Some Combinatorial Operators in Language Theory

J.-G. Luque, L. Mignot and F. Nicart

May 1, 2014

Contents

1 Introduction 1

2 What are operads? 2
   2.1 Operadic structures ............................................ 3
   2.2 Free operad, morphisms, suboperads, quotients etc. ............ 5

3 Multitildes operad 6
   3.1 Extended to multitilde regular expression ....................... 6
   3.2 The operators $\gg$ and $\leftrightarrow$ ......................... 8
   3.3 The Operad of multitildes ..................................... 10

4 Actions on languages 11
   4.1 An operad of boolean vectors and its action on languages .......... 11
   4.2 Action of $\mathcal{T}$ on languages ......................... 12

5 $\mathcal{T}$, RAS and POSet 14
   5.1 From $\mathcal{T}$ to RAS$^\leq$ .................................... 14
   5.2 From RAS$^\leq$ to POSet$^\leq$ .................................. 15
   5.3 Pseudotransitive tildes .................................... 16

6 Consequences and perspectives 17
   6.1 Enumeration ................................................ 17
   6.2 Finite languages ........................................... 18
   6.3 Regular languages ........................................... 20

7 Conclusion 20

Abstract

Multitildes are regular operators that were introduced by Caron et al. in order to increase the number of Glushkov automata. In this paper, we study the family of the multitilde operators from an algebraic point of view using the notion of operad. This leads to a combinatorial description of already known results as well as new results on compositions, actions and enumerations.

1 Introduction

Regular expressions have been studied from several years since they have numerous applications: pattern matching, compilation, verification, bio-informatics etc. From a well known result (Kleene [10]), regular languages can be represented by both automata and regular expressions. From an
expression, it is possible to compute an automaton whose number of states is a linear function of the alphabetical width (i.e. the number of occurrences of alphabet symbols) \[16, 9, 1, 5\]. In the opposite direction, there exists no construction providing the linearity. For instance, Ehrenfeucht and Zeiger \[7\] showed a one parameter family of automata whose shortest equivalent regular expressions have a width exponentially growing with the numbers of states. However this property occurs when the automaton is the Gluskov automaton of an expression. Note the characterization of such an automaton is due to Caron and Ziadi \[4\].

Multitilde operators have been introduced by Caron et al. \[3\] in the aim to increase expressiveness of expressions for a bounded length. The study of the equivalence of two multitilde expressions leads naturally to a notion of composition. On the other hand, there is an algebraic way to represent the compositions of operators, namely operads. The aim of this paper is to describe some properties of multitildes in terms of operads.

Usually, operads are used to encode types of algebras by describing the universal operations which act on the elements of any algebras of a given type together with the ways of composing them. To be more concise, an operad is given by a set of symbols (the operations) and a composition law which satisfies some rules such as associativity. Such a study of the compositions of operations appeared earlier in the work of M. Lazard \[11\] and was referred as analyseurs. The word operad is the contraction of operations and monad. This terminology first appeared in the field of algebraic topology in a paper of May about the geometry of iterated loop spaces \[15\]. The notion arose simultaneously in Kelly’s categorical works on coherence and the definitions used in \[15\] have been constructed after conversations between the two mathematicians.

In the 1990’s, the interest in the topic was renewed with the works of Ginzburg and Kapranov on the Koszul duality \[8\]. One of the recent developments of this theory deals with Hopf algebras and combinatorics \[6\].

Readers interested by the history of this notion and its recent development can refer to \[14, 12, 13\].

Several materials about operads are recalled in Section 2. In his PhD thesis \[17\], one of the authors introduced a composition on multitildes. We show (Section 3) that this confers an operadic structure to the set of multitildes. Throughout the paper, we describe derived structures (isomorphic operads, suboperads and quotient operads) which allow us to model several behaviors and properties of the multitilde operators. For instance, we define in Section 4 an operadic structure on boolean vectors and use it to describe the action of multitildes on languages. In Section 5, we introduce an operadic structure on POSets in the aim to formalize the equivalence of the actions of two multitildes. In Section 6, we show that the representation by POSets is optimal in the sense that two different operators have different actions on \(k\)-tuples of languages. As a consequence, we enumerate the different operations which can be encoded by a multitilde. Finally, we give applications to the representation of finite languages and also regular languages.

2 What are operads?

Operad theory has been developed in the aim to study prototypical algebras that model classical properties such as commutativity, associativity etc.. In most cases, operads are considered to study algebras with an underlying vector space but the structure can be more generally used to study \(K\)-modules on monoid (\(K\) semiring). In the most general context, operads are described in terms of category theory. In our case, since we apply the operad theory to the theory of languages, we will set \(K = B\), the boolean semiring. We have adapted the definitions of this section to the boolean context to give a simplified version of the theory but most of them admits well-known generalizations. More precisely, we describe Set-operads that are operads whose underlying graded spaces are simply sets.
2.1 Operadic structures

Operads are algebraic structures which mimic the composition of \( n \)-ary operators. More explicitly, the construction starts with an underlying graded set \( \Psi = \bigcup_{n \in \mathbb{N}} \Psi(n) \); elements of \( \Psi(n) \) are called \( n \)-ary operations. This set is endowed with functions

\[
\circ : \Psi(n) \times \Psi(k_1) \times \cdots \times \Psi(k_n) \to \Psi(k_1 + \cdots + k_n)
\]
called compositions and has a special element \( 1 \in \Psi(1) \) called identity. This confers to \( \Psi \) a structure of multicategory with one object \([12]\); more precisely the operations of the operad are the morphisms of the multicategory.

Furthermore, the compositions satisfy the two rules:

1. **Associativity:**

\[
\begin{align*}
(p \circ (p_1 \circ \cdots \circ p_{n-1} \circ p_n)) \circ (p_1 \circ \cdots \circ p_{n-1} \circ p_n) &= (p \circ (p_1 \circ \cdots \circ p_{n-1} \circ p_n)) \circ (p_1 \circ \cdots \circ p_{n-1} \circ p_n) \\
(1) 
\end{align*}
\]

2. **Identity:**

\[
\begin{align*}
p \circ (1, \ldots, 1) = 1 \circ p = p. \\
(2)
\end{align*}
\]

The structure of operad is easier to manipulate if we introduce partial composition operations \( \circ_i \) which split the composition. These operations are defined by:

\[
\begin{align*}
\circ_i : \Psi(m) \times \Psi(n) &\to \Psi(m + n - 1) \\
(p_1, p_2) &\mapsto p_1 \circ_i p_2 := p_1 \circ \left(1, \ldots, 1, p_2, 1, \ldots, 1\right)_{i-1 \times m-j-1 \times} \\
(3)
\end{align*}
\]

when \( 1 \leq i \leq n \).

Let \( p_1 \in \Psi(m) \), \( p_2 \in \Psi(n) \) and \( p_3 \in \Psi(q) \). The partial compositions satisfy the two associative rules which are deduced from the definition \([3]\) and the associativity of the compositions \([1]\):

1. **Associativity 1:**

   If \( 1 < j < i \leq n \) then

   \[
   (p_1 \circ_i p_2) \circ_j p_3 = (p_1 \circ_j p_3) \circ_{i+j-q-1} p_2 \\
   (4)
   \]

2. **Associativity 2:**

   If \( i \leq n \) then

   \[
   p_1 \circ_j (p_2 \circ_i p_3) = (p_1 \circ_j p_2) \circ_{i+j-q-1} p_3. \\
   (5)
   \]

The composition is graphically interpreted by grafting trees together; a \( n \)-array operation is represented by a \( n \)-ary tree and a composition \( \circ_i \) consists in grafting the root of a tree onto the \( i \)th leaf of another tree, the resulting tree is associated to an element of \( \Psi \).

Consider two operations \( p_1 \in \Psi(m) \) and \( p_2 \in \Psi(n) \):

\[
\begin{align*}
&\begin{array}{c}
1 \quad 2 \quad \cdots \quad m \\
\end{array} \\
&\begin{array}{c}
\text{P}_1 \\
\end{array} \\
&\begin{array}{c}
\text{P}_2 \\
\end{array}
\end{align*}
\]

The composition \( p_1 \circ_i p_2 \) is represented by the tree:
Now the two associativity rules are easily understood in terms of trees:

1. **Associativity 1:** The left hand side of (4) gives

   \[
   \begin{array}{c}
   1 \quad 2 \quad \cdots \quad i \quad i+1 \quad \cdots \quad i+n-1 \\
   1 \quad 2 \quad \cdots \quad p_3 \quad \cdots \quad m+n-1 \\
   \end{array}
   \]

   whilst the right hand side reads

   \[
   \begin{array}{c}
   1 \quad 2 \quad \cdots \quad q \\
   1 \quad 2 \quad \cdots \\
   i \quad i+1 \quad \cdots \quad i+m-1 \\
   \end{array}
   \]

2. **Associativity 2:** Drawing the left hand side of (5), we find

   \[
   \begin{array}{c}
   1 \quad 2 \quad \cdots \\
   j \quad j+1 \quad \cdots \quad j+q-1 \\
   \end{array}
   \]

   \[
   \begin{array}{c}
   1 \quad 2 \quad \cdots \quad p_3 \quad \cdots \quad m+q-1 \\
   \end{array}
   \]

   whilst the right hand side reads

   \[
   \begin{array}{c}
   1 \quad 2 \quad \cdots \\
   j \quad j+1 \quad \cdots \\
   \end{array}
   \]

   \[
   \begin{array}{c}
   1 \quad 2 \quad \cdots \quad p_3 \quad \cdots \quad m+q-1 \\
   \end{array}
   \]

2. **Associativity 2:** Drawing the left hand side of (5), we find
and the right hand side of (5) gives

In short, the compositions satisfy the same branching rules than the trees together with a correct relabeling of the leaves.

