t}\ (R; S^{\downarrow})\ \mathbf{v}_2$ then there exist terms $\mathbf{u}_1$ and $\mathbf{u}_2$ such that $\mathbf{t}\ R\ \mathbf{u}_1\ S^{\downarrow}\ \mathbf{v}_1$ and $\mathbf{t}\ R\ \mathbf{u}_2\ S^{\downarrow}\ \mathbf{v}_2$. Thus, by confluence, of $R \cup S_0$ there exists a term $\mathbf{w}$ such that $\mathbf{u}_1\ (R \cup S_0)^*\ \mathbf{w}$ and $\mathbf{u}_2\ (R \cup S_0)^*\ \mathbf{w}$. Thus $\mathbf{u}_1{\downarrow}\ (R; S^{\downarrow})^*\ \mathbf{w}{\downarrow}$ and $\mathbf{u}_2{\downarrow}\ (R; S^{\downarrow})^*\ \mathbf{w}{\downarrow}$ i.e. $\mathbf{v}_1\ (R; S^{\downarrow})^*\ \mathbf{w}{\downarrow}$ and $\mathbf{v}_2\ (R; S^{\downarrow})^*\ \mathbf{w}{\downarrow}$.

As the relation $R; S^{\downarrow}$ is locally confluent and terminating, it is confluent.

Finally, if we have $\mathbf{t}\ (R \cup S)^*\ \mathbf{u}_1$ and $\mathbf{t}\ (R \cup S)^*\ \mathbf{u}_2$ then we have $\mathbf{t}{\downarrow}\ (R; S^{\downarrow})^*\ \mathbf{u}_1{\downarrow}$ and $\mathbf{t}{\downarrow}\ (R; S^{\downarrow})^*\ \mathbf{u}_2{\downarrow}$. Thus, there exists a term $\mathbf{w}$ such that $\mathbf{u}_1{\downarrow}\ (R; S^{\downarrow})^*\ \mathbf{w}$ and and $\mathbf{u}_2{\downarrow}\ (R; S^{\downarrow})^*\ \mathbf{w}$. Thus $\mathbf{u}_1\ (R \cup S)^*\ \mathbf{w}$ and $\mathbf{u}_2\ (R \cup S)^*\ \mathbf{w}$. □

[1] http://cime.lri.fr/

Proposition 2.10 *Let S be a scalar rewrite system. The rewrite system $R \cup S$ is confluent on terms containing variables of sort E but no variables of sort K.*

Proof. We use Proposition 2.9 on the set of semi-open terms, i.e. terms with variables of sort E but no variables of sort K. As S is ground confluent and terminating it is confluent and terminating on semi-open terms, S subsumes S_0 because S is a scalar rewrite system and R commutes with S^* because 0 and 1 are normal terms. □

Remark 2.11 Confluence on semi-open terms implies ground confluence in any extension of the language with constants for vectors, typically base vectors.

Proposition 2.12 *Let $\mathbf{t}$ be a normal term whose variables are among $\mathbf{x}_1, ..., \mathbf{x}_n$. The term $\mathbf{t}$ is $\mathbf{0}$ or a term of the form $\lambda_1.\mathbf{x}_{i_1} + ... + \lambda_k.\mathbf{x}_{i_k} + \mathbf{x}_{i_{k+1}} + ... + \mathbf{x}_{i_{k+l}}$ where the indices $i_1, ..., i_{k+l}$ are distinct and $\lambda_1, ..., \lambda_k$ are neither 0 nor 1.*

Proof. The term $\mathbf{t}$ is a sum $\mathbf{u}_1 + ... + \mathbf{u}_n$ of normal terms that are not sums (we take $n = 1$ if $\mathbf{t}$ is not a sum).

A normal term that is not a sum is either $\mathbf{0}$, a variable, or a term of the form $\lambda.\mathbf{v}$. In this case, λ is neither 0 nor 1 and $\mathbf{v}$ is neither $\mathbf{0}$, nor a sum of two vectors nor a product of a scalar by a vector, thus it is a variable.

