Acta Cybernetica **23** (2017) 191–201.

Initial Algebra for a System of Right-Linear Functors*

Anna Labella[a] and Rocco De Nicola[b]

Abstract

In 2003 we showed that right-linear systems of equations over regular expressions, when interpreted in a category of trees, have a solution whenever they enjoy a specific property that we called *hierarchicity* and that is instrumental to avoid critical mutual recursive definitions. In this note, we prove that a right-linear system of polynomial endofunctors on a cocartesian monoidal closed category which enjoys parameterized *left list arithmeticity*, has an initial algebra, provided it satisfies a property similar to hierarchicity.

Keywords: regular expressions, monoidal categories, system of functors

1 Introduction

Our paper [4] acknowledges that "*the ideals that led to the work stemmed from discussions with Zoltán Ésik*"; as a homage to Zoltán here we generalise the results of [4] to a much larger setting. There we defined the class of the linear systems whose solution is expressible as a tuple of nondeterministic regular expressions [3] when they are interpreted as trees of actions rather than as sets of action sequences. We exactly characterized those systems that have a regular expression as a "canonical" solution, and showed that any regular expression can be obtained as a canonical solution of a system of the defined class.

The key ingredient for obtaining the wanted solution was our restriction to "hierarchical" equations that were instrumental to avoid critical mutual recursive definitions. Indeed, if we model variables as nodes of graphs and their dependences as directed arcs, we required that whenever a variable y depends on x, (x is at the beginning of a loop that contains y) we have that y never occurs in other loops originated by other variables different from x.

Thus, in [4] we proved that a right-linear system of equations, interpreted in a category of trees has a solution whenever it is hierarchical. In this short note

*This work is dedicated to Zoltán Ésik whose unexpected and untimely death left us shattered and without words.

[a]Dipartimento di Informatica, Sapienza Università di Roma, E-mail: labella@di.uniroma1.it
[b]IMT School for Advanced Studies Lucca, E-mail: rocco.denicola@imtlucca.it

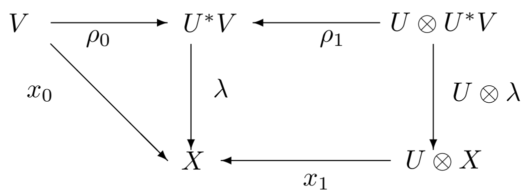
DOI: 10.14232/actacyb.23.1.2017.12

we prove that a right-linear system of polynomial endofunctors on a cocartesian monoidal closed category which enjoys parameterized *left list arithmeticity*, has an initial algebra, provided it satisfies a property similar again to a hierarchicity condition. We could thus say that the "solution" for the system provided here is canonical in a strict sense.

2 Initial algebras and *llist*-arithmeticity

In order to introduce an initial algebra for a linear polynomial endofunctor expressed in terms of (canonical) sum $+$ and a possibly non commutative tensor product $\otimes$, we have to consider a notion of recursive object which generalizes Cockett definition [2] of $rec(U, V)$, where the canonical product $\times$ played the role of multiplication. As a matter of fact, we still ask for an initial algebra for an endofunctor $U \otimes (-) + V : C \to C$ in a monoidal category $(C, \otimes, I)$, but we have to be aware of a non commutative situation. We chose to have the *left composition*, because our result is particularly meaningful for categories which are monoidal (right) closed whose objects have an elegant representation (see Proposition 1).

Definition 1. *Given a cocartesian monoidal category $(C, \otimes, I)$, we call U^*V the initial algebra of the functor $U \otimes (-) + V$, if it does exist. In that case there is a morphism $U \otimes (U^*V) + V \to U^*V$ canonical w.r.t. any other $U \otimes (-) + V$-algebra. This means that, U^*V is equipped with two morphisms ρ_0, ρ_1 such that, given another object X with two similar morphisms x_0, x_1, there is a unique morphism λ making the following diagram commute.*



In case C is a partial order, U^*V is the minimal solution of the corresponding inequation $U \otimes X + V \leq X$. But, in any case, being U^*V an initial algebra, we have that $U \otimes U^*V + V \simeq U^*V$ ($U \otimes U^*V + V = U^*V$, in the case of partial order), i.e. it is an initial fixed point.

When $\otimes$ is the canonical product, we do get the well known definition of $rec(U, V)$ provided by Cockett [2], i.e., the V-parameterized $list(U)$, that becomes $list(U)$ when $V \simeq \mathbf{1}$ is the terminal object. Since the constant value of the tensor product we consider is on the left and the tensor product is non-commutative, we will talk about left lists, that we will refer as *llist* and as *parameterized llist*.