Let us give one of the simplest example of operads. Consider a set $S$ and the set of the functions $\text{Map}_S(n)$ from $S^n$ to $S$. The set $\text{Map}_S := \bigcup_{n \in \mathbb{N}} \text{Map}_S(n)$ endowed with the classical composition defined by

$$(f \circ (g_1, \ldots, g_n))(s_{11}, \ldots, s_{1k_1}, \ldots, s_{n1}, \ldots, s_{nk_n}) := f(g_1(s_{11}, \ldots, s_{1k_1}), \ldots, g(s_{n1}, \ldots, s_{nk_n}))$$

for each $f \in \text{Map}_S(n)$, $g_i \in \text{Map}_S(k_i)$, $s_{ij} \in S$, is an operad.

### 2.2 Free operad, morphisms, suboperads, quotients etc.

The definition of the morphism induces the existence of operads with universal properties called free operads. Let $G = (G_k)_k$ be a collection of sets, the set $\text{Free}_G(n)$ is the set of planar rooted trees with $n$ leaves with labeled nodes where nodes with $k$ branches are labeled by the elements of $G_k$. The free operad on $G$ is obtained by endowing the set $\text{Free}_G = \bigcup_{n \in \mathbb{N}} \text{Free}_G(n)$ with the composition $p_1 \circ_i p_2$
which consists in grafting the $i$th leaf of $p_1$ with the root of $p_2$. Note that $\text{Free}_G$ contains a copy of $G$ which is the set of the trees with only one inner node (the root) labeled by elements of $G$; for simplicity we will identify it with $G$. The universality means that for any map $\phi : G \to \Psi$ it exists a unique morphism of operad $\phi : \text{Free}_G \to \Psi$ such that $\phi(g) = \phi(g)$ for each $g \in G$.

Consider the equivalence relation $\equiv$ on $\text{Free}_G$ defined by $p_1 \equiv p_2$ if and only if $\phi(p_1) = \phi(p_2)$. Obviously this relation is compatible with the composition in the sense that $p_1 \equiv p_1'$ and $p_2 \equiv p_2'$ implies $p_1 \circ p_2 \equiv p_1' \circ p_2'$. Hence, the operad $\Psi$ can be defined as a quotient (up to an isomorphism) of the free operad: $\Psi = \text{Free}_G/\equiv$. More generally, when an equivalence relation $\equiv$ on the elements of an operad $\Psi$ is compatible with the composition, this induces a structure of operad on the set $\Psi/\equiv$; the resulting operad is called the quotient of $\Psi$ by $\equiv$.

The dual notion of quotient is the notion of suboperad whose definition is very classical: a suboperad of an operad $\Psi$ is a subset of $\Psi$ which contains 1 and is stable by composition.

Consider a set $S$ together with an action of an operad $\Psi$. That is: for each $p \in \Psi(n)$ we define a map $p : S^n \to S$. We will say that $S$ is a $\Psi$-module if the action of $\Psi$ is compatible with the composition in the following sense: for each $p_1 \in \Psi(m)$, $p_2 \in \Psi(n)$, $1 \leq i \leq m$, $s_1, \ldots, s_{m+n-1} \in S$ one has:

$$p_1(s_1, \ldots, s_{i-1}, p_2(s_i, \ldots, s_{i+n-1}), s_{i+n}, \ldots, s_{m+n-1}) = (p_1 \circ p_2)(s_1, \ldots, s_{m+n-1}),$$

i.e. there is a morphism of operads from $\text{Map}_S$ to $\Psi$.

3 Multitildes operad

Multitildes operators have been introduced by Caron et al. [3] in order to increase the number of regular languages represented by an expression of a fixed width. The only operation involved in the computation of the languages they denote is the addition of the empty word in several catenation factors of the main interest of multitildes operators is that, for any simple regular expression with no star, one can define new regular expressions, the EMTREs (Extended to Multitilde Regular Expression) [3]. One of the main interest of multitildes operators is that, for any simple regular expression with no star, there exists an equivalent EMTRE with only one k-ary operator which is a multitilde. We will see that this equivalence involves a natural composition which defines a structure of operad.

In this section, we first recall the main definitions and results about multitildes then we describe in more details the operad structure acting on languages.

3.1 Extended to multitilde regular expression

Let $\Sigma$ be an alphabet and $\epsilon$ be the empty word. A regular expression $E$ over $\Sigma$ is inductively defined by $E = \emptyset$, $E = \epsilon$, $E = a$, $E = (F \cdot G)$, $E = (F + G)$, $E = (F')$ where $a \in \Sigma$ and $F$, $G$ are regular expressions. The language denoted by the expression $E$ is inductively defined by $L(\emptyset) = \emptyset$, $L(\epsilon) = \{\epsilon\}$, $L(a) = \{a\}$, $L(F + G) = L(F) \cup L(G)$, $L(F \cdot G) = L(F) \cdot L(G)$, $L(F') = L(F)^\epsilon$ where $a \in \Sigma$ and $F$, $G$ are regular expressions.

Let $i, j, n$ be three positive integers. Let $(E_1, \ldots, E_n)$ be a list of $n$ expressions. The catenation $E_i \cdot E_{i+1} \cdots E_j$ is denoted by $E_{i\cdots j}$ and for convenience the language $L(E_{i\cdots j})$ equals $\{\epsilon\}$ when $i > j$. A list $(E_1, E_{i+1}, \ldots, E_j)$ is denoted by $E_{1_{\ldots j}}$. Given a set $S$, we denote by $\#S$ the number of elements of $S$. Suppose that $i \leq j$. We set $[i, j] = \{i, i+1, \ldots, j-1, j\}$, $\lfloor i, j \rfloor_S = \{(k, k') \mid k, k' \in [i, j] \text{ and } k \leq k'\}$ and $S_n = P([1, n]_S)$ the set of the subsets of $[1, n]_S$. The set of indices of $S \in S_n$ is $I_S = [1, \#S]$ and it holds $S = \{([i_k, f_k])_{k \in I_S} \mid \text{for all } k \in I_S, (i_k, f_k) \in [1, n]_S^2\}$.

A multitilde of arity $n$ is a formal symbol $\sim_T$ where $T \subset S_n$. Remark for convenience, we use the same symbol to denote several operations with different arities. For instance, the multitilde $\sim_{(1,3)}$
should be $n$-ary for any $n \geq 3$. These symbols together with the regular operations define a new family of expressions.

**Definition 1 (3).** An Extended to Multitilde Regular Expression (EMTRE) over an alphabet $\Sigma$ is inductively defined by:

\[
E = \emptyset, \quad E = a, \quad E = (F + G), \quad E = (F \cdot G), \quad E = (F^*), \quad E = (\sim_i(T_{1,n})),
\]

with $a \in \Sigma, F$ and $G$ are two EMTREs, $E_{1,n}$ a list of EMTREs and $T$ a list in $S_n$.

Now let us recall how to extend the notion of languages denoted by regular expressions to EMTREs, by defining the action of multitilde operators.

A set $S \in S_n$ is said free if and only if for all $(i, f), (i', f')$ in $S$ such that $(i, f) \neq (i', f')$, the condition $[i, f] \cap [i', f'] = \emptyset$ holds. Let $S \in S_n$, we denote by $\mathcal{F}(S)$ the set of all free subsets of $S$.

Let $L_1, \ldots, L_n$ be a list of languages over an alphabet $\Sigma$ and let $T$ be a list in $S_n$. We set $\mathcal{W}_T(L_1, \ldots, L_n) = L_1' \cdots L_n'$ where

$$L_k' = \begin{cases} \{e\} & \text{if } k \in \bigcup_{(i,f)\in T} [i, f] \\ L_k & \text{otherwise.} \end{cases}$$

**Definition 2.** Let $L_{1,n}$ be a list of languages over an alphabet $\Sigma$. Let $T$ be a list in $S_n$. The language $\sim_T(L_{1,n})$ is defined by:

$$\sim_T(L_{1,n}) := \bigcup_{S \in \mathcal{F}(T)} \mathcal{W}_S(L_{1,n}).$$

Let $E_{1,n}$ be a list of EMTREs, the language denoted by $\sim_T(E_{1,n})$ where $T \in S_n$ is inductively defined by

$$L(\sim_T(E_{1,n})) := \sim_T(L(E_1), \ldots, L(E_n)).$$

As usual two EMTREs $E$ and $F$ are said equivalent if they denote the same language; we write $E \equiv F$.

Note all these definitions are slight rewordings to those given in [3].