As the term $\mathbf{t}$ is normal, if $n > 1$ then none of the $\mathbf{u}_i$ is $\mathbf{0}$. Hence, the term $\mathbf{t}$ is either $\mathbf{0}$ or a term of the form

$$\lambda_1.\mathbf{x}_{i_1} + ... + \lambda_k.\mathbf{x}_{i_k} + \mathbf{x}_{i_{k+1}} + ... + \mathbf{x}_{i_{k+l}}$$

where $\lambda_1, ..., \lambda_k$ are neither 0 nor 1. As the term $\mathbf{t}$ is normal, the indices $i_1, ..., i_{k+l}$ are distinct. □

2.2 Vectorial spaces

With respect to the notion of model, algorithms play the same role as sets of axioms: an algorithm may or may not be valid in a model, exactly like a set of axioms may or may not be valid in a model.

The notion of validity may be used to study sets of axioms, typically building a model is a way to prove that some proposition is not provable from a set of axioms. But validity can also be used in the other direction: to define classes of algebras as classes of models of some theories. For instance, given a field $\mathcal{K} = \langle K, +, \times, 0, 1\rangle$ the class of $\mathcal{K}$-vectorial spaces can be defined as follows.

Definition 2.13 (Vectorial space) The algebra $\langle E, +, ., \mathbf{0}\rangle$ is a $\mathcal{K}$-vectorial space if and only if the algebra $\langle K, +, \times, 0, 1, E, +, ., \mathbf{0}\rangle$ is a model of the 2-sorted set of axioms

$$\forall\mathbf{u}\forall\mathbf{v}\forall\mathbf{w}\ ((\mathbf{u}+\mathbf{v})+\mathbf{w}=\mathbf{u}+(\mathbf{v}+\mathbf{w}))$$
$$\forall\mathbf{u}\forall\mathbf{v}\ (\mathbf{u}+\mathbf{v}=\mathbf{v}+\mathbf{u})$$
$$\forall\mathbf{u}\ (\mathbf{u}+\mathbf{0}=\mathbf{u})$$
$$\forall\mathbf{u}\ \exists\mathbf{u}'\ (\mathbf{u}+\mathbf{u}'=\mathbf{0})$$
$$\forall\mathbf{u}\ (1.\mathbf{u}=\mathbf{u})$$
$$\forall\lambda\forall\mu\forall\mathbf{u}\ (\lambda.(\mu.\mathbf{u})=(\lambda.\mu).\mathbf{u})$$
$$\forall\lambda\forall\mu\forall\mathbf{u}\ ((\lambda+\mu).\mathbf{u}=\lambda.\mathbf{u}+\mu.\mathbf{u})$$
$$\forall\lambda\forall\mathbf{u}\forall\mathbf{v}\ (\lambda.(\mathbf{u}+\mathbf{v})=\lambda.\mathbf{u}+\lambda.\mathbf{v})$$

We now prove that, the class of $\mathcal{K}$-vectorial spaces can be defined as the class of models of the rewrite system R.

Proposition 2.14 *Let $\mathcal{K}=\langle K, +, \times, 0, 1\rangle$ be a field. The algebra $\langle E, +, ., \mathbf{0}\rangle$ is a $\mathcal{K}$-vectorial space if and only if the algebra $\langle K, +, \times, 0, 1, E, +, ., \mathbf{0}\rangle$ is a model of the rewrite system R.*

Proof. We first check that all the rules of R are valid in all vectorial spaces, i.e. that the propositions

$$(\mathbf{u}+\mathbf{v})+\mathbf{w}=\mathbf{u}+(\mathbf{v}+\mathbf{w})$$
$$\mathbf{u}+\mathbf{v}=\mathbf{v}+\mathbf{u}$$
$$\mathbf{u}+\mathbf{0}=\mathbf{u}$$
$$0.\mathbf{u}=\mathbf{0}$$
$$1.\mathbf{u}=\mathbf{u}$$
$$\lambda.\mathbf{0}=\mathbf{0}$$
$$\lambda.(\mu.\mathbf{u})=(\lambda.\mu).\mathbf{u}$$
$$\lambda.\mathbf{u}+\mu.\mathbf{u}=(\lambda+\mu).\mathbf{u}$$
$$\lambda.\mathbf{u}+\mathbf{u}=(\lambda+1).\mathbf{u}$$
$$\mathbf{u}+\mathbf{u}=(1+1).\mathbf{u}$$
$$\lambda.(\mathbf{u}+\mathbf{v})=\lambda.\mathbf{u}+\lambda.\mathbf{v}$$

are theorems of the theory of vectorial spaces.