For a generic tensor product, $\otimes$, we have that the initial algebra of $U \otimes (-) + V$ is U^*V that we call parameterized *llist*(U); in case $V \simeq I$ the initial algebra is U^*I that we call *llist*(U).

One can easily prove (see Adámek theorem in [1]) that in a monoidal cocartesian category, which has colimits for every countable chain, there is an initial algebra for all the functors above. Such initial algebras can be obtained as initial fixed points, i.e., as colimits of the chain built starting from the initial object 0 and then repeatedly applying the functor.

Proposition 1. *In a monoidal cocartesian, chain cocomplete category C, semidistributive on the right, in the sense of [5], we have that:*

1. *There is a canonical morphism $U^*V \to U^*I \otimes V$*
2. *If tensor product distributes on the right w.r.t. chain colimits, e.g. it has a right adjoint, then we have $U^*V \simeq U^*I \otimes V$*

Proof.

1. It suffices to prove that $U^*I \otimes V$ is a fixed point of the same functor as U^*V. From this, the existence of the required canonical morphism would follow because U^*V is the initial fixed point. To prove that $U^*I \otimes V$ is a fixed point, let us apply the functor $U \otimes (-) + V$ to $U^*I \otimes V$. By using the associativity law and the right distributivity law, we get the following series of isomorphisms:

   $$U \otimes (U^*I \otimes V) + V \simeq (U \otimes U^*I) \otimes V + I \otimes V \simeq (I + U \otimes U^*I) \otimes V \simeq U^*I \otimes V.$$

2. If the tensor product preserves chain colimits, it preserves also fixed points. In particular it is true in case C is monoidal (right)-closed.

□

If we write U^* instead of U^*I, Proposition 1 allows us to interchangeably use $U^* \otimes V$ and U^*V when working with monoidal closed categories.

By relying on Proposition 1 we have that if C has $llist(U)$, it has also parameterized $llist(U)$. In analogy with the case of categories with cartesian product where a category having (parameterized) *list*s is called *list*-arithmetic [1], we will call our category $left - list$-arithmetic or *llist*-arithmetic when it has (parameterized) *llist*s.

Proposition 2. *Given a cocartesian monoidal right closed category C which has initial algebra for the functor $U \otimes (-) + I$, it has initial algebra for all the functors $U \otimes (-) + V$.*

Proof. The proof follows from Proposition 1. If U^* is an initial algebra for functor $U \otimes (-) + I$, $U^* \otimes V$ is an initial algebra for functor $U \otimes (-) + V$. □

We can now consider three instances of *llist*-arithmetic categories that build on A^*, the free monoid generated by an alphabet A:

[1] This name is related to the fact that, when a *list*-arithmetic category is also a pretopos, it is possible to develop arithmetic in it and we speak of an arithmetic universe in the sense of Joyal [7].

$P(A^*)$ the algebra of sets of words on A, a monoidal category w.r.t. concatenation whose morphisms are inclusions. Here parameterized $llist$(U,V), i.e. U^*V, is the binary Kleene star U^*V and as a consequence of Proposition 1 we have that it is reducible to the unary star because $P(A^*)$ is a monoidal right closed category (the derivation operation is right adjoint to concatenation). In this case, the tensor product distributes over sums on both sides.

$Set|A^*$ the topos of A^*-labelled sets, where the (non-commutative) tensor product is obtained from the concatenation in A^*. By taking a (commutative) monoid M, we could obtain from $Set|M$ a (commutative) monoidal structure.

$Tree(A)$ is generalization of $P(A^*)$. Structured sets of computations are organised as a category of generalised trees built over a (complete) meet-semilattice monoid generated from A. The tensor product $\otimes$ is provided by the concatenation of trees allowed by the concatenation of A^*. This concatenation is non commutative and only right-distributive w.r.t. sums [5], but also right closed. The category $Tree(A)$ has initial algebra for functors $s \otimes (-) + t$, i.e it is *llist*-arithmetic with the *llist* s^*t given by iteration of a tree s, followed every time by a copy of t [4][2].