Every regular expression $E$ that does not use any star operator can be turned into an equivalent expression $F = \sim_T(e_1, \ldots, e_n)$ where $\sim_T$ is a $S_n$ and $\forall k \in [1, n], e_k \in \Sigma \cup \{\emptyset\}$. Conversion formulas are given Table [1]

| $\emptyset$ | $\sim_0(\emptyset)$ |
|-------------|----------------------|
| $\epsilon$ | $\sim_{(1,3)}(\emptyset)$ |
| $a$ | $\sim_0(a)$ |

| $E_1 + E_2$ | $\sim_{(1,2),(2,3)}(E_1', \emptyset, E_2')$ |
|-------------|---------------------------------|
| $E_1 \cdot E_2$ | $\sim_{0}(E_1', E_2')$ |

**Table 1: Multitildes conversion.**

One of the authors has shown in [17] how to compose multitilde operators preserving languages:

$$L\left(\sim_T(E_1, \ldots, E_{k-1}, \sim_T(E'_1, \ldots, E'_{k+1}), \ldots, E_n)\right) \equiv L\left(\sim_{\cap T}(E_1, \ldots, E_{k-1}, E'_1, \ldots, E'_{k+1}, \ldots, E_n)\right).$$

We do not recall the original definition here but we describe this operation using new operators, namely $\Rightarrow$ and $\mathcal{O}$ in the sequel of this section. We show that this composition endows the set of multitildes with a structure of operad.
3.2 The operators \( \gg \) and \( \circ \rightarrow \)

The first operators to be defined are the \( \gg \) operators, parametrized by any integer \( k \). Any operator \( \gg_k \) will increase both elements of a couple \((x, y)\) of integers by \( k \):

\[
\gg_k(x, y) := (x + k, y + k).
\]

These operators are commutative ones:

\[
\gg_k(\ell)(x, y) := \ell(\gg_k(x, y))
\]

Indeed, both of the expressions result in the same couple:

\[
\gg_k(\gg_k(x, y)) = (x + k + \ell, y + k + \ell) = \gg_k(\gg_k(x, y))
\]

The second family of operators is the set of \( \circ \rightarrow \) operators for any integers \( n, k \), which shift elements in couples by inserting \( n \) elements in position \( k \) (see Example 1 for details):

\[
k_n \circ \rightarrow (x, y) := \begin{cases} 
(x, y) & \text{if } y < k, \\
(x, y + n - 1) & \text{if } x \leq k \leq y, \\
\llbracket i, \infty \rrbracket(x, y) & \text{otherwise.}
\end{cases}
\]

Example 1. Let \( n \) be a positive integer, and \( x \leq y \) be two integers in \([1, n]\). The operators \( k_n \circ \rightarrow \) transforms the couple \((x, y)\) performing an insertion of \( n \) elements at position \( k \) presented in Figure 1. After an insertion of 6 elements in position 5 (using the operator \( 5_6 \circ \rightarrow \)), couples \((1, 3), (3, 7)\) and \((7, 8)\) are respectively transformed into \((1, 3), (3, 12)\) and \((12, 13)\).

![Figure 1: The \( 5_6 \circ \rightarrow \) operator.](image)

For convenience, definition of \( k_n \circ \rightarrow \) operators can be rewritten using the characteristic function \( \mathbb{1}_E(x) \) defined for any integer set \( E \) and any integer \( x \) by:

\[
\mathbb{1}_E(x) = \begin{cases} 
1 & \text{if } x \in E, \\
0 & \text{otherwise.}
\end{cases}
\]

Since we will only use interval, we will denote by \( \mathbb{1}_k(x) \) for any integer \( k \) the function \( \mathbb{1}_{[k, \infty)}(x) \).

Lemma 1. Let \( k, n, x, y \) be four integers. Then:

\[
k_n \circ \rightarrow (x, y) = (x + \mathbb{1}_{k+1}(x) \times (n - 1), y + \mathbb{1}_k(y) \times (n - 1)).
\]

Next lemma presents the commutation of \( \gg \) and \( \circ \rightarrow \).

Lemma 2. Let \( x, y, i, m \) be five integers with \( x \leq y \). Then:

\[
\begin{split}
\gg_n \circ \rightarrow (\circ \rightarrow (x, y)) &= \circ \rightarrow (\gg_n(x, y)).
\end{split}
\]

Proof. From formulas for \( \gg \) and \( \circ \rightarrow \) operators:

\[
\begin{split}
\gg_n \circ \rightarrow (\circ \rightarrow (x, y)) &= \gg_n(x + \mathbb{1}_{i+1}(x) \times (m - 1), y + \mathbb{1}_i(y) \times (m - 1)) \\
&= (x + n + \mathbb{1}_{i+1}(x) \times (m - 1), y + n + \mathbb{1}_i(y) \times (m - 1)) \\
&= (x + n + \mathbb{1}_{i+n+1}(x + n) \times (m - 1), y + n + \mathbb{1}_i(y + n) \times (m - 1)) \\
&= \circ \rightarrow (x + n, y + n) \\
&= \circ \rightarrow (\gg_n(x, y))
\end{split}
\]

\(\square\)
The $\circ\to$ operators commute in a particular way, described as follows:

**Lemma 3.** Let $x, y, m, n, k, \ell$ be six integers with $x \leq y$ and $k < \ell$. Then:

\[
\begin{align*}
\circ \to (^{\ell,n}_{k,m})(\circ \to (x, y)) &= \circ \to (^{\ell+1,n}_{k,m})(x, y) \\
\end{align*}
\]

**Proof.** From formulas for $\circ\to$ operators:

\[
\begin{align*}
\circ \to (^{\ell,n}_{k,m})(x, y) &= \circ \to (x + \circ \to \ell+y\times(n-1), y + \circ \to \ell+y\times(n-1)) \\
&= (x', y') \\
\end{align*}
\]

with $x' = x + \circ \to \ell+y\times(n-1) + \circ \to \ell+k+y\times(n-1) \times (m-1)$
and $y' = y + \circ \to \ell+y\times(n-1) + \circ \to \ell+k+y\times(n-1) \times (m-1)$.

Since $k < \ell$, then

\[
\begin{align*}
\circ \to \ell+k+y\times(n-1) &= \circ \to \ell+(m+1) \times (x + \circ \to \ell+k+y\times(n-1)) \\
\circ \to \ell+k+y\times(n-1) &= \circ \to \ell+k+y\times(n-1) \\
\end{align*}
\]

Consequently,

\[
\begin{align*}
x' &= x + \circ \to \ell+k+y\times(n-1) + \circ \to \ell+m+y\times(n-1) \times (m-1) \\
y' &= y + \circ \to \ell+k+y\times(n-1) + \circ \to \ell+m+y\times(n-1) \times (m-1) \\
\end{align*}
\]

Finally,

\[
\begin{align*}
(x', y') &= \circ \to \ell+m+y\times(n-1), y + \circ \to \ell+k+y\times(n-1) \\
&= \circ \to (x, y). \\
\end{align*}
\]

Finally, another combination of $\circ\to$ operators:

**Lemma 4.** Let $i, n, k, m, x, y$ be six integers such that $x \leq y$ and $0 \leq i < n$. Then:

\[
\begin{align*}
\circ \to (^{i,n}_{k,m})(\circ \to (x, y)) &= \circ \to (^{i+1,n}_{k,m})(x, y) \\
\end{align*}
\]

**Proof.** From formulas for $\circ\to$ operators:

\[
\begin{align*}
\circ \to (^{i,n}_{k,m})(\circ \to (x, y)) &= \circ \to (x + \circ \to \ell+k+y\times(n-1), y + \circ \to \ell+k+y\times(n-1)) \\
&= (x', y') \\
\end{align*}
\]

with $x' = x + \circ \to \ell+k+y\times(n-1) + \circ \to \ell+k+y\times(n-1) \times (m-1)$
and $y' = y + \circ \to \ell+k+y\times(n-1) + \circ \to \ell+k+y\times(n-1) \times (m-1)$.

Since $0 \leq i < n$, then

\[
\begin{align*}
\circ \to \ell+k+y\times(n-1) &= \circ \to \ell+k+y\times(n-1) \\
\circ \to \ell+k+y\times(n-1) &= \circ \to \ell+k+y\times(n-1) \\
\end{align*}
\]

Finally,

\[
\begin{align*}
(x', y') &= (x + \circ \to \ell+k+y\times(n-1), y + \circ \to \ell+k+y\times(n-1) \\
&= \circ \to (x, y). \\
\end{align*}
\]

The definition domain of $\circ\to$ operators can be extended to sets of couples by the following way. Let $T \subset [1, n]^2$ for any integer $n$:

\[
\begin{align*}
\circ \to (T) &= \{ \circ \to ((x, y)) : (x, y) \in T \}, \\
\circ \to (T) &= \{ \circ \to ((x, y)) : (x, y) \in T \}. \\
\end{align*}
\]

From an algebraic point of view, operators $\circ\to$ and $\Rightarrow$ are linearly extended. Relations are still valid:

**Proposition 1.** Let $k, \ell, n, i, m$ be five integers, and let $T$ be a set of couples $(x, y)$ with $x \leq y$. Then:

\[
\begin{align*}
1. \Rightarrow (\Rightarrow (T)) &= \Rightarrow (\Rightarrow (T)), \\
2. \Rightarrow_\ell (\Rightarrow_\ell (T)) &= \Rightarrow_\ell (\Rightarrow_\ell (T)), \\
3. \circ \to (\Rightarrow_\ell (T)) &= \circ \to (\Rightarrow_\ell (T)) \\
\end{align*}
\]
Proof. According to definition of \( \circ \):
\[
T_1 \circ_k (T_2 \circ_i T_3) = T_1 \circ_k \left( \circ^{i_p}(T_2) \cup \circ^{i-1}(T_3) \right)
\]
\[
= \circ^{k,n+p-1}(T_1) \cup \circ^{k-1,i_p}(T_2) \cup \circ^{k-1,i-1}(T_3).
\]

Following Proposition 1 on the sets of couples computed in the right member:

1. from relation (1): \( \circ^{k,n+p-1}(T_1) \cup \circ^{k-1,i_p}(T_2) \cup \circ^{k-1,i-1}(T_3) \)
2. from relation (2): \( \circ^{i_p}(T_2) \cup \circ^{i-1}(T_3) \)

□

3.3 The Operad of multitildes

Multitildes are \( k \)-ary operators for any integer \( k > 0 \) represented by a subset of \( [1, n]^{\le k} \). The set of \( k \)-ary multitildes is denoted by \( \mathcal{T}_k \). The set of multitildes (i.e. \( \bigcup_{k \in \mathbb{N}} \mathcal{T}_k \)) is denoted by \( \mathcal{T} \).

