Discrete Mathematics and Theoretical Computer Science DMTCS vol. **12:2**, 2010, 381–400

Combinatorial Route to Algebra: The Art of Composition & Decomposition

Pawel Blasiak†

H. Niewodniczański Institute of Nuclear Physics, Polish Academy of Sciences, Kraków, Poland

received 31st July 2009, *revised 23rd March 2010*, *accepted 10th August 2010*.

We consider a general concept of composition and decomposition of objects, and discuss a few natural properties one may expect from a reasonable choice thereof. It will be demonstrated how this leads to multiplication and co-multiplication laws, thereby providing a generic scheme furnishing combinatorial classes with an algebraic structure. The paper is meant as a gentle introduction to the concepts of composition and decomposition with the emphasis on combinatorial origin of the ensuing algebraic constructions.

Keywords: composition/decomposition of objects, multiplication/co-multiplication, combinatorics, algebra

1 Introduction

A great deal of concrete examples of abstract algebraic structures are based on combinatorial constructions. Their advantage comes from simplicity steaming from the use of intuitive notions of enumeration, composition and decomposition which oftentimes provide insightful interpretations and neat pictorial arguments. In the present paper we are interested in clarifying the concept of composition/decomposition and the development of a general scheme which would furnish combinatorial objects with an algebraic structure.

In recent years the subject of combinatorics has grown into maturity. It gained a solid foundation in terms of combinatorial classes consisting of objects having size, subject to various constructions transforming classes one into another, see *e.g.* [FS09, BLL98, GJ83]. Here, we will augment this framework by considering objects which may *compose* and *decompose* according to some internal law. This idea was pioneered by G.-C. Rota who considered *monoidal* composition rules and introduced the concept of section coefficients showing how they lead to the co-algebra and bi-algebra structures [JR79]. It was further given a firm foundation by A. Joyal [Joy81] on the grounds of the theory of species. In the present paper we discuss this approach from a modern perspective and generalize the concept of composition in order to give a proper account for indeterminate (*non-monoidal*) composition laws according to which two given objects may combine in more than one way. Moreover, we will provide a few natural conditions one might expect from a reasonable composition/decomposition rule, and show how they lead to

†Email: `Pawel.Blasiak@ifj.edu.pl`

algebra, co-algebra, bi-algebra and Hopf algebra structures [Bou89, Swe69, Abe80]. Our treatment has the virtue of a direct scheme translating combinatorial structures into algebraic ones – it only has to be to checked whether the law of composition/decomposition of objects satisfies certain properties. We note that these ideas, however implicit in the construction of some instances of bi-algebras, have never been explicitly exposed in full generality. We will illustrate this framework on three examples of classical combinatorial structures: words, graphs and trees. For words we will provide three different composition/decomposition rules leading to the free algebra, symmetric algebra and shuffle algebra (the latter one with a non-monoidal composition law) [Lot83, Reu93]. In the case of graphs, except of trivial rules leading to the commutative and co-commutative algebra, we will also describe the Connes-Krimer algebra of trees [Kre98, CK98]. One can find many other examples of monoidal composition laws in the seminal paper [JR79]; for instances of non-monoidal rules see *e.g.* [GL89, BDH$^+$10]. A comprehensive survey of a recent development of the subject with an eye on combinatorial methods can be found in [Car07].

The paper is written as a self-contained tutorial on the combinatorial concepts of composition and decomposition explaining how they give rise to algebraic structures. We start in Section 2 by briefly recalling the notions of multiset and combinatorial class. In Section 3 we precise the notion of composition/decomposition and discuss a choice of general conditions which lead to the construction of algebraic structures in Section 4. Finally, in Section 5 we illustrate this general scheme on a few concrete examples.