3 Right-linear hierarchical systems of functors

It is a result of classical theory of regular languages [8] that we can consider a grammar on an alphabet A as a continuous operator from $P(A^*)^n$ to $P(A^*)^n$ consisting of a system of n linear equations in n variables. This system can be "solved" by repeatedly applying the rule

$$X = U^*V \text{ implies } X = U \otimes X + V \qquad (* - rule)$$

In this way, we obtain a minimal fixed point for the operator associated with the grammar. In the present categorical context, we could say this rule is a direct consequence the *llist*-arithmeticity of the considered structure.

In [4], we extended this result to the category $Tree(A)$, but, due to the fact that only a right side distributivity of tensor product w.r.t. sum holds, we had to restrict the class of solvable systems by considering only so-called right-linear hierarchical systems (*rlhs*) that allowed us to avoid critical mutual recursive definitions. Formulated according to the current terminology the result of [4] is described by the following proposition.

Proposition 3. *In the category $Tree(A)$, the $* - rule$ provides a solution for hierarchical (see below) finite right-linear systems of polynomial equations.* □

Now we do generalize this result again and show that *llist*-arithmeticity in a cocartesian right-distributive monoidal category C all finite right-linear hierarchical systems of functors have an initial algebras.

[2] Actually, $Tree(A)$ is a coherent *llist*-arithmetic category, but not a pre-topos because not all its monos are regular.

When such a category C contains as its objects the elements of an alphabet A, some of the objects of C can be rendered as regular expressions generated by means of the following BNF starting from the elments a of an alphabet A.

$$E ::= \; 0 \mid I \mid a \mid E+E \mid E \otimes E \mid E^* \qquad \text{where } a \text{ is in } A.$$

In such a grammar, 0 denotes the initial object of our category, I denotes the unit for $\otimes$, that is the tensor product of C. Moreover $+$ stands for the coproduct of C and * denotes the *llist*-constructor.

Our result will be formulated by relying on such terminology. Indeed, if we suppose that C is cocartesian monoidal closed and elements of A are its objects, then the interpretation of *llist*s will allow the construction of parameterized *llist*s, as described in Proposition 2.

Our aim is to prove that, by relying on the following rule

$$U \otimes X + V \to X \; \textit{implies} \; U^*V \to X \qquad (\textit{initiality} - \textit{rule})$$

that guarantees that if there is a morphism $U \otimes X + V \to X$ then there is a canonical morphism $U^*V \to X$, it is possible to find an initial algebra for every right-linear hierarchical system of functors on regular expressions.

Summing up, we will extend the result proved in [4] for $Tree(A)$ to a category C with the properties mentioned above. To this aim we have to formulate it in terms of functors instead of equations of linear functions in order to prove that the obtained solution is canonical because it is the initial algebra of the system of functors.

We need to provide some definitions.

Definition 2.

- *Given a category C interpreting regular expressions, a functor $F : C^n \to C$ of the form $\sum_{1 \leq i \leq n} U_i \otimes X_i + V$ is called* right-linear *polynomial functor in n variables.*
- *A right-linear polynomial functor is called* simple *when all U_i and V do not contain the $()^*$ operator.*
- *A right-linear polynomial system of functors of dimension n is a n-tuple*

$$\Phi =< F_1, \dots, F_n >: C^n \to C^n$$

of right-linear polynomial functors in n variables.

Given a functor $F : C \times D \to C$, for every object d of D, we can consider the endofunctor $F((-), d) : C \to C$.

Moreover, if F is a functor with n-argument $\sum_{1 \leq i \leq n} U_i \otimes X_i + V : C^n \to C$, we can write it as $U_1 \otimes X_1 + \sum_{2 \leq i \leq n} U_i \otimes X_i + V : C \times C^{n-1} \to C$.

Then, taking d as $\sum_{2 \leq i \leq n} U_i \otimes X_i + V$, *llist*-arithmeticity implies that the functor $U_1 \otimes (-) + d : C \to C$ has an initial algebra $U_1^* \otimes d$ for every d. Obviously, the same can be done for every index i.

Let us recall now Bekič theorem about initial algebras of functors [6] that is important because it means that the simultaneous construction of an initial algebra for a system of n operators in n variables can be replaced by recursively constructing of initial algebras for one operator at a time.