Let \( k, n \) be two integers such that \( 1 \leq k \leq n \). The partial composition \( \circ_k \) of multitildes can be expressed from \( \circ \) and \( \gg \gg \) operators as follows:
\[
\circ_k : \mathcal{T}_m \otimes \mathcal{T}_n \to \mathcal{T}_{m+n-1}
\]
\[
T_1 \circ_k T_2 = (T_1) \circ (T_2) \gg \gg \gg (T_3)
\]

Note this composition is straightforwardly the same than those defined by one of the authors in [17]. It satisfies the property of commutativity and associativity defining an operad.

Proposition 2 (Commutativity). Let \( m, n, p, k, \ell \) be five integers such that \( 1 \leq k < \ell \leq m \). Let \( T_1 \in \mathcal{T}_m \), \( T_2 \in \mathcal{T}_n \) and \( T_3 \in \mathcal{T}_p \) be two multitildes. Then:
\[
(T_1 \circ_\ell T_2) \circ_k T_3 = (T_1 \circ_k T_3) \circ_{\ell + p - 1} T_2.
\]

Proof. From the definition of composition:
\[
(T_1 \circ_\ell T_2) \circ_k T_3 = \circ^{k_p}(\circ^{\ell_n}(T_1)) \cup \circ^{k_p}(\circ^{\ell-1_n}(T_2)) \cup \circ^{k-1}(T_3).
\]

Following Proposition 1 on the sets of couples computed in the right member:

1. From relation (3): \( \circ^{k_p}(\circ^{\ell_n}(T_1)) \)
2. Since \( k < \ell \), from relation (2):
\[
\circ^{\ell_p}(\circ^{\ell-1_n}(T_2)) = \circ^{\ell_p}(\circ^{\ell-1_n}(T_2)) = \circ^{\ell_p}(\circ^{\ell-1_n}(T_2)).
\]

Consequently: \( (T_1 \circ_\ell T_2) \circ_k T_3 = \circ^{k_p}(\circ^{\ell_n}(T_1)) \cup \circ^{\ell_p}(\circ^{\ell-1_n}(T_2) \cup \circ^{k-1}(T_3)).
\]

Then:
\[
(T_1 \circ_\ell T_2) \circ_k T_3 = \circ^{k_p}(\circ^{\ell_n}(T_1)) \cup \circ^{\ell_p}(\circ^{\ell-1_n}(T_2) \cup \circ^{k-1}(T_3)) \cup \circ^{\ell_p}(\circ^{\ell-1_n}(T_2) \cup \circ^{k-1}(T_3)).
\]

Finally:
\[
(T_1 \circ_\ell T_2) \circ_k T_3 = (T_1 \circ_k T_3) \circ_{\ell + p - 1} T_2.
\]

The associativity of \( \circ_k \) is shown by the following proposition.

Proposition 3. Let \( m, n, p, i, k \) be five integers such that \( i < n \). Let \( T_1 \in \mathcal{T}_m \), \( T_2 \in \mathcal{T}_n \) and \( T_3 \in \mathcal{T}_p \) be three multitildes. Then:
\[
T_1 \circ_k (T_2 \circ_i T_3) = (T_1 \circ_k T_2) \circ_{k+1} T_3.
\]

Proof. According to definition of \( \circ_k \):
\[
T_1 \circ_k (T_2 \circ_i T_3) = T_1 \circ_k \left( \circ^{i_p}(T_2) \cup \circ^{i-1}(T_3) \right)
\]
\[
= \circ^{k,p+1}(T_1) \cup \circ^{k-1,i_p}(T_2) \cup \circ^{k-1,i-1}(T_3).
\]

Following Proposition 1 on the sets of couples computed in the right member:

1. from relation (1): \( \circ^{k,p+1}(T_1) \cup \circ^{k-1,i_p}(T_2) \cup \circ^{k-1,i-1}(T_3) \)
2. from relation (2): \( \circ^{i_p}(T_2) \cup \circ^{i-1}(T_3) \)
3. Since \( i < n \), from relation (4): \( k_{i,n} + p - 1 \to (T_1) = k_{i+1,n} \to (O \to (T_1)). \)

Consequently: \( T_1 \circ_k (T_2 \circ_k T_3) = (O \to (T_1)) \cup (O \to (T_2)) \cup (O \to (T_3)). \)
The right side of the expression is the definition of \( \alpha_{k+1} \) and finally:
\[
T_1 \circ_k (T_2 \circ_k T_3) = (O \to (T_1)) \cup (O \to (T_2)) \circ_{k+1} T_3 = (T_1 \circ_k T_2) \circ_{k+1} T_3.
\]

From the partial composition \( \circ_k \), ones can define the composition \( \circ \) as follows:
\[
\circ : \left\{ T_m \otimes T_{n_1} \otimes \cdots \otimes T_{n_m} \to T_{n_1 + \cdots + n_m} \right\} \circ (T_{1_1}, \ldots, T_{1_m}) = (\cdots (T_1 \circ_m T_m') \circ_{m-1} T_{m-1}') \cdots) \circ_1 T_1'.
\]

**Theorem 1.** The structure \((T, \circ)\) is an operad.

## 4 Actions on languages

In this section, we recover the construction of language associated to a multitildes expression \((\mathcal{B})\) by means of an operad based on boolean vectors. We show the connection between this operad and the operad \(T\).

### 4.1 An operad of boolean vectors and its action on languages

Denote by \( \mathcal{B} = \{0, 1\} \) and \( \mathcal{B}_n = \mathcal{P}(\mathcal{B}^n) \). We endow the set \( \mathcal{B} = \bigcup_{n \in \mathbb{N}} \mathcal{B}_n \) with a structure of operad whose composition is as follows:

Let \( E \in \mathcal{B}_m \) and \( F \in \mathcal{B}_n \), the set \( E \circ_k F \) is defined by
\[
E \circ_k F := \{ [e_1, \ldots, e_{k-1}, e_k f_1, \ldots, e_k f_n, e_{k+1}, \ldots, e_m] : [e_1, \ldots, e_m] \in E, [f_1, \ldots, f_n] \in F \}.
\]

**Proposition 4.** The structure \((\mathcal{B}, \circ)\) is an operad.

**Proof.** The set \( 1 := \{[1]\} \) is the identity for the composition \( \circ_k \). Indeed,
\[
1 \circ_k F = F
\]
and
\[
E \circ_k \{1\} := \{ [e_1, \ldots, e_{k-1}, 0, e_{k+1}, \ldots, e_m] : [e_1, \ldots, e_{k-1}, 0, e_{k+1}, \ldots, e_m] \in E \} \cup \{ [e_1, \ldots, e_{k-1}, 1, e_{k+1}, \ldots, e_m] : [e_1, \ldots, e_{k-1}, 1, e_{k+1}, \ldots, e_m] \in E \} = E.
\]

It remains to prove the two associativity rules:

1. Let \( E \in \mathcal{B}_m, F \in \mathcal{B}_n, G \in \mathcal{B}_p \) and \( 1 \leq j < i \leq m \). One has
\[
(E \circ_j F) \circ_j G = \{ [e_1, \ldots, e_{j-1}, e_j g_1, \ldots, e_j g_n, e_{j+1}, \ldots, e_{i-1}, e_i f_1, \ldots, e_i f_n, e_{i+1}, \ldots, e_n] : [e_1, \ldots, e_m] \in E, [f_1, \ldots, f_n] \in F, [g_1, \ldots, g_p] \in G \} = \{ [e_1, \ldots, e_{j-1}, e_j g_1, \ldots, e_j g_n, e_{j+1}, \ldots, e_m] : [e_1, \ldots, e_m] \in E, [g_1, \ldots, g_p] \in G \} \circ_{j+p-1} F = (E \circ_j G) \circ_{j+p-1} F.
\]

2. Let \( E \in \mathcal{B}_m, F \in \mathcal{B}_n, G \in \mathcal{B}_p \) and \( 1 \leq i \leq n \) and \( 1 \leq j \leq m \). One has
\[
E \circ_j (F \circ_l G) = \{ [e_1, \ldots, e_{j-1}, e_j f_1, e_j g_1, \ldots, e_j g_p, e_{j+1}, \ldots, e_{i-1}, e_i f_1, \ldots, e_i f_n, e_{i+1}, \ldots, e_m] : [e_1, \ldots, e_m] \in E, [f_1, \ldots, f_n] \in F, [g_1, \ldots, g_p] \in G \} = (E \circ_j F) \circ_{i+j-1} G.
\]
This proves that $\mathcal{B}$ is an operad.

We define the action of $\mathcal{B}_n$ on the languages by

$$E(L_1, \ldots, L_n) = \bigcup_{[e_1, \ldots, e_n] \in E} L_1^{e_1} \cdot L_2^{e_2} \cdot \cdots \cdot L_n^{e_n}.$$

for each $E \in \mathcal{B}$.

**Proposition 5.** The sets $2^\Sigma$ (the set of the languages over $\Sigma$) and $\text{Reg}(\Sigma^*)$ (the set of regular languages over $\Sigma$) are both $\mathcal{B}$-modules.