2 Preliminaries

2.1 Multiset

A basic object of our study is a *multiset*. It differs from a set by allowing multiple copies of elements and formally can be defined as a pair (A, m), where A is a set and $m : A \longrightarrow \mathbb{N}_{\geqslant 1}$ is a function counting multiplicities of elements.[(i)] For example the multiset $\{a, a, b, c, c, c\}$ is described by the underlying set $A = \{a, b, c\}$ and the multiplicity function $m(a) = 2$, $m(b) = 1$ and $m(c) = 3$. Note that each set is a multiset with multiplicities of all elements equal to one. It is a usual practice to drop the multiplicity function m in the denotation of a multiset (A, m) and simply write A as its character should be evident from the context (in the following we will mainly deal with multisets!). Extension of the conventional set-theoretical operations to multisets is straightforward by taking into account copies of elements. Accordingly, the sum of two multisets (A, m_A) and (B, m_B) is the multiset $A \uplus B = (A \cup B, m_{A\cup B})$ where

$$m_{A\cup B}(x) = \begin{cases} m_A(x) + m_B(x)\,, & x \in A \cap B \\ m_A(x)\,, & x \in A - B \\ m_B(x)\,, & x \in B - A \end{cases}\quad, \tag{1}$$

whilst the product is defined as $A \times B = (A \times B, m_{A\times B})$, where $m_{A\times B}((a, b)) = m_A(a) \cdot m_B(b)$. We note that one should be cautious when comparing multisets and not forget that equality involves the coincidence of the underlying sets as well as the multiplicities of the corresponding elements. Similarly, inclusion of multisets $(A, m_A) \subset (B, m_B)$ should be understood as the inclusion of the underlying sets $A \subset B$ with the additional condition $m_A(x) \leqslant m_B(x)$ for $x \in A$.

(i) For equivalent definition of a multiset based on the SEQ construction subject to appropriate equivalence relation see [FS09] p. 26.

2.2 Combinatorial class

In the paper we will be concerned with the concept of *combinatorial class* $\mathcal{C}$ which is a denumerable collection of objects. Usually, it is equipped with the notion of *size* $|\cdot| : \mathcal{C} \longrightarrow \mathbb{N}$ which counts some characteristic carried by objects in the class, *e.g.* the number of elements they are build of. The size function divides $\mathcal{C}$ into disjoint subclasses $\mathcal{C}_n = \{\Gamma \in \mathcal{C} : |\Gamma| = n\}$ composed of objects of size n only. Clearly, we have $\mathcal{C} = \bigcup_{n\in\mathbb{N}}$. A typical problem in combinatorial analysis consists in classifying objects according to the size and counting the number of elements in $\mathcal{C}_n$.

In the sequel we will often use the *multiset construction*. For a given combinatorial class $\mathcal{C}$ it defines a new class $\mathrm{MS}\textsc{et}(\mathcal{C})$ whose objects are multisets of elements taken from $\mathcal{C}$. We note that the size of $\Gamma \in \mathrm{MS}\textsc{et}(\mathcal{C})$ is canonically defined as the sum of sizes of all its elements, *i.e.* $|\Gamma| = \sum_{\gamma\in\Gamma} |\gamma|$.

3 Combinatorial Composition & Decomposition

We will consider a combinatorial class $\mathcal{C}$ consisting of objects which can *compose* and *decompose* within the class. In this section we precisely define both concepts and discuss a few natural conditions one might expect from a reasonable composition/decomposition rule.

3.1 Composition

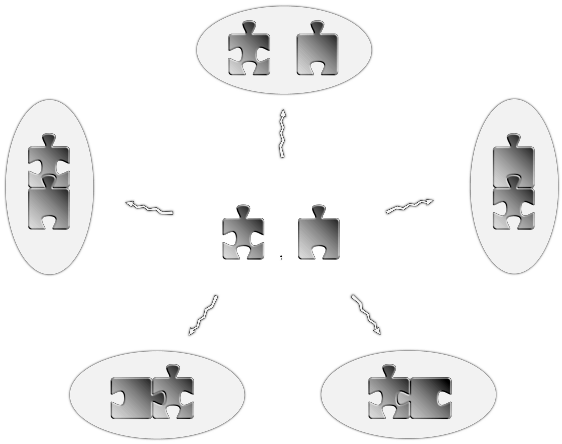


Fig. 1: Illustration of the concept of composition. Two puzzles may compose in six possible ways, two of which give the same result (two at the bottom).