Bekič theorem

Given two functors $F : C \times D \to C$ and $G : C \times D \to D$, let $(F^{\mu}(d), \chi_d)$ be an initial $F((-), d)$-algebra for each object d of D and suppose that there exists an initial algebra, say $< \xi, \zeta >$ with $\xi : F^{\mu}(\beta) \to \alpha$, $\zeta : G(\alpha, \beta) \to \beta$, of the functor $< F^{\mu} \circ pr_D, G >: C \times D \to C \times D$, where the first component is obtained by composing the projection $pr_D : C \times D \to D$ with the functor constructing the D-parameterized initial algebra $F^{\mu} : D \to C$; then the pair $< \chi_{\beta}, G(\xi, \beta) \bullet \zeta >$ where

- $\chi_{\beta} : F(F^{\mu}(\beta), \beta) \to F^{\mu}(\beta)$
- $G(\xi, \beta) \bullet \zeta : G(F^{\mu}(\beta), \beta) \to \beta$

is an initial algebra of the functor $< F, G >: C \times D \to C \times D$.

□

To understand the impact of this theorem in our context, let us consider a simple case where F and G are two right-linear polynomial functors over two variables[3] and C coincides with D:

$$\begin{cases} F \equiv ax + by \\ G \equiv a'x + b'y + c' \end{cases}$$

We take the initial algebra a^*by (not depending on x) associated with the first functor, when it is considered as $F((-), by)$, and then we substitute this value in the expression of G to obtain $a'a^*by + b'y + c'$. We can get from this an initial algebra for the pair $< F, G >$, i.e. for the system. Indeed, using distributivity on the right, we get $(a'a^*b + b')y + c'$ and thus, thanks to the initiality rule, we get $(a'a^*b+b')^*c'$ as the second component of the initial algebra for the functor system above.

It is worth noting that this has been possible only because there was no constant term in the definition of F. Indeed, the slightly different system

$$\begin{cases} F' \equiv ax + by + c \\ G \equiv a'x + b'y + c' \end{cases}$$

is not solvable using the same machinery. In fact, in this case, we would obtain $a^*(by + c)$ as initial algebra for $F'((-), by + c)$, and once this is substituted in G we would get $a'a^*(by + c) + b'y + c'$, but then, due to the lack of left distributivity, the initiality rule cannot be used.

[3] In the sequel we will often omit the symbol $\otimes$ using juxtaposition to replace it and use small letters for variables.

In general, for n functors, we have that, if F is F_1, then G is $< F_2, \ldots, F_n >$ and the initial algebras can be inductively obtained by performing appropriate substitutions. We fix a variable, say x_1, in the expression of F_1 and consider the parameterization of F_1 w.r.t. the sum of all the monomials not containing x_1. We calculate the parameterized initial algebra and substitute this value everywhere for x_1. Please notice that a constant functor has this constant with its identity as an initial algebra.

Summing up, in our case the first requirement of Bekič's theorem is always satisfied because our category is *llist*-arithmetic; but we have to impose additional conditions on the system of functors in order to meet the second requirement.

Given the system $\Phi \equiv < F_1, \ldots, F_n >$ with variables $x_1 \ldots, x_n$, let us now define a different indexing for both, functors and variables. This will allow us to introduce a (partial) order on the set of variables in Φ in such a way that we can exclude their mutual interference when we build the initial algebra step by step. The partial ordering, $\leq$, is obtained by using a string of natural numbers as index for every variable while guaranteeing that two different variables do not have the same index. For any two indexed variables x_s and x_t, we will write $x_s \leq x_t$ if t is a prefix of s.

We will consider *hierarchical* (*rlhs*) any system for which it is possible to introduce an indexing that satisfies a number of conditions that we will introduce below.

Definition 3. *Let $x_s \leq x_t$, we say that*

- x_s *is* ruled by x_t *if x_t appears in F_s.*
- x_s *is* recursive *if x_s appears in the expression of F_s;*
- x_s *is* strictly recursive *if it is recursive and is not ruled by any other variable.*

Definition 4. Right Linear Hierarchical Systems - rlhs
A system of right-linear functors Φ whose variables are ordered by $\leq$ is hierarchical *if it is possible to associate, as index to every functor, a common prefix of the indexes of the variables appearing in its expression (either of the same length or at most one number longer), with the only possible exception for one of the variables, in case this rules on all the others. Moreover, the indexing has to guarantee that:*

1. *the ordering of the indexes of the functors is tree-shaped;*
2. *a variable can be ruled by at most another one;*
3. *If x_i appears in F_i (it is recursive), then alternatively, either F_i does contain a constant different from* 0*, or it does contain a variable ruling on x_i;*
4. *If x_i does not appear in F_i, then all variables that are immediately smaller than x_i (one number more in their index) appear in F_i and, some of them can have a common ruling variable while the others are strictly recursive in the functor corresponding to them.*

If we look at the examples above, we have that we can index the first system as

$$\begin{cases} F_0 \equiv ax_0 + bx_\epsilon \\ F_\epsilon \equiv a'x_0 + b'x_\epsilon + c' \end{cases}$$

to obtain a *rlhs*.