**Proof.** Let us show first the compatibility of the action previously defined with the composition. One has

$$E(L_1, \ldots, L_{i-1}, F(L_i, \ldots, L_{i+n-1}), L_{i+n}, \ldots, L_{m+n-1}) = \bigcup_{[e_1, \ldots, e_n] \in E} L_1^{e_1} \cdot L_2^{e_2} \cdot \cdots \cdot L_{i-1}^{e_i} (L_i^{e_i} \cdot F(L_{i+1}^{e_{i+1}} \cdot \cdots \cdot L_{i+n-1}^{e_{i+n-1}})^{e_i} L_{i+n}^{e_{i+n}} \cdots L_{m+n-1}^{e_{m+n-1}}.$$

Since $e_i \in \{0, 1\}$,

$$E(L_1, \ldots, L_{i-1}, F(L_i, \ldots, L_{i+n-1}), L_{i+n}, \ldots, L_{m+n-1}) = \bigcup_{[e_1, \ldots, e_n] \in E} L_1^{e_1} \cdot L_2^{e_2} \cdot \cdots \cdot L_{i-1}^{e_i} L_i^{e_i} L_{i+1}^{e_{i+1}} \cdots L_{i+n-1}^{e_{i+n-1}} \cdots L_{m+n-1}^{e_{m+n-1}}$$

$$= \bigcup_{[e_1, \ldots, e_{i+n-1}] \in E \cup F} L_1^{e_1} \cdot \cdots \cdot L_{i+n-1}^{e_{i+n-1}}$$

$$= E \circ_i F(L_1, \ldots, L_{i+n-1}).$$

Hence, $2^\Sigma$ is a $\mathcal{B}$-module. Note a finite union of catenation of regular languages is regular, so each operator $E \in \mathcal{B}$ maps $\text{Reg}(\Sigma^*)$ on $\text{Reg}(\Sigma^*)$. This implies that it is also a $\mathcal{B}$-module.

### 4.2 Action of $\mathcal{T}$ on languages

We define a map $V: \mathcal{T}_k \to \mathcal{B}_k$ by

$$V(T) = \{v(S) : S \in \mathcal{T}(T)\}$$

with

$$v(S) = (v_1, \ldots, v_k) \text{ where } v_j = \begin{cases} 0 & \text{if } j \in \bigcup_{(x, y) \in \mathbb{Z}} [x, y] \\ 1 & \text{otherwise.} \end{cases}$$

**Example 2.** Consider $T = \{(1, 2), (2, 3), (3, 4), (4, 4)\} \in \mathcal{T}_4$. The images of the elements of $\mathcal{T}(T)$ are

$$v(((1, 2), (3, 4))) = (0, 0, 0, 0)$$

$$v(((1, 2), (4, 4))) = (0, 0, 1, 0)$$

$$v(((2, 3), (4, 4))) = (1, 0, 0, 0)$$

$$v(((1, 2))) = (0, 0, 1, 1)$$

$$v(((2, 3))) = (1, 0, 0, 1)$$

$$v(((3, 4))) = (1, 1, 0, 0)$$

$$v(((4, 4))) = (1, 1, 1, 0)$$

$$v(\emptyset) = (1, 1, 1, 1)$$

Hence

$$V(T) = \{(0, 0, 0, 0), (0, 0, 1, 0), (1, 0, 0, 0), (0, 0, 1, 1), (1, 0, 0, 1), (1, 1, 0, 0), (1, 1, 1, 0), (1, 1, 1, 1)\}$$

**Theorem 2.** The map $V$ is a morphism of operads.
Proof. First the image of $\emptyset \in T_1$ by $V$ is the set 1.
Now let us examine the image of a composition. Let $T_1 \in T_m, T_2 \in T_n$ and $1 \leq k \leq m$.
We need the following lemma which explains how the composition modifies the free subsets.

**Lemma 5.**

$$\mathcal{F}(T_1 \circ_k T_2) = \{\circ \mapsto S : S \in \mathcal{F}(T_1)\} \cup$$
$$\{\circ \mapsto T : S \in \mathcal{F}(T_1), T \in \mathcal{F}(T_2),$$
$$\forall (x, y) \in S, x \geq k \text{ or } y < k - 1\}$$  \hspace{1cm} (8)

**Proof.** For simplicity, we will denote by $\mathcal{R}$ the right hand side of equality (8). Note $\mathcal{R} \subseteq \mathcal{F}(T_1 \circ_k T_2)$ is obtained remarking that each element of $\mathcal{R}$ is free.
Now let us prove the inclusion $\mathcal{F}(T_1 \circ_k T_2) \subseteq \mathcal{R}$. Straightforwardly from the definition of $\mathcal{F}$, we have the inclusion

$$\mathcal{F}(T_1 \circ_k T_2) \subset \{\circ \mapsto S \uplus \mathcal{R} : S \in \mathcal{F}(T_1), T \in \mathcal{F}(T_2)\}.$$  

Hence, if $R \in \mathcal{F}(T_1 \circ_k T_2)$, there exist $S \in \mathcal{F}(T_1)$ and $T \in \mathcal{F}(T_2)$ such that $R = \circ \mapsto S \uplus \mathcal{R}$. If $T = \emptyset$ then $R \in \{\circ \mapsto S : S \in \mathcal{F}(T_1)\} \subset \mathcal{R}$. Suppose $T \neq \emptyset$ and $R \notin \mathcal{R}$. Consequently there exist two couples $(x, y + n), (x' + k - 1, y' + k - 1) \in R$ with $(x, y) \in S, x \leq k, y > k - 1$ and $(x', y') \in T$. Hence, $[x, y + n] \cap [x' + k - 1, y' + k - 1] = [x' + k - 1, y' + k - 1] / \emptyset$. This contradicts the fact that $R$ is free, proves the second inclusion and then the equality of the two sets. \hfill \Box

End of the proof From equation (8) we have

$$V(T_1 \circ_k T_2) = \{v(\circ \mapsto S) : S \in \mathcal{F}(T_1)\} \cup$$
$$\{v(\circ \mapsto T) : S \in \mathcal{F}(T_1), T \in \mathcal{F}(T_2),$$
$$\forall (x, y) \in S, x \geq k \text{ or } y < k - 1\}.$$  

Set $V_1 := \{v(\circ \mapsto T) : S \in \mathcal{F}(T_1), T \in \mathcal{F}(T_2), \forall (x, y) \in S, x \geq k \text{ or } y < k - 1\}$ and $V_2 := \{v(\circ \mapsto S) : S \in \mathcal{F}(T_1)\}$. We have

$$V_1 = \{v(\circ \mapsto S \uplus \mathcal{R}) : S \in \mathcal{F}(T_1), T \in \mathcal{F}(T_2), v(s) = 1\}.$$  

Hence,

$$V_1 = \{(s_{1}, \ldots, s_{k-1}, s_{k}t_{1}, \ldots, s_{k}t_{n}, s_{k+1}, \ldots, s_{m}) : (s_{1}, \ldots, s_{m}) \in V(T_{1}),$$
$$(t_{1}, \ldots, t_{n}) \in V(T_{2}), s_{k} = 1\}.$$  

On another hand $V_2$ splits as

$$V_2 = V_{2}^{0} \cup V_{2}^{1}$$

where

$$V_{2}^{i} = \{v(\circ \mapsto S) : S \in \mathcal{F}(T_{1}) \text{ and } v(S)_{k} = i\}.$$  

We have $V_{2}^{1} \subset V_{1}$ (it suffices to put $T = \emptyset$ in $V_1$) and

$$V_{2}^{0} = \{(s_{1}, \ldots, s_{k-1}, s_{k}t_{1}, \ldots, s_{k}t_{n}, s_{k+1}, \ldots, s_{m}) : (s_{1}, \ldots, s_{m}) \in V(T_{1}),$$
$$(t_{1}, \ldots, t_{n}) \in V(T_{2}), s_{k} = 0\}.$$  

It follows that $V(T_1 \circ_k T_2) = V(T_1) \circ_k V(T_2)$. \hfill \Box

**Corollary 1.** The sets $2^{2^\Sigma}$ and Reg($\Sigma^*$) are both $\mathcal{T}$-module.

**Proof.** It suffices to set $T(L_{1}, \ldots, L_{k}) := V(T)(L_{1}, \ldots, L_{k})$. \hfill \Box

Note the action of $\mathcal{T}$ on languages matches with the definition of $\sim_{\mathcal{T}}$.  

13
Proposition 6.

\[ \sim_{T}(L_1, \ldots, L_k) = T(L_1, \ldots, L_k) \]

Proof. We have

\[ \mathcal{W}_S(L_1, \ldots, L_k) = L'_1 \cdots L'_k \]

with

\[ L'_i = \begin{cases} \{ e \} & \text{if } i \in \bigcup_{(x,y) \in S} [x, y] \\ L_k & \text{otherwise.} \end{cases} \]

Hence,

\[ \mathcal{W}_S(L_1, \ldots, L_k) = L^r(S)_1 \cdots L^r(S)_k. \]

And the result follows from

\[ \sim_{T}(L_1, \ldots, L_k) = \bigcup_{S \in \bar{\mathcal{F}}(T)} \mathcal{W}_S(L_1, \ldots, L_k) = \bigcup_{v \in V(T)} L^v_1 \cdots L^v_k = T(L_1, \ldots, L_k). \]

\[ \square \]

5 \( \mathcal{T} \), RAS and POSet

In this section, we give a combinatorial description of the operad \( \mathcal{T} \). In particular we prove that it is isomorphic to an operad whose underlying set is the set \( \text{RAS}^\leq \) of reflexive and (necessarily) antisymmetric subrelations of the order \( \leq \) on \( \mathbb{N} \setminus \{0\} \). In our context, we define a quotient \( \text{POSet}^\leq \) of \( \text{RAS}^\leq \) whose elements are indexed by partial ordered sets. This construction is based on the transitive closure equivalence relation on \( \text{RAS} \); the equivalent classes are indexed by POSets. We prove that the operad \( \text{POSet}^\leq \) is isomorphic to a quotient of \( \mathcal{T} \) which is compatible in a natural way with the action on languages.