Composition of objects in a combinatorial class $\mathcal{C}$ is a prescription how from two objects make another one in the same class. In general, such a rule may be indeterminate that is allow two given objects to compose in a number of ways. Furthermore, it can happen that some of these various possibilities produce the same outcome. See Fig. 1 for illustration. Therefore, a complete description of composition should keep a record of all the options which is conveniently attained by means of the multiset construction. Here is the formal definition:

Definition 1 (Composition)
For a given combinatorial class $\mathcal{C}$ the composition rule is a mapping

$$\blacktriangleleft : \mathcal{C} \times \mathcal{C} \quad \longrightarrow \quad \text{MSET}(\mathcal{C}) \;, \tag{2}$$

assigning to each pair of objects $\Gamma_2, \Gamma_1 \in \mathcal{C}$ the multiset, denoted by $\Gamma_2 \blacktriangleleft \Gamma_1$, consisting of all possible compositions of Γ_2 with Γ_1, where multiple copies keep an account of the number of ways in which given outcome occurs. Sometimes, for brevity we will write $(\Gamma_2, \Gamma_1) \rightsquigarrow \Gamma \in \Gamma_2 \blacktriangleleft \Gamma_1$.

Note that this definition naturally extends to the mapping $\blacktriangleleft : \text{MSET}(\mathcal{C}) \times \text{MSET}(\mathcal{C}) \longrightarrow \text{MSET}(\mathcal{C})$ which for given two multisets $\Gamma_2, \Gamma_1 \in \text{MSET}(\mathcal{C})$ take their elements one by one, compose and collect the results all together, *i.e.*

$$\Gamma_2 \blacktriangleleft \Gamma_1 = \biguplus_{\gamma_2 \in \Gamma_2, \gamma_1 \in \Gamma_1} \gamma_2 \blacktriangleleft \gamma_1 \;. \tag{3}$$

At this point the concept of composition is quite general, and its further development obviously depends on the choice of the rule. One supplements this construction with additional constraints. Below we discuss some natural conditions one might expect from a reasonable composition rule.

(C1) ***Finiteness.*** It is sensible to assume that objects compose only in a finite number of ways, *i.e.* for each $\Gamma_2, \Gamma_1 \in \mathcal{C}$ we have

$$\# \, \Gamma_2 \blacktriangleleft \Gamma_1 < \infty \;. \tag{4}$$

(C2) ***Triple composition.*** Composition applies to more that two objects as well. For given $\Gamma_3, \Gamma_2, \Gamma_1 \in \mathcal{C}$ one can compose them successively and construct the multiset of possible compositions. There are two possible scenarios however: one can either start by composing the first two $(\Gamma_2, \Gamma_1) \rightsquigarrow \Gamma'$ and then composing the outcome with the third $(\Gamma_3, \Gamma') \rightsquigarrow \Gamma$, or change the order and begin with $(\Gamma_3, \Gamma_2) \rightsquigarrow \Gamma''$ followed by the composition with the first $(\Gamma'', \Gamma_1) \rightsquigarrow \Gamma$. It is plausible to require that both scenarios lead to the same multiset. This condition is a sort of associativity property which in a compact form reads

$$\Gamma_3 \blacktriangleleft (\Gamma_2 \blacktriangleleft \Gamma_1) = (\Gamma_3 \blacktriangleleft \Gamma_2) \blacktriangleleft \Gamma_1 \;. \tag{5}$$

Note that it justifies dropping of the brackets in the denotation of triple composition $\Gamma_3 \blacktriangleleft \Gamma_2 \blacktriangleleft \Gamma_1$. Clearly, the procedure generalizes to multiple compositions and Eq. (5) entails analogous condition in this case as well.

(C3) ***Neutral object.*** Often, in a class there exists a neutral object, denoted by Ø, which composes with elements of the class only in a trivial way, *i.e.* $(\text{Ø}, \Gamma) \rightsquigarrow \Gamma$ and $(\Gamma, \text{Ø}) \rightsquigarrow \Gamma$. In other words, for each $\Gamma \in \mathcal{C}$ we have

$$\text{Ø} \blacktriangleleft \Gamma = \{\Gamma\} \quad \& \quad \Gamma \blacktriangleleft \text{Ø} = \{\Gamma\} \;. \tag{6}$$

Note that if Ø exists, it is unique.