Seven of them are axioms of the theory of vectorial spaces, the propositions $\lambda.\mathbf{u}+\mathbf{u}=(\lambda+1).\mathbf{u}$ and $\mathbf{u}+\mathbf{u}=(1+1).\mathbf{u}$ are consequence of $1.\mathbf{u}=\mathbf{u}$ and $\lambda.\mathbf{u}+\mu.\mathbf{u}=(\lambda+\mu).\mathbf{u}$. Let us prove that $0.\mathbf{u}=\mathbf{0}$. Let $\mathbf{u}'$ be such that $\mathbf{u}+\mathbf{u}'=$

$\mathbf{0}$. Then $0.\mathbf{u} = 0.\mathbf{u}+\mathbf{0} = 0.\mathbf{u}+\mathbf{u}+\mathbf{u}' = 0.\mathbf{u}+1.\mathbf{u}+\mathbf{u}' = 1.\mathbf{u}+\mathbf{u}' = \mathbf{u}+\mathbf{u}' = \mathbf{0}$. Finally $\lambda.\mathbf{0} = \mathbf{0}$ is a consequence of $0.\mathbf{u} = \mathbf{0}$ and $\lambda.(\mu.\mathbf{u}) = (\lambda.\mu).\mathbf{u}$.

Conversely, we prove that all axioms of vectorial spaces are valid in all models of R. The validity of each of them is a consequence of the validity of a rewrite rule, except $\forall\mathbf{u}\exists\mathbf{u}'\ (\mathbf{u}+\mathbf{u}' = \mathbf{0})$ that is a consequence of $\mathbf{u}+(-1).\mathbf{u} = \mathbf{0}$ itself being a consequence of $\lambda.\mathbf{u} + \mu.\mathbf{u} = (\lambda + \mu).\mathbf{u}$ and $0.\mathbf{u} = \mathbf{0}$. □

2.3 Universality

Proposition 2.15 *Let* $\mathbf{t}$ *and* $\mathbf{u}$ *be two terms whose variables are among* $\mathbf{x}_1, ..., \mathbf{x}_n$*. The following propositions are equivalent:*

(i) *the normal forms of* $\mathbf{t}$ *and* $\mathbf{u}$ *are identical modulo AC,*

(ii) *the equation* $\mathbf{t} = \mathbf{u}$ *is valid in all* $\mathcal{K}$*-vectorial spaces,*

(iii) *and the denotation of* $\mathbf{t}$ *and* $\mathbf{u}$ *in* K^n *for the assignment* $\phi = \mathbf{e}_1/\mathbf{x}_1, ..., \mathbf{e}_n/\mathbf{x_n}$*, where* $\mathbf{e}_1, ..., \mathbf{e}_n$ *is the canonical base of* K^n*, are identical.*

Proof. Proposition (i) implies proposition (ii) and proposition (ii) implies proposition (iii). Let us prove that proposition (iii) implies proposition (i).

Let $\mathbf{t}$ be a normal term whose variables are among $\mathbf{x}_1, ..., \mathbf{x}_n$. The *decomposition* of $\mathbf{t}$ along $\mathbf{x}_1, ..., \mathbf{x}_n$ is the sequence $\alpha_1, ..., \alpha_n$ such that if there is a subterm of the form $\lambda.\mathbf{x}_i$ in $\mathbf{t}$, then $\alpha_i = \lambda$, if there is a subterm of the form $\mathbf{x}_i$ in $\mathbf{t}$, then $\alpha_i = 1$, and $\alpha_i = 0$ otherwise.

Assume $[\![\mathbf{t}]\!]_\phi = [\![\mathbf{u}]\!]_\phi$. Let $\mathbf{e}_1, ..., \mathbf{e}_n$ be the canonical base of K^n and $\phi = \mathbf{e}_1/\mathbf{x}_1, ..., \mathbf{e}_n/\mathbf{x}_n$. Call $\alpha_1, ..., \alpha_n$ the coordinates of $[\![\mathbf{t}]\!]_\phi$ in $\mathbf{e}_1, ..., \mathbf{e}_n$. Then the decompositions of the normal forms of $\mathbf{t}$ and $\mathbf{u}$ are both $\alpha_1, ..., \alpha_n$ and thus they are identical modulo AC. □

3 Bilinearity

3.1 An algorithm

Definition 3.1 (The rewrite system R') Consider a language with four sorts: K for scalars and E, F, and G for the vectors of three vector spaces, the symbols $+$, $\times$, 0, 1 for scalars, three copies of the symbols $+$, . and $\mathbf{0}$ for each sort E, F, and G and a symbol $\otimes$ of rank $\langle E, F, G\rangle$.