5.1 From \( \mathcal{T} \) to \( \text{RAS}^\leq \)

Since multitildes are defined by a set of couples \((x, y)\) such that \(x \leq y\), they can be seen as antisymmetric relations compatible with the natural order on \( \mathbb{N} \), i.e. \((x, y) \in T\) implies \(x \leq y\).

On the other hand some multitildes expressions are equivalent. For instance, \( \sim_{\tilde{a} \tilde{b} \tilde{c} \tilde{d}} \) and \( \sim_{\tilde{a} \tilde{b} \tilde{c} \tilde{d}} \) denote the same language : \( \{ e, ab, cd, abcd \} \). This phenomenon is not very natural when stated in terms of relations. Nevertheless, up to a slight transformation, this can seen as a transitive closure. Let us be more precise and define a graduated bijection \( \phi \) between \( \mathcal{T}_k \) and \( \text{RAS}^\leq_k := \text{RAS}^\leq \cap \{1, \ldots, k+1\}^2 \) for any integer \(k\):

\[ \phi : \begin{cases} \mathcal{T}_k \longrightarrow \text{RAS}^\leq_k \\ T \longrightarrow \{(x, y + 1) \mid (x, y) \in T\} \cup \{(x, x) \mid x \in [1, n]\} \end{cases} \]

The fact that \( \phi \) is a bijection is obvious and the inverse bijection \( \phi^{-1} \) is given by:

\[ \phi^{-1} : \begin{cases} \text{RAS}^\leq_k \longrightarrow \mathcal{T}_k \\ R \longrightarrow \{(x, y) \mid (x, y + 1) \in R \land x \neq y\} \end{cases} \]

We endow the set \( \text{RAS}^\leq \) with a structure of operad, setting:

\[ R_1 \circ_j R_2 = \phi(\phi^{-1}(R_1) \circ_i \phi^{-1}(R_2)). \]

Moreover, the bijection \( \phi \) is in fact an isomorphism from the operad \( (\mathcal{T}, \circ) \) to \( (\text{RAS}^\leq, \circ') \).

We define the operator \( n^k \) acting on pairs of integers \((x, y)\) by

\[ n^k \diamond (x, y) = \begin{cases} (x, y) & \text{if } y \leq k \\ (x, y + n - 1) & \text{if } x \leq k < y \\ (x + n - 1, y + n - 1) & \text{otherwise.} \end{cases} \]
We extend the definition to the set of pairs by \( n^k \rightarrow (S) = \{ n^k \rightarrow (x, y) : (x, y) \in S \} \). Note \( n^k \rightarrow \) acts by relabelling \( x \rightarrow x + n - 1 \) when \( x > k \). Hence, we set as for the multitildes:

\[
R_1 \triangledown_k R_2 := n^k \rightarrow (R_1) \cup \mathbin{\preccurlyeq}(R_2),
\]

for \( R_1 \in \mathrm{RAS}^\leq_m \), \( R_2 \in \mathrm{RAS}^\leq_n \) and \( k \leq m \).

The following lemma shows that \( \mathrm{RAS}^\leq \) is closed under the action of the compositions \( \triangledown_k \) and that these binary operators are compatible with the graduation.

**Lemma 6.** Let \( R_1 \in \mathrm{RAS}^\leq_m \) and \( R_2 \in \mathrm{RAS}^\leq_n \) then \( R_1 \triangledown_k R_2 \in RAS_{m+n-1}^\leq \).

**Proof.** The antisymmetry is straightforward from the definition of \( \triangledown_k \). So it suffices to prove the reflexivity, that is to check \((x, x) \in R_1 \triangledown_k R_2 \) for each \( 1 \leq x \leq m + n \). But, from the definitions, \((x, x) \in \triangledown_k(R_1) \) when \( 1 \leq k \) or \( n + k \leq x \leq n + m \) and \((x, x) \in \mathbin{\preccurlyeq}(R_2) \) is \( k \leq x \leq n + k \). So for each \( 1 \leq x \leq m + n \) we obtain \((x, x) \in R_1 \triangledown_k R_2 \). Hence, \( R_1 \triangledown_k R_2 \in \mathrm{RAS}^\leq_{m+n-1} \).

From the partial composition \( \triangledown_k \) we define

\[
\triangledown : \{ \mathrm{RAS}^\leq_m \otimes \mathrm{RAS}^\leq_n \otimes \cdots \otimes \mathrm{RAS}^\leq_{n_m} \rightarrow R \triangleleft (R_1, \ldots, R_m) \rightarrow \mathrm{RAS}^\leq_{n_1 + \cdots + n_m} \} = (\cdots ((R \triangleleft R_1) \triangledown (R_{m-1} \triangledown R_{m-1}) \cdots) \triangledown R_1.
\]

And more precisely:

**Proposition 7.** We have \( \triangledown' \triangleleft \triangledown \) and \( \triangledown' = \triangledown \).

**Proof.** We have \( \phi \triangledown n^k = n^k \rightarrow \phi \) and \( \phi \mathbin{\preccurlyeq} n^k = k \rightarrow \phi \). So

\[
\phi(T_1 \triangledown k T_2) = \phi(T_1) \triangledown k \phi(T_2),
\]

for each \( T_1 \in T_m, T_2 \in T_n \) and \( 1 \leq k \leq m \). Since \( \phi : (T, \triangledown) \rightarrow (\mathrm{RAS}^\leq, \triangledown') \) is an isomorphism of operads, we prove the result. \( \square \)

It follows that:

**Theorem 3.** The structure \( (\mathrm{RAS}^\leq, \triangledown) \) defines an operad isomorphic to \( (T, \triangledown) \).

### 5.2 From \( \mathrm{RAS}^\leq \) to \( \mathrm{POSet}^\leq \)

Any reflexive and antisymmetric relation \( R \) can be turned into a partial order when applying the transitive closure denoted by \( \gamma(R) \).

**Note,** since \( n^k \rightarrow \) and \( \mathbin{\preccurlyeq} \) are just relabelling, they commute with the operator \( \gamma \):

\[
\gamma \triangledown' = \triangledown' \gamma, \quad \mathbin{\preccurlyeq} \gamma = \gamma \mathbin{\preccurlyeq}.
\]

We need the following lemma

**Lemma 7.**

\[
\gamma(R_1 \triangledown k R_2) = \gamma(R_1 \triangledown R_2)
\]

**Proof.** It suffices to prove that \( \gamma R_1 \triangledown_k \gamma R_2 \subseteq \gamma(R_1 \triangledown_k R_2) \). Indeed by transitivity, this implies \( \gamma(R_1 \triangledown_k \gamma R_2) \subseteq \gamma(R_1 \triangledown_k R_2) \) and the reverse inclusion is obvious.

We have \( \gamma R_1 \triangledown_k \gamma R_2 = \triangledown k (\gamma R_1) \cup \mathbin{\preccurlyeq}(\gamma R_2) \). From eq (2) we obtain \( \gamma R_1 \triangledown_k \gamma R_2 = (\gamma \triangledown k (R_1)) \cup (\gamma \mathbin{\preccurlyeq} R_2) \subseteq \gamma(R_1 \triangledown_k R_2) \), as expected. \( \square \)
As an immediate consequence, the transitive closure is compatible with the structure of operads that is:

**Lemma 8.** If $\gamma(R_1) = \gamma(R_1')$ and $\gamma(R_2) = \gamma(R_2')$ then $\gamma(R_1 \diamond_k R_2) = \gamma(R_1' \diamond_k R_2')$.

Denote by $\equiv_\gamma$ the equivalence relation on $\text{RAS}^\leq$ defined by $R_1 \equiv_\gamma R_2$ if and only if $\gamma(R_1) = \gamma(R_2)$. The quotient $\text{RAS}^\leq / \equiv_\gamma$ is automatically endowed with a structure of operad whose composition is deduced from $\gamma$; we denote by $\diamond$ the associated composition. The equivalence classes are indexed by the element $\text{POSet}^\leq := \gamma(\text{RAS}^\leq)$. Let us set $\bowtie_k := \gamma \bowtie_k$ and $\bowtie := \gamma \bowtie$.

Immediately we obtain:

**Proposition 8.** The structure $(\text{POSet}^\leq, \bowtie)$ is an operad isomorphic to $(\text{RAS}^\leq / \equiv_\gamma, \diamond')$.

### 5.3 Pseudotransitive tildes

A multitilde is said *pseudotransitive* if it is the image of a POSet by $\phi^{-1}$. The set of pseudotransitive multitildes will be denoted by

$$\text{PTT} := \phi^{-1}(\text{POSet}^\leq).$$

This set is endowed with an operad structure induced by the partial products

$$T_1 \bowtie_k T_2 := \phi^{-1}(\phi(T_1) \bowtie_k \phi(T_2)) = \phi^{-1} \gamma \phi(T_1 \bowtie_k T_2).$$

For simplicity, set $\hat{\gamma} := \phi^{-1} \gamma \phi$. The equivalence relation $\equiv$ on $\mathcal{T}$ defined by $T_1 \equiv T_2$ if and only if $\hat{\gamma}(T_1) = \hat{\gamma}(T_2)$ is compatible with the operad structure. Indeed,

$$\hat{\gamma}(T_1 \bowtie_k T_2) = \phi^{-1} \gamma \phi(T_1 \bowtie_k T_2) = \phi^{-1} \gamma(\phi(T_1) \bowtie_k \phi(T_2)).$$

But, $\hat{\gamma}(T) = \hat{\gamma}(T')$ implies $\phi(T) \equiv_\gamma \phi(T')$. Since $\equiv_\gamma$ is compatible with the composition in $\text{RAS}^\leq$ we obtain

$$\hat{\gamma}(T_1 \bowtie_k T_2) = \phi^{-1} \gamma(\phi(T_1') \bowtie_k \phi(T_2')) = \hat{\gamma}(T_1' \bowtie_k T_2').$$

Hence, the quotient $\mathcal{T} / \equiv_\gamma$ is isomorphic to $(\text{PTT}, \bowtie)$.