(C4) ***Symmetry.*** Sometimes the composition rule is such that the order in which elements are composed is irrelevant. Then for each $\Gamma_2, \Gamma_1 \in \mathcal{C}$ the following commutativity condition holds

$$\Gamma_2 \blacktriangleleft \Gamma_1 = \Gamma_1 \blacktriangleleft \Gamma_2 \;. \tag{7}$$

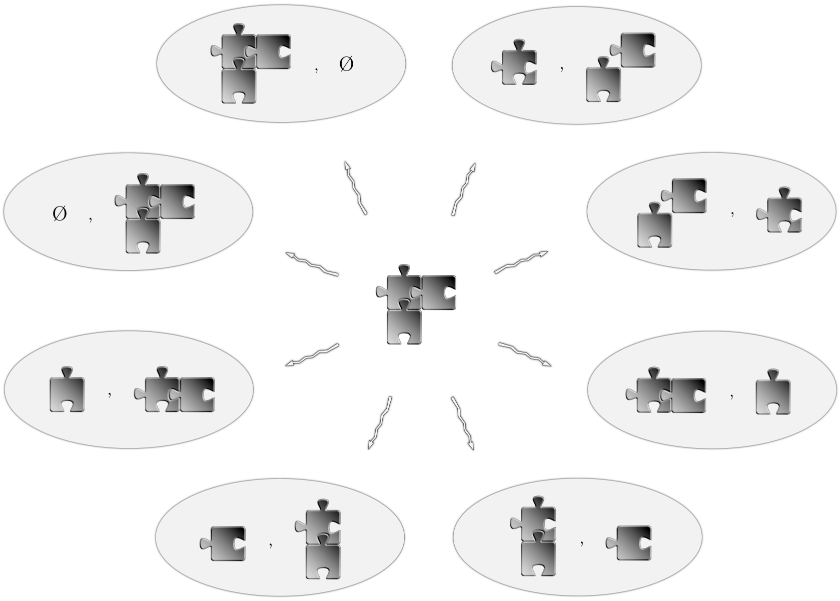


Fig. 2: Illustration of the concept of decomposition. A compound puzzle decomposes into ordered pairs of pieces in eight possible ways. Note that two outcomes occur twice: the two at the bottom-left corner and the two at the bottom-right are the same.

3.2 Decomposition

Suppose that a combinatorial class $\mathcal{C}$ allows for *decomposition* of objects, *i.e.* splitting into ordered pairs of pieces from the same class. In general, there might be various ways of splitting an object following a given rule and, moreover, some of them may yield the same result. See Fig. 2 for illustration. The whole collection of possibilities is again properly described by the notion of multiset. Hence, we have the definition

Definition 2 (Decomposition)
Decomposition rule in a combinatorial class $\mathcal{C}$ is a mapping

$$\langle\,\cdot\,\rangle : \mathcal{C} \quad\longrightarrow\quad \mathrm{MS{\small ET}}\,(\mathcal{C}\times\mathcal{C})\ , \tag{8}$$

which for each object $\Gamma\in\mathcal{C}$ defines the multiset, denoted by $\langle\Gamma\rangle$, comprised of all pairs (Γ'',Γ') which are splittings of Γ, with multiple copies keeping a record of possible decompositions producing the same outcome. Concisely, we will write $\Gamma \rightsquigarrow (\Gamma'',\Gamma')\in\langle\Gamma\rangle$.

Extension of the definition to the mapping $\langle\,\cdot\,\rangle : \mathrm{MS{\small ET}}\,(\mathcal{C}) \longrightarrow \mathrm{MS{\small ET}}\,(\mathcal{C}\times\mathcal{C})$ is straightforwardly given by collecting all together decompositions of elements taken from $\Gamma\in\mathrm{MS{\small ET}}(\mathcal{C})$, *i.e.*

$$\langle\Gamma\rangle = \biguplus_{\gamma\in\Gamma}\ \langle\gamma\rangle\ . \tag{9}$$

Below, analogously as in Section 3.1 we consider some general conditions which one might require from a reasonable decomposition rule. We observe that most of them are in a sense dual to those discussed for the composition rule, which reflects the opposite character of both procedures. Note, however, that the decomposition rule is so far unrelated to composition – insofar as the latter might be even not defined – and the conditions should be treated as independent.