The system R' is the rewrite system formed by three copies of the rules of the system R and the rules

$$(\mathbf{u} + \mathbf{v}) \otimes \mathbf{w} \longrightarrow (\mathbf{u} \otimes \mathbf{w}) + (\mathbf{v} \otimes \mathbf{w})$$

$$(\lambda.\mathbf{u}) \otimes \mathbf{v} \longrightarrow \lambda.(\mathbf{u} \otimes \mathbf{v})$$

$$\mathbf{u} \otimes (\mathbf{v} + \mathbf{w}) \longrightarrow (\mathbf{u} \otimes \mathbf{v}) + (\mathbf{u} \otimes \mathbf{w})$$
$$\mathbf{u} \otimes (\lambda.\mathbf{v}) \longrightarrow \lambda.(\mathbf{u} \otimes \mathbf{v})$$
$$\mathbf{0} \otimes \mathbf{u} \longrightarrow \mathbf{0}$$
$$\mathbf{u} \otimes \mathbf{0} \longrightarrow \mathbf{0}$$

Proposition 3.2 *The rewrite system R' terminates.*

Proof. We extend the interpretation of Definition 2.3 with

$$|\mathbf{u} \otimes \mathbf{v}| = (3|\mathbf{u}| + 2)(3|\mathbf{v}| + 2)$$

□

Proposition 3.3 *For any scalar rewrite system S, the system $R' \cup S$ terminates.*

Proof. As in Proposition 2.4. □

Proposition 3.4 *The rewrite system $R' \cup S_0$ is confluent.*

Proof. As in the proof of Proposition 2.6, we prove local confluence by checking that all critical pair close. □

Proposition 3.5 *Let S be a scalar rewrite system. The rewrite system $R' \cup S$ is confluent on terms containing variables of sort E, F, and G but no variables of sort K.*

Proof. Using Proposition 2.9. □

Proposition 3.6 *Let $\mathbf{t}$ be a normal term whose variables of sort E are among $\mathbf{x}_1, ..., \mathbf{x}_n$, whose variables of sort F are among $\mathbf{y}_1, ..., \mathbf{y}_p$, and that has no variables of sort G and K. If $\mathbf{t}$ has sort E or F, then it has the same form as in Proposition 2.12. If it has sort G, then it has the form*

$$\lambda_1.(\mathbf{x}_{i_1} \otimes \mathbf{y}_{j_1}) + ... + \lambda_k.(\mathbf{x}_{i_k} \otimes \mathbf{y}_{j_k}) + (\mathbf{x}_{i_{k+1}} \otimes \mathbf{y}_{j_{k+1}}) + ... + (\mathbf{x}_{i_{k+l}} \otimes \mathbf{y}_{j_{k+l}})$$

where the pairs of indices $\langle i_1, j_1 \rangle, ..., \langle i_{k+l}, j_{k+l} \rangle$ are distinct and $\lambda_1, ..., \lambda_k$ are neither 0 nor 1.

Proof. The term $\mathbf{t}$ is a sum $\mathbf{u}_1 + ... + \mathbf{u}_n$ of normal terms that are not sums (we take $n = 1$ if $\mathbf{t}$ is not a sum).

A normal term that is not a sum is either $\mathbf{0}$, a term of the form $\mathbf{v} \otimes \mathbf{w}$, or of the form $\lambda.\mathbf{v}$. In this case, λ is neither 0 nor 1 and $\mathbf{v}$ is neither $\mathbf{0}$, nor a sum of two vectors nor a product of a scalar by a vector, thus it is of the form $\mathbf{v} \otimes \mathbf{w}$.

In a term of the form $\mathbf{v} \otimes \mathbf{w}$, neither $\mathbf{v}$ nor $\mathbf{w}$ is a sum, a product of a scalar by a vector or $\mathbf{0}$. Thus both $\mathbf{v}$ and $\mathbf{w}$ are variables.