Due to the presence of a constant in both the expressions, we cannot provide a similar ordering for the second system. Indeed, none of the two functors can satisfy condition 3. of Definition 4.

$$\begin{cases} F_0 \equiv ax_0 + bx_\epsilon + c \\ F_\epsilon \equiv a'x_0 + b'x_\epsilon + c' \end{cases}$$

We give now a more complex example of a right-linear system that can be indexed in such a way that it is *rlhs*. Please notice that, in the example below, we provide directly the indexed equational system. The original one can be recovered by giving different names to the variables and the functors with different indexes. After presenting the indexed system we also outline the procedure to obtain its initial algebra.

Example 1. Consider the following system

$$\Phi :< F_{211}, F_{212}, F_{21}, F_{22}, F_{11}, F_1, F_2, F_\epsilon >$$

where we have indexed functors and, accordingly, their variables.

$F_{211} \equiv cx_{211} + x_2$

$F_{212} \equiv ax_{212} + bx_2$

$F_{21} \equiv x_{211} + x_{212}$

$F_{22} \equiv cx_{22} + I$

$F_{11} \equiv ax_{11} + b$

$F_1 \equiv x_{11} + c$

$F_2 \equiv x_{21} + x_{22}$

$F_\epsilon \equiv ax_1 + x_2 + a$

All variables under x_{21} depend on the ruling variable x_2 (which becomes recursive when substitutions are made into F_2), and the corresponding functors do not contain any constant. We will start from F_{211} and F_{212}, that are leaf functors according to $\leq$. The parametrized initial algebra for F_{211} is c^*x_2 while the one for F_{212} is a^*bx_2. Since F_{211} and F_{212} have no constant and x_{211} and x_{212} have the same ruling variable, we can substitute c^*x_2 and a^*bx_2 in F_{21} to obtain $c^*x_2+a^*bx_2 = (c^*+a^*b)x_2$ thanks to right distributivity. We can now consider F_{22}. Its initial algebra is the constant c^*, thus in F_2 we can replace x_{21} and x_{22} with

$(c^* + a^*b)x_2$ and c^*, which yields $(c^* + a^*b)x_2 + c^*$ that has $(c^* + a^*b)^*c^*$ as initial algebra. We consider now F_{11}, whose initial algebra is a^*b, that we substitute in F_1 to obtain $(a^*b) + c$ as initial algebra; finally once we substitute all variables in F_ϵ with the corresponding initial algebras we get the constant $(a(a^*b+c) + (c^* + a^*b)^*c^* + a$. Which is the basis for obtaining the full solution by means of appropriate substitutions.

Theorem 1. *In a right semidistributive category C where we have a parameterized initial algebra for linear polynomial functors (parameterized llists), all right-linear hierarchical systems of functors, with chosen indexing, have a family of regular expressions as their initial algebra.*

Proof. Given a hierarchical right-linear functors system, we can find an initial algebra for it by repeatedly using the $(initiality - rule)$ above, and by relying on Bekič theorem. This theorem provides an initial algebra for a system of functors in presence of parameterized initial algebras; to take advantage of it we need to show that the restricted set of hierarchical systems satisfy its two conditions. The first condition, i.e. the existence of parameterized initial algebras for a chosen functor in a recursive variable holds by hypothesis, the second one corresponds to the fact that the reduced system (with fewer functors) obtained after substituting the initial algebra has still an initial algebra. We then proceed by induction on the length of indexes starting from the longest ones. This is possible because, by exploiting right distributivity, we can take out a (ruling) variable as a common factor from terms containing it. This is due to our definition of *rlhs*.

Let us start by considering a functor F_i which has maximal index. The expression corresponding to F_i cannot contain variables with indexes longer than i. Thus, the expression may contain x_i and at most a single variable, say x_t ruling on it (t is a strict prefix of i), moreover when such ruling variable is present the expression does not contain any constants.

Let us consider the two cases separately:

1. In case x_i is strictly recursive, we obtain as initial algebra of F_i a constant term which might be 0 in case the expression contains only 0 as a constant.
2. In case the expression of F_i contains a variable x_t ruling on x_i, the initiality rule gives a parameterized (w.r.t. x_t) initial algebra.