(D1) ***Finiteness.*** One may reasonably expect that objects decompose in a finite number of ways only, *i.e.* for each $\Gamma \in \mathcal{C}$ we require

$$\# \langle \Gamma \rangle < \infty \,. \tag{10}$$

(D2) ***Triple decomposition.*** Decomposition into pairs naturally extends to splitting an object into three pieces $\Gamma \rightsquigarrow (\Gamma_3, \Gamma_2, \Gamma_1)$. An obvious way to carry out the multiple splitting is by applying the same procedure repeatedly, *i.e.* decomposing one of the components obtained in the preceding step. Following this prescription one usually expects that the result does not depend on the choice of the component it is applied to. In other words, we require that we end up with the same collection of triple decompositions when splitting $\Gamma \rightsquigarrow (\Gamma'', \Gamma_1)$ and then splitting the left component $\Gamma'' \rightsquigarrow (\Gamma_3, \Gamma_2)$, as in the case when starting with $\Gamma \rightsquigarrow (\Gamma_3, \Gamma')$ and then splitting the right component $\Gamma' \rightsquigarrow (\Gamma_2, \Gamma_1)$. This condition can be seen as a sort of co-associativity property for decomposition, and in explicit form boils down to the following equality between multisets of triple decompositions

$$\biguplus_{(\Gamma'', \Gamma') \in \langle \Gamma \rangle} \{\Gamma''\} \times \langle \Gamma' \rangle = \biguplus_{(\Gamma'', \Gamma') \in \langle \Gamma \rangle} \langle \Gamma'' \rangle \times \{\Gamma'\} \,. \tag{11}$$

The above procedure directly extends to splitting into multiple pieces $\Gamma \rightsquigarrow (\Gamma_n, ...\Gamma_1)$ by iterated decomposition. Clearly, the condition of Eq. (11) asserts the same result no matter which way decompositions are carried out. Hence, we can consistently define the multiset consisting of multiple decompositions of an object as

$$\langle \Gamma \rangle^{(n)} = \biguplus_{\Gamma \rightsquigarrow (\Gamma_n, ...\Gamma_1)} \{(\Gamma_n, ..., \Gamma_1)\} \,, \tag{12}$$

with the convention $\langle \Gamma \rangle^{(1)} = \langle \Gamma \rangle$.

(D3) ***Void object.*** Oftentimes, a class contains a void (or empty) element Ø, such that objects decompose in a trivial way. It should have the property that any object $\Gamma \neq$ Ø splits into a pair containing either Ø or Γ in exactly two ways

$$\Gamma \rightsquigarrow (\text{Ø}, \Gamma) \qquad \& \qquad \Gamma \rightsquigarrow (\Gamma, \text{Ø}) \,, \tag{13}$$

and Ø $\rightsquigarrow$ (Ø, Ø). Clearly, if Ø exists, it is unique.

(D4) ***Symmetry.*** For some rules the order between components in decompositions is immaterial, *i.e.* it allows for the exchange $(\Gamma', \Gamma'') \longleftrightarrow (\Gamma'', \Gamma')$. In this case we have the following symmetry condition

$$(\Gamma', \Gamma'') \in \langle \Gamma \rangle \iff (\Gamma'', \Gamma') \in \langle \Gamma \rangle \,, \tag{14}$$

and multiplicities of (Γ', Γ'') and (Γ'', Γ') in $\langle \Gamma \rangle$ are the same.

(D5) ***Finiteness of multiple decompositions.*** Recall multiple decompositions $\Gamma \rightsquigarrow (\Gamma_n, ...\Gamma_1)$ considered in condition **(D4)** and observe that we may go with the number of components to any $n \in \mathbb{N}$. However, if one takes into account only nontrivial decompositions, *i.e.* such that do not contain void Ø components, it is often the case that the process terminates after a finite number of steps. In other words, for each $\Gamma \in \mathcal{C}$ there exists $N \in \mathbb{N}$ such that for all $n \geqslant N$ one has

$$\{\Gamma \rightsquigarrow (\Gamma_n, ...\Gamma_1) : \Gamma_n, ..., \Gamma_1 \neq \text{Ø}\} = \emptyset \,. \tag{15}$$

3.3 Compatibility

Now, let us take a combinatorial class $\mathcal{C}$ which admits both composition and decomposition of objects at the same time. We give a simple *compatibility* condition for both procedures to be consistently combined together.