We explicitly describe $\hat{\gamma}$.

**Lemma 9.** The set $\hat{\gamma}(T)$ is the smallest set $S$ such that the two following assertions are satisfied:

1. $T \subset S$
2. $(i, k, (k + 1, j)) \in S$ implies $(i, j) \in S$.

**Proof.** Since $\gamma(\phi(T))$ is the transitive closure of $\phi(T)$, it is the smallest set $R$ such that

1. $\phi(T) \in R$
2. $(i, k, (k, j)) \in R$ implies $(i, j) \in R$.

Hence, $\phi^{-1}(\gamma(\phi(T)))$ is such that if $(i, k, (k + 1, j)) \in \phi^{-1}(\gamma(\phi(T)))$ implies $(i, j) \in \phi^{-1}(\gamma(\phi(T)))$. Indeed let $(i, k, (k + 1, j)) \in \hat{\gamma}(T)$ then $(i, k + 1, (k + 1, j + 1)) \in \gamma(\phi(T))$. Since $\gamma(\phi(T))$ is transitive, we also have $(i, j + 1) \in \gamma(\phi(T))$. The image of this pair is $\phi^{-1}(i, j + 1) = (i, j) \in \hat{\gamma}(T)$. Let $S$ be a subset of $\phi^{-1}(\gamma(\phi(T)))$ verifying

1. $T \subset S$
2. $(i, k, (k + 1, j)) \in S$ implies $(i, j) \in S$. 

Suppose that \((i, j) \notin \phi^{-1}(\gamma(\phi(T)))\), since \(\phi\) is a bijection we obtain \(\phi(i, j) \notin \gamma(\phi(T))\). This contradicts the minimality of \(\gamma(\phi(T))\) and prove the result. \(\square\)

The set \(\bar{\gamma}(T)\) will be referred as the pseudotransitive closure of \(T\).

**Lemma 10.** For each tilde \(T \in \mathcal{T}\), \(V(T) = V(\bar{\gamma}(T))\) holds.

**Proof.** Suppose that \((i, k), (k+1, j) \in T\). Let us prove \(V(T \cup \{(i, j)\}) = V(T)\). Suppose \((i, j) \notin T\) (otherwise the result is obvious). We compare the set \(E := \{v(S) : S \in \mathcal{T}(T)\}\) and \(F := \{v(S) : S \in \mathcal{T}(T \cup \{(i, j)\})\}\). Obviously, \(F(T) \subset \mathcal{T}(T \cup \{(i, j)\})\) hence \(E \subset F\). Let us prove the reverse inclusion and let \(S \subset \mathcal{T}(T \cup \{(i, j)\})\). If \((i, j) \notin S\) then \(S \subset \mathcal{T}(T)\) and then \(v(S) \in E\). Now suppose \((i, j) \in S\) and set \(S' = S \cup \{(i, k), (k+1, j) \setminus \{(i, j)\}\}\). Obviously \(S' \subset \mathcal{T}(T)\). Set \(v(S) = (s_1, \ldots, s_n)\) and \(v(S') = (s'_1, \ldots, s'_{n+1})\). We have \(s_l = s'_l\) if \(l \notin [i, j]\) and 0 otherwise. But \(s'_{i+j} = 0\) when \(l \in [i, k] \cup [k+1, j] = [i, j]\). So \(v(S) = v(S') \in E\). \(\square\)

As a consequence, we define an action of \(\text{POSet}^\Sigma\) and \(\text{PTT}\) on languages:

**Theorem 4.** The sets \(2^{\Sigma^*}\) and \(\text{Reg}^*(\Sigma')\) are \(\text{POSet}^\Sigma\)-module and \(\text{PTT}\)-module.

**Proof.** From Corollary [1] \(2^{\Sigma^*}\) and \(\text{Reg}(\Sigma')\) are \(\mathcal{T}\)-module.

Note if \(T_1 \equiv T_2\) are two tildes of \(\mathcal{T}\) then, by Lemma [10] \(T_1\) and \(T_2\) have the same action on \(2^{\Sigma^*}\) and \(\text{Reg}(\Sigma')\). These set are \(\text{PTT}\)-modules. Since \(\text{PTT}\) is isomorphic to \(\text{POSet}^\Sigma\) as an operad, they are also \(\text{POSet}^\Sigma\)-modules. \(\square\)

### 6 Consequences and perspectives

#### 6.1 Enumeration

The purpose of this section is to compute an upper bound of the number \(n(L_1, \ldots, L_k)\) of languages that can be obtained by the action of tilde on the \(k\)-tuple of languages \((L_1, \ldots, L_k)\).

First, Lemma [11] implies that \(n(L_1, \ldots, L_k)\) equals to the number of languages obtained by applying a \(\text{PTT}\) on \((L_1, \ldots, L_k)\).

From the previous section, the number of \(\text{PTT}\) with arity \(k\) equals the numbers \(p_k\) of \(\text{POSet}^\Sigma\) on \([1, \ldots, k+1]\) that are contained in the usual linear order on integers. So,

\[
\text{n}(L_1, \ldots, L_k) \leq p_k. \tag{10}
\]

The sequence of \(p_k\) has no known closed form but the first values can be found on [18] (A006455); the first values are

\[
2, 7, 40, 357, 4824, 96428, 2800472, 116473461, 6855780268, 656505147444, 64824245807684, \ldots
\]

Note this is also the number of \((k+1) \times (k+1)\) upper triangular idempotent boolean matrices with all diagonal entries 1.

Let us show that the bound is reached for \(L_1 = \{a_1\}, \ldots, L_k = \{a_k\}\) where \(a_1, \ldots, a_k\) are distinct letters.

Let \(P_1 \neq P_2\) be two \(\text{PTT}\)s with arity \(k\) and \((i, j) \in P_1 \setminus P_2\). We have \([(i, j)] \in \mathcal{T}(P_1)\) and then \((1^{i-1}, 0^{j-i}, 1^{n-j+1}) \in V(P_1)\). Suppose \((1^{i-1}, 0^{j-i}, 1^{n-j+1}) \in V(P_2)\). This implies that there exists a set \([(i, k_1), (k_1 + 1, k_2), \ldots, (k_m + 1, j)] \in \mathcal{T}(P_2)\) for some \(m\). So \([(i, k_1), (k_1 + 1, k_2), \ldots, (k_m + 1, j)] \subset P_2\) and from Lemma [9] we obtain \((i, j) \in P_2\). This contradicts our hypothesis. Then,

**Corollary 2.** Let \(a_1, \ldots, a_k\) be \(k\) distinct letters. We have

\[
n(\{a_1\}, \ldots, \{a_k\}) = p_k.
\]
Example 3. There are 7 languages obtained by applying PTT on a pair of letters ([a], [b]):

| tildes  | languages |
|---------|-----------|
| ∅       | ab        |
| (1, 1)  | ab, b     |
| (1, 1), (1, 2) | ab, b, ε |
| (2, 2)  | ab, a     |
| (2, 2), (1, 2) | ab, a, ε |
| (1, 1), (2, 2), (1, 2) | ab, a, b, ε |
| (1, 2)  | ab, ε     |

6.2 Finite languages

Every finite languages can be generated by a multitilde acting on a sequences of elements in \(\Sigma_0 := \{[a]\}_{a \in \Sigma} \cup \{\emptyset\}\).

Proposition 9. For any finite language \(L\) over an alphabet \(\Sigma\), it exists a multitilde \(T \in T_k\) and a \(k\)-tuples \((L_1, \ldots, L_k)\) for some integer \(k\), with \(L_1, \ldots, L_k \in \Sigma_0\) such that

\[ L = T(L_1, \ldots, L_k). \]

Proof. Note first, the set of finite languages is a \(T\)-module, since it is contained in \(2^\Sigma\), which by Corollary [1] is a \(T\)-module, and is closed by the actions of \(T\). Straightforwardly, we have \(\emptyset = \emptyset(\emptyset)\), \(\varepsilon = \emptyset(1, 1)(\emptyset)\) and \([a] = \emptyset(\{a\})\) for any \(a \in \Sigma\).

Hence, since \(T\) is an operad (Theorem [1]), it suffices to prove that the union and the catenation of two languages can be obtained by using a multitilde. We verify easily that

\[ L_1L_2 = \emptyset_2(L_1, L_2) \quad \text{and} \quad L_1 \cup L_2 = \{(1, 2), (2, 3)\}(L_1, \emptyset, L_2). \]

This ends the proof. \(\square\)

Let us give some examples.

Example 4. The set of all subwords of a given word \(a_1 \cdots a_k\), is easily expressed as a multitilde by

\[ \{(i, i) : i \in [1, k]\} (a_1, \ldots, a_k). \]

Note the corresponding POSet is the natural order relation on \([1, k + 1]\).