When substituting these terms in F_s with s an immediate prefix of i ($i = s\,n$), we obtain a sum of constants and of terms all containing the same variable x_t (due to condition 4. in Definition 4, no other term with another ruling variable can be present in F_s). We can then take x_t as a common factor. Now we distinguish two cases, if $t = s$ we can proceed as above because all variables with an index longer than t have been eliminated. If t is instead a strict prefix of s, we can operate further substitutions until we reach functor F_t possibly having other terms with the same variable and nomore variables with a longer index. In this way we have in any case reduced the system to a smaller one, still *rlhs*, producing an initial algebra at every step. At this point we can apply the procedure again. □

It could seem that a very particular kind of linear functors system is taken into account, but we can prove that any regular expression, when interpreted in C is the initial algebra of some finite linear hierarchical system of functors.

Theorem 2. *Given a cocartesian monoidal semidistributive category C, where we can interpret regular expressions in such a way that the $()^*$ operator is the llist-operator corresponding to the tensor product $\otimes$, every regular expression E can be obtained as the initial algebra of the root component of a n-tuple of right-linear hierarchical system of simple polynomial functors $< F_1, \ldots, F_n >: C^n \to C^n$.* □

In order to prove item 1. of Theorem 2 we need to transform every regular expression in normal form and we will show that any normal form can be first associated with a system of simple quadratic polynomial functors and that the system can be associated with a *rlhs* of simple linear polynomial functors. The way we obtain such a *rlhs* guarantees that the original regular expression E is the component in the initial algebra of the generated *rlhs* associated to the root functor.

Here we omit the details of the proof, it proceeds along the same lines of the corresponding one in [4] while referring to functors rather than to equations. In particular we need to use normal forms similar to that of Definition 1. in [4] and functor systems associated to them like in Definition 5. of [4]. From the functor systems we will do obtain system of quadratic functors (like in Proposition 1. in [4]) which we transform into linear ones (Proposition 2. in [4]). The fact that the regular expression E is the initial algebra of the root component of the system will descend from the construction, while the verification of the hierarchicity of the system is now almost immediate by the chosen indexing strategy.

4 Conclusions

A classical result of the theory of regular languages [8] states that we can obtain solutions of systems of linear equations over regular expressions interpreted as languages variables.

In [4] we showed that right-linear systems of equations over regular expressions, when interpreted in a category of trees, have a solution whenever they enjoy a specific property that we called *hierarchicity.*

Here, we have completed the generalisation by considering cocartesian non commutative monoidal categories where the tensor product preserves colimits and a property similar to hierarchicity is satisfied. The key requirement for this kind of categories was the presence of an iteration operator thought of as initial algebra of a linear polynomial functor. The existence of such initial algebra is a form of a one-side list arithmeticity. Now list arithmeticity is a key ingredient to develop arithmetics in a pretopos [7]: this fact could suggest further investigations about a connection between results in (possibly non deterministic) language theory and in an arithmetic based on a one-side natural number object.

References

[1] Adámek, J., Koubek, V. Least Fixed Point of a Functor. *Journal of Computer and System Sciences*, 19: 163–178, 1979.

[2] Cockett, J.R.B. *list*-arithmetic distributive categories: locoi. *Journal of Pure and Applied Algebra*, 66: 1–29, 1990.

[3] Corradini, F., De Nicola, R., Labella, A. Models of Nondeterministic Regular Expressions. *Journal of Computer and System Science*, 59: 412–449, 1999.

[4] De Nicola, R., Labella, A. Nondeterministic regular expressions as solutions of equational systems, *Theoretical Computer Science*, 302: 179–189, 2003.

[5] Labella, A. Categories with sums and right distributive tensor product. *Journal of Pure and Applied Algebra*, 178 (3): 273–296, 2003.

[6] Lehmann, D. J., Smyth, M. B. Algebraic specification of data types: a synthetic approach. *Mathematical Systems Theory*, 14 (2): 97-139, 1981.

[7] Maietti, M.E. Joyal's arithmetic universe as list-arithmetic pretopos, *Theory and Applications of Categories*, 24 (3): 39–83, 2010.

[8] Moll, R.N., Arbib, M.A., Kfoury, A.J. *An Introduction to Formal Language Theory* Springer-Verlag, Berlin, 1987.