(CD1) ***Composition–decomposition compatibility.*** Suppose we are given a pair of objects $(\Gamma_2, \Gamma_1) \in \mathcal{C} \times \mathcal{C}$ which we want to decompose. We may think of two consistent decomposition schemes which involve composition as an intermediate step. We can either start by composing them together $\Gamma_2 \blacktriangleleft \Gamma_1$ and then splitting all the resulting objects into pieces, or first decompose each of them separately into $\langle\Gamma_2\rangle$ and $\langle\Gamma_1\rangle$ and then compose elements of both multisets in a component-wise manner. One may reasonably expect the same outcome no matter which procedure is applied. The formal description of compatibility comes down to the equality of multisets

$$\langle \Gamma_2 \blacktriangleleft \Gamma_1 \rangle = \biguplus_{\substack{(\Gamma_2'', \Gamma_2') \in \langle \Gamma_2 \rangle \\ (\Gamma_1'', \Gamma_1') \in \langle \Gamma_1 \rangle}} (\Gamma_2'' \blacktriangleleft \Gamma_1'') \times (\Gamma_2' \blacktriangleleft \Gamma_1') \,. \tag{16}$$

We remark that this property implies that the void and neutral object of conditions **(D3)** and **(C3)** are the same, and hence common denotation Ø.

Oftentimes, composition/decomposition of objects come alongside with the notion of size. It is usually the case when their defining rules make use of the same characteristics which are counted by the size function. Here is a useful condition connecting these concepts:

(CD2) ***Compatibility with size.*** It may happen that the considered composition rule preserves size. This means that when composing two objects $(\Gamma_2, \Gamma_1) \rightsquigarrow \Gamma$ sizes of both components add up, *i.e.*

$$|\Gamma| = |\Gamma_2| + |\Gamma_1| \quad \text{for} \quad \Gamma \in \Gamma_2 \blacktriangleleft \Gamma_1 \,. \tag{17}$$

This requirement boils down to the restriction on the mapping of Eq. (2) in the following way:[(ii)]

$$\blacktriangleleft : \mathcal{C}_i \times \mathcal{C}_j \quad \longrightarrow \quad \text{MSET}\,(\mathcal{C}_{i+j}) \,. \tag{18}$$

A parallel condition for decomposition implies that after splitting $\Gamma \rightsquigarrow (\Gamma'', \Gamma')$ the original size of an object distributes between the parts, *i.e.*

$$|\Gamma| = |\Gamma''| + |\Gamma'| \quad \text{for} \quad (\Gamma'', \Gamma') \in \langle \Gamma \rangle \,. \tag{19}$$

[(ii)] Recall, that the size function $|\cdot| : \mathcal{C} \longrightarrow \mathbb{N}$ divides $\mathcal{C}$ into disjoint subclasses $\mathcal{C}_n = \{\Gamma \in \mathcal{C} : |\Gamma| = n\}$, such that $\mathcal{C} = \bigcup_{n \in \mathbb{N}} \mathcal{C}_n$.

This translates into the constraint on the mapping of Eq. (8) as:

$$\langle \cdot \rangle : \mathcal{C}_k \quad \longrightarrow \quad \mathrm{MSET}\Big(\biguplus_{i+j=k} \mathcal{C}_i \times \mathcal{C}_j \Big) \,. \tag{20}$$

In the following, we will assume that there is a single object of size zero, *i.e.* $\mathcal{C}_0 = \{\emptyset\}$.

4 Construction of Algebraic Structures

We will demonstrate the way in which combinatorial objects can be equipped with natural algebraic structures based on the composition/decomposition concept. The key role in the argument play the conditions discussed in Section 3 which provide a route to systematic construction of the algebra, co-algebra, bi-algebra and Hopf algebra structures. We note that most of combinatorial algebras can be systematically treated along these lines.