As the term $\mathbf{t}$ is normal, if $n > 1$ then none of the $\mathbf{u}_i$ is $\mathbf{0}$. Hence, the term $\mathbf{t}$ is either $\mathbf{0}$ or a term of the form $\lambda_1.(\mathbf{x}_{i_1} \otimes \mathbf{y}_{j_1}) + ... + \lambda_k.(\mathbf{x}_{i_k} \otimes \mathbf{y}_{j_k}) + (\mathbf{x}_{i_{k+1}} \otimes \mathbf{y}_{j_{k+1}}) + ... + (\mathbf{x}_{i_{k+l}} \otimes \mathbf{y}_{j_{k+l}})$ where $\lambda_1, ..., \lambda_k$ are neither 0 nor 1. As the term $\mathbf{t}$ is normal, the pairs of indices are distinct. □

3.2 Bilinearity

Definition 3.7 (Bilinear function) Let E, F, and G be three vectorial spaces on the same field. A function $\otimes$ from $E \times F$ to G is said to be *bilinear* if

$$(\mathbf{u} + \mathbf{v}) \otimes \mathbf{w} = (\mathbf{u} \otimes \mathbf{w}) + (\mathbf{v} \otimes \mathbf{w})$$

$$(\lambda.\mathbf{u}) \otimes \mathbf{v} = \lambda.(\mathbf{u} \otimes \mathbf{v})$$

$$\mathbf{u} \otimes (\mathbf{v} + \mathbf{w}) = (\mathbf{u} \otimes \mathbf{v}) + (\mathbf{u} \otimes \mathbf{w})$$

$$\mathbf{u} \otimes (\lambda.\mathbf{v}) = \lambda.(\mathbf{u} \otimes \mathbf{v})$$

Proposition 3.8 *Let $\mathcal{K} = \langle K, +, \times, 0, 1\rangle$ be a field. The structures $\langle E, +, ., \mathbf{0}\rangle$, $\langle F, +, ., \mathbf{0}\rangle$, $\langle G, +, ., \mathbf{0}\rangle$ are $\mathcal{K}$-vectorial spaces and $\otimes$ is a bilinear function from $E \times F$ to G if and only if $\langle K, +, \times, 0, 1, E, +, ., \mathbf{0}, F, +, ., \mathbf{0}, G, +, ., \mathbf{0}, \otimes\rangle$ is a model of the system R'.*

Proof. The validity of the rules of the three copies of the system R, express that $\langle E, +, ., \mathbf{0}\rangle$, $\langle F, +, ., \mathbf{0}\rangle$, $\langle G, +, ., \mathbf{0}\rangle$ are $\mathcal{K}$-vectorial spaces. The validity of the six other rules is the validity of the axioms of Definition 3.7 plus the two extra propositions $\mathbf{0} \otimes \mathbf{u} = \mathbf{0}$ and $\mathbf{u} \otimes \mathbf{0} = \mathbf{0}$ that are consequences of these axioms. □

3.3 Universality

Definition 3.9 (Tensorial product) Let E and F be two vectorial spaces, the pair formed by the vectorial space G and the bilinear function from $E \times F$ to G is a *tensorial product* of E and F if for all bases $(\mathbf{e}_i)_{i \in I}$ of E and $(\mathbf{e'}_j)_{j \in J}$ of F the family $(\mathbf{e}_i \otimes \mathbf{e'}_j)_{\langle i,j\rangle}$ is a base of G.

Example 3.10 Let $\otimes$ be the unique bilinear function such that $\mathbf{e}_i \otimes \mathbf{e'}_j = \mathbf{e''}_{p(i-1)+j}$ where $\mathbf{e}_1, ..., \mathbf{e}_n$ is the canonical base of K^n, $\mathbf{e'}_1, ..., \mathbf{e'}_p$ that of of K^p, and $\mathbf{e''}_1, ..., \mathbf{e''}_{np}$ that of K^{np}. Then K^{np} together with $\otimes$ is the tensorial product of K^n and K^p.