Example 5. The set of all the prefixes (resp. suffixes) of a given word \(a_1 \cdots a_k\) admits a nice expression in terms of multitildes:

\[ P_k := \{(i, k) : i \in [1, k]\} (a_1, \ldots, a_k) \quad \text{(resp.} S_k := \{(1, i) : i \in [1, k]\} (a_1, \ldots, a_k)) . \]

The corresponding POSets are

```
  k
   ↓
  k+1
    ↓
  2
   ↓
  1
```

```
  k+1
   ↑
  1
   ↑
  3
   ↑
  2
```
Example 6. As a consequence, the set of all the factors of the word \(a_1 \ldots a_k\) is obtained by the action of \(F_k := P_k \cup S_k\). Graphically, the corresponding POSet is:

![Diagram](image)

For instance, consider the multitilde

\[
F_3 := \{(1, 1), (1, 2), (1, 3), (2, 3), (3, 3)\}
\]

its free subsets are

\[
\{\emptyset\} \cup \{\{c\} : c \in F_3\} \cup \{(1, 1), (2, 3), (3, 3), (1, 3), (3, 3)\}
\]

The corresponding boolean vectors are

\[
(1, 1, 1), (0, 1, 1), (0, 0, 1), (0, 0, 0), (1, 0, 0), (1, 1, 0), (0, 1, 0).
\]

When acting on \((a, b, c)\) they respectively give \(a_{bc}, \{bc\}, \{c\}, \{a\}, \{ab\}\) and \(\{b\}\).

Example 7. More generally, we have the following property:

If \(L = T(L_1, \ldots, L_k)\) with \(L_i \in \{a : a \in \Sigma\} \cup \{\emptyset\}\) we have

1. The set of the prefixes of any words in \(L\), \(\text{Pref}(L)\) is a subset of \((T \cup P_k)(L_1, \ldots, L_k)\)
2. The set of the suffixes of any words in \(L\), \(\text{Suff}(L)\) is a subset of \((T \cup S_k)(L_1, \ldots, L_k)\).
3. The set of the factors of any words in \(L\), \(\text{Fact}(L)\) is a subset of \((T \cup F_k)(L_1, \ldots, L_k)\).

One has only to show the first assertion. Indeed, the second point is obtained by symmetry from the first and the third is simply the composition of the first and the second.

The free subsets of \(T \cup P_k\) are

\[
\mathcal{F}(T \cup P_k) = \mathcal{F}(T) \cup \{S' = S \cup \{(i, k) : i \in [1, k], S \in \mathcal{F}(T), S' \text{ is free}\}
\]

Consider a proper prefix \(p\) of word \(w = ps \in L\). Let \(I \subset [1, k]\) such that \(i \in I\) if and only if \(L_i \neq \emptyset\). Without loss of generality, we suppose that all the \(L_i\) for \(i \in I\) are distinct. Set \(s = a_{s'}\) with \(a_i \in \Sigma\) and let \(j\) be the unique integer such that \(L_j = \{a_j\}\). Since \(w \in L\), it exists a free list \(S\) such that \(\nu(S)(L_1, \ldots, L_k) = [w]\). Furthermore, \(S' = \{(j_1, j_2) : (j_1, j_2) \in S, \llbracket j_1, j_2 \rrbracket \subset [1, j] \} \subset \mathcal{F}(T)\). Indeed, there is no pair \((j_1, j_2)\) in \(S\) such that \(j \in \llbracket j_1, j_2 \rrbracket\) (otherwise the letter \(a_j\) does not appear in \(w\)). Hence, \(S\) is composed only by pairs \((j_1, j_2)\) in \(T\) such that \(j_2 < j\) or \(j_1 < j_1\). It follows that \(S' = \{(j_1, j_2) : j_2 < j_1\}\) is a free list of \(T\) because \(S\) is free. So \(S' \cup (j + 1, k) \subset T \cup P_k\) is free and \(\nu(S')((j + 1, k))(L_1, \ldots, L_k) = \nu(S')(L_1, \ldots, L_k) = p\). It follows that \(p \in (T \cup P)(L_1, \ldots, L_k)\) and our claim.

The inclusions are strict in the general case. For instance, consider \(0_4([a], [b], [c], \emptyset) = \emptyset\) and \((0_4 \cup P_4)([a], [b], [c], \emptyset) = [\epsilon, a, ab, abc]\).

Note if each \(L_i \neq \emptyset\) the inclusions are equalities. Although this is a sufficient condition, it is not necessary as shown by \((1, 3), (3, 5))((a), [b], \emptyset, [c], [d]) = (\epsilon, [a], [ab], [c], [cd]) = \text{Pref}(ab, cd)).

19
6.3 Regular languages

In the previous section, we showed that each finite language can be obtained by the action of a multiltile on a k-tuple in $\Sigma_k^k$. As a consequence, any regular languages can be written as a combination of multiltiles and stars acting on a k-tuple in $\Sigma_k^k$.

Consider the smallest operad $T^*$ containing $T$ together with a new formal 1-ary operator denoted $\star$. The elements of $T^*$ are trees whose nodes are multiltiles or $\star$ and are such that the roots of the subtrees immediately issued of a multiltile are not multiltiles (indeed, the composition $\circ$ of two multiltiles gives another multiltile).

To define properly an action of $T^*$ on languages, it suffices to define the action of the operator $\star$ which is naturally $\star(L) = L'$. Note $\star(\star(L)) = (L')^* = L' = \star(L)$. To simplify the operators we introduce the smallest equivalence relation $\equiv$ compatible with the composition such that $\star \circ \star \equiv \star$, and the operad $T^* = T^*/\equiv$. Straightforwardly, the following result holds:

**Proposition 10.**

1. The sets $2^{\Sigma^k}$ and $\text{Reg}(\Sigma^k)$ are both $T^*$ and $T^*$-modules.
2. Each regular language can be obtained by the action of an element in $T^*$ on a k-tuple in $\Sigma_k^k$.

Similarly to the operad $T$, there is a very interesting underlying combinatoric structure to the operad $T^*$ involving pairs of relations and a generalization of the dissections of polygons. This study is deferred to a forthcoming paper.

7 Conclusion

In this paper, we introduced several operads which allow us to represent multiltile operations defined by one of the authors in [3]. Each of these operads plays a specific role in the understanding of these tools. For instance, the operad $T$ encodes the structure of composition of the multiltile operators. The operad $B$ explains how the multiltiles act on languages. Finally, the operads $\text{RAS}^\leq$ and $\text{POSet}^\leq$ establish a relation between these objects and well-known combinatorial structures. This formalizes the equivalence of two multiltiles and enumerates the different inequivalent multiltiles.

This study is a first step towards the construction of a new combinatorial approach to the theory of languages. First, regular languages can be seen as the result of the action of an element of $T^*$ on a k-tuple of letters. Second, another action of $T$ on languages have been defined in [3] and combined with the action of multiltiles in [2]. Similar operadic constructions can be defined for these operators. These two points will be studied in forthcoming papers.

The multiltile have been introduced in the aim to enlarge the class of automata which can be represented as an expression using a number of symbols which is a linear function of the number of states. It remains to study the algorithmic improvement which can be brought by the different structures of operads.

References

[1] V. Antimirov. Partial derivatives of regular expressions and finite automaton constructions. *Theoret. Comput. Sci.*, 155:291–319, 1996.

[2] P. Caron, J.-M. Champarnaud, and L. Mignot. Acyclic automata and small expressions using multi-tilde-bar operators. *Theoret. Comput. Sci.*, 411(38–39):3423–3435, 2010.

[3] P. Caron, J.-M. Champarnaud, and L. Mignot. Multi-bar and multi-tilde regular operators. *Journal of Automata, Languages and Combinatorics*, 16(1):11–26, 2011.

[4] P. Caron and D. Ziadi. Characterization of Glushkov automata. *Theoret. Comput. Sci.*, 233(1–2):75–90, 2000.
[5] J.-M. Champarnaud and D. Ziadi. From c-continuations to new quadratic algorithms for automata synthesis. *Internat. J. Algebra Comput.*, 11(6):707–735, 2001.

[6] F. Chapoton. Operads and algebraic combinatorics of trees. *Séminaire Lotharingien de Combinatoire*, B58c:1–27, 2007.

[7] A. Ehrenfeucht and H.-P. Zeiger. Complexity measures for regular expressions. *J. Comput. Syst. Sci.*, 12(2):134–146, 1976.

[8] V.L. Ginzburg and M. Kapranov. Koszul duality for operads. *Duke Math. J.*, 76(1):203–272, 1994.

[9] V. M. Glushkov. The abstract theory of automata. *Russian Mathematical Surveys*, 16:1–53, 1961.

[10] S. Kleene. Representation of events in nerve nets and finite automata. *Automata Studies*, Ann. Math. Studies 34:3–41, 1956. Princeton U. Press.

[11] M. Lazard. Lois de groupes et analyseurs. *Annales scientifiques de l’E.N.S. 3ième série*, 72(4):299–400, 1955.

[12] T. Leinster. *Higher operads, Higher categories*. Cambridge University Press, 2004.

[13] J.-L. Loday and B. Vallette. *Algebraic Operads*. draft available at http://www-irma.u-strasbg.fr/~loday/PAPERS/LodayVallette.pdf, 2010.

[14] J.P. May. Operads, algebras and modules. *Operads Proceedings of Renaissance Conferences Hartford CTLuminy 1995*, 202:1–14, 1971.

[15] J.P. May. *The geometry of iterated loop spaces*. Number 271 in Lecture Notes in Mathematics. Springer-Verlag, 1972.

[16] R. F. McNaughton and H. Yamada. Regular expressions and state graphs for automata. *IEEE Transactions on Electronic Computers*, 9:39–57, March 1960.

[17] L. Mignot. *Des Codes Barres pour les Langages Rationnels*. PhD thesis, LITIS, Université de Rouen, France, 2010. available online, URL: http://ludovicmignot.free.fr.

[18] N. J. A. Sloane. The on-line encyclopedia of integer sequences. URL: http://oeis.org/.