4.1 Vector space

For a given combinatorial class $\mathcal{C}$ we will consider a vector space $\mathscr{C}$ over a field $\mathbb{K}$ which consists of (finite) linear combinations of elements in $\mathcal{C}$, *i.e.*

$$\mathscr{C} = \mathbb{K}\,\mathcal{C} = \Big\{ \sum\nolimits_i \alpha_i\, \Gamma_i : \; \alpha_i \in \mathbb{K},\; \Gamma_i \in \mathcal{C} \Big\} \,. \tag{21}$$

Addition of elements and multiplication by scalars in $\mathscr{C}$ has the usual form

$$\sum\nolimits_i \alpha_i\, \Gamma_i + \sum\nolimits_i \beta_i\, \Gamma_i \;=\; \sum\nolimits_i (\alpha_i + \beta_i)\, \Gamma_i \,, \tag{22}$$

$$\alpha \sum\nolimits_i \beta_i\, \Gamma_i \;=\; \sum\nolimits_i (\alpha\, \beta_i)\, \Gamma_i \,. \tag{23}$$

Clearly, elements of $\mathcal{C}$ are independent and span the whole vector space. Hence, $\mathscr{C}$ comes endowed with the distinguished basis which, in addition, carries a combinatorial meaning. We will call $\mathcal{C}$ the *combinatorial basis* of $\mathscr{C}$.

4.2 Multiplication & co-multiplication

Having defined the vector space $\mathscr{C}$ built on a combinatorial class $\mathcal{C}$ we are ready to make use of its combinatorial content. Below we provide a general scheme for constructing an algebra and co-algebra structures [Bou89] based on the notions of composition and decomposition discussed in Section 3.

Suppose $\mathcal{C}$ admits *composition* as defined in Section 3.1. We will consider a bilinear mapping

$$* : \mathscr{C} \times \mathscr{C} \longrightarrow \mathscr{C} \tag{24}$$

defined on basis elements $\Gamma_2, \Gamma_1 \in \mathcal{C}$ as the sum of all possible compositions of Γ_2 with Γ_1, *i.e.*

$$\Gamma_2 * \Gamma_1 = \sum_{\Gamma \in \Gamma_2 \blacktriangleleft \Gamma_1} \Gamma \,. \tag{25}$$

Note, that although all coefficients in the defining Eq. (25) are equal to one, some of the terms in the sum may appear several times; this is because $\Gamma_2 \blacktriangleleft \Gamma_1$ is a multiset. One rightly anticipates that multiplicities of elements will play the role of *structure constants* of the algebra. Such defined mapping is a natural candidate for *multiplication* and we have the following statement

Proposition 1 (Algebra)

The vector space $\mathscr{C}$ with the multiplication defined in Eq. (25) forms an associative algebra with unit $(\mathscr{C},+,*,\emptyset)$ if conditions **(C1)** *–* **(C3)** *hold. Under condition* **(C4)** *it is commutative.*

Proof: Condition **(C1)** guarantees that the sum in Eq. (25) is finite – hence it is well defined. Conditions **(C3)** and **(C4)** directly translate into the existence of the unit element Ø and commutativity respectively. Associativity is the consequence of bilinearity of multiplication and condition **(C2)** which asserts equality of multisets resulting from two scenarios of triple composition $(\Gamma_3,\Gamma_2,\Gamma_1) \rightsquigarrow \Gamma$; it is straightforward to check for basis elements that

$$\Gamma_3 * (\Gamma_2 * \Gamma_1) = \sum_{\Gamma \in \Gamma_3 \blacktriangleleft \Gamma_2 \blacktriangleleft \Gamma_1} \Gamma \;\; = (\Gamma_3 * \Gamma_2) * \Gamma_1 \,. \tag{26}$$

□

Now, we will consider $\mathcal{C}$ equipped with the notion of *decomposition* as described in Section 3.2. Let us take a linear mapping

$$\Delta : \mathscr{C} \quad \longrightarrow \quad \mathscr{C} \otimes \mathscr{C} \tag{27}$$

defined on basis elements $\Gamma \in \mathcal{C}$ as the sum of all splittings into pairs, which in explicit form reads

$$\Delta\,(\Gamma) = \sum_{(\Gamma'',\Gamma') \in \langle \Gamma \rangle} \Gamma'' \otimes \Gamma' \,. \tag{28}$$

Repetition of terms in Eq. (28) leads after simplification to coefficients which are multiplicities of elements in the multiset $\langle \Gamma \rangle$. These numbers are sometimes called *section coefficients*, see [JR79]. We will also need a linear mapping

$$\varepsilon : \mathscr{C} \quad \longrightarrow \quad \mathbb{K} \tag{29}$$

which extracts the expansion coefficient standing at the void Ø. It is defined on basis elements $\Gamma \in \mathcal{C}$ in a canonical way

$$\varepsilon(\Gamma) = \begin{cases} 1 & \text{if} \quad \Gamma = \emptyset \,, \\ 0 & \text{otherwise} \,. \end{cases} \tag{30}$$