Proposition 3.11 *Let* $\mathbf{t}$ *and* $\mathbf{u}$ *be two terms whose variables of sort* E *are among* $\mathbf{x}_1, ..., \mathbf{x}_n$*, whose variables of sort* F *are among* $\mathbf{y}_1, ..., \mathbf{y}_p$*, and that have no variables of sort* G *and* K*. The following propositions are equivalent:*

(i) *the normal forms of* $\mathbf{t}$ *and* $\mathbf{u}$ *are identical modulo* AC*,*

(ii) *the equation* $\mathbf{t} = \mathbf{u}$ *is valid in all structures formed by three vectorial spaces and a bilinear function,*

(iii) *the equation* $\mathbf{t} = \mathbf{u}$ *is valid in all structures formed by two vectorial spaces and their tensorial product,*

(iv) *and the denotation of* $\mathbf{t}$ *and* $\mathbf{u}$ *in* K^{np} *for the assignment*

$$\phi = \mathbf{e}_1/\mathbf{x}_1, ..., \mathbf{e}_n/\mathbf{x_n}, \mathbf{e'}_1/\mathbf{y}_1, ..., \mathbf{e'}_p/\mathbf{y_p}$$

where $\mathbf{e}_1, ..., \mathbf{e}_n$ *is the canonical base of* K^n*,* $\mathbf{e'}_1, ..., \mathbf{e'}_p$ *that of* K^p *and* $\otimes$ *is the unique bilinear function such that* $\mathbf{e}_i \otimes \mathbf{e'}_j = \mathbf{e''}_{p(i-1)+j}$ *where* $\mathbf{e''}_1, ..., \mathbf{e''}_{np}$ *is the canonical base of* K^{np}*.*

Proof. Proposition (i) implies proposition (ii), proposition (ii) implies proposition (iii) and proposition (iii) implies proposition (iv). Let us prove that proposition (iv) implies proposition (i).

Let $\mathbf{t}$ be a normal term of sort G with variables of sort E among $\mathbf{x}_1, ..., \mathbf{x}_n$, variables of sort F among $\mathbf{y}_1, ..., \mathbf{y}_p$, and no variables of sort G and K. The *decomposition* of $\mathbf{t}$ along $\mathbf{x}_1, ..., \mathbf{x}_n$, $\mathbf{y}_1, ..., \mathbf{y}_p$, is the sequence $\alpha_1, ..., \alpha_{np}$ such that if there is a subterm of the form $\lambda.(\mathbf{x}_i \otimes \mathbf{y}_j)$ in $\mathbf{t}$, then $\alpha_{p(i-1)+j} = \lambda$, if there is a subterm of the form $\mathbf{x}_i \otimes \mathbf{y}_j$ in $\mathbf{t}$, then $\alpha_{p(i-1)+j} = 1$, and $\alpha_{p(i-1)+j} = 0$ otherwise.

Assume $[\![\mathbf{t}]\!]_\phi = [\![\mathbf{u}]\!]_\phi$. Call $\alpha_1, ..., \alpha_{np}$ the coordinates of $[\![\mathbf{t}]\!]_\phi$ in $\mathbf{e''}_1, ..., \mathbf{e''}_{np}$. Then the decompositions of the normal forms of $\mathbf{t}$ and $\mathbf{u}$ are both $\alpha_1, ..., \alpha_{np}$ and thus they are identical modulo AC. □

Conclusion

We usually define an algebra by three components: a set, some operations defined on this set and some propositions that must be valid in the algebra. For instance a $\mathcal{K}$-vectorial space is defined by a set E, the operations $\mathbf{0}$, $+$ and . and the equations of Definition 2.13.

We can, in a more computation-oriented way, define an algebra by a set, operations on this set and an algorithm on terms constructed upon these operations that must be valid in the algebra. For instance a $\mathcal{K}$-vectorial space is defined by a set E, the operations $\mathbf{0}$, $+$ and . and the algorithm R.

This algorithm is a well-known algorithm in linear algebra: it is the algorithm that transforms any linear expression into a linear combination of the unknowns. This algorithm is, at a first look, only one among the many algorithms used in linear algebra, but it completely defines the notion of vectorial space: a vectorial space is any algebra where this algorithm is valid, it is any algebra where linear expressions can be transformed this way into linear combinations of the unknowns.

Acknowledgements

The authors want to thank Evelyne Contejean, Claude Kirchner and Claude Marché for comments on a previous draft of this paper.

References

[1] P. Arrighi and G. Dowek, Operational semantics for formal tensorial calculus, *2nd International Workshop on Quantum Programming Languages*, Helsinki, 2004.

[2] N. Dershowitz and J.-P. Jouannaud, *Rewrite systems*, Handbook of theoretical computer science (vol. B): formal models and semantics, MIT Press, 1991.

[3] G.E. Peterson and M.E. Stickel, *Complete sets of reductions for some equational theories*, Journal of the ACM, 28, 2, p.233-264, 1981.