These mappings play the role of *co-multiplication* and *co-unit* in the construction of a co-algebra as explained in the following proposition

Proposition 2 (Co-algebra)

If conditions **(D1)** *–* **(D3)** *are satisfied the mappings Δ and ε defined in Eqs. (28) and (30) respectively are the co-multiplication and co-unit which make the vector space $\mathscr{C}$ into a co-algebra $(\mathscr{C},+,\Delta,\varepsilon)$. It is co-commutative if condition* **(D4)** *holds.*

Proof: The sum in Eq. (28) is well defined as long as the number of compositions is finite, *i.e.* condition **(D1)** is satisfied. From equivalence of triple splittings $\Gamma \rightsquigarrow (\Gamma_3, \Gamma_2, \Gamma_1)$ obtained in two possible ways considered in condition **(D2)**, one readily verifies for a basis element that

$$(\Delta \otimes Id) \circ \Delta(\Gamma) = \sum_{\Gamma \rightsquigarrow (\Gamma_3, \Gamma_2, \Gamma_1)} \Gamma_3 \otimes \Gamma_2 \otimes \Gamma_1 \ = (Id \otimes \Delta) \circ \Delta(\Gamma)\ , \tag{31}$$

which by linearity extends on all $\mathscr{C}$ proving co-associativity of co-multiplication defined in Eq. (28).
The co-unit $\varepsilon : \mathscr{C} \longrightarrow \mathbb{K}$ by definition should satisfy the equalities

$$(\varepsilon \otimes Id) \circ \Delta = Id = (Id \otimes \varepsilon) \circ \Delta\ , \tag{32}$$

where the identification $\mathbb{K} \otimes \mathscr{C} = \mathscr{C} \otimes \mathbb{K} = \mathscr{C}$ is implied. We will check the first one for a basis element Γ by direct calculation

$$(\varepsilon \otimes Id) \circ \Delta(\Gamma) = \sum_{(\Gamma_1, \Gamma_2) \in \langle \Gamma \rangle} \varepsilon(\Gamma_1) \otimes \Gamma_2 = 1 \otimes \Gamma = \Gamma = Id(\Gamma)\ . \tag{33}$$

Note that we have applied condition **(D3)** by taking all terms in the sum equal to zero except the unique decomposition $(\emptyset, \Gamma)$ picked up by ε in accordance with the definition of Eq. (30). The identification $1 \otimes \Gamma = \Gamma$ completes the proof of the first equality in Eq. (32); verification of the second one is analogous. Check of co-commutativity of the co-product under condition **(D4)** is immediate.

□

4.3 Bi-algebra and Hopf algebra structure

We have seen in Propositions 1 and 2 how the notions of composition and decomposition lead to algebra and co-algebra structure respectively. Both schemes can be combined together so to furnish $\mathscr{C}$ with a bi-algebra structure.

Theorem 1 (Bi-algebra)

If condition **(CD1)** *is satisfied, than the algebra and co-algebra structures in $\mathscr{C}$ are compatible and combine into a bi-algebra $(\mathscr{C}, +, *, \emptyset, \Delta, \varepsilon)$.*

Proof: The structure of a bi-algebra requires that the co-multiplication $\Delta : \mathscr{C} \otimes \mathscr{C} \longrightarrow \mathscr{C}$ and the co-unit $\varepsilon : \mathscr{C} \longrightarrow \mathbb{K}$ of the co-algebra preserve multiplication in $\mathscr{C}$. Thus, we need to verify for basis elements Γ_1 and Γ_2 that

$$\Delta(\Gamma_2 * \Gamma_1) \quad = \quad \Delta(\Gamma_2) * \Delta(\Gamma_1)\ , \tag{34}$$

with component-wise multiplication in the tensor product $\mathcal{G} \otimes \mathcal{G}$ on the right-hand-side, and that

$$\varepsilon(\Gamma_2 * \Gamma_1) \quad = \quad \varepsilon(\Gamma_2)\, \varepsilon(\Gamma_1)\ , \tag{35}$$

with terms on the right-hand-side multiplied in the field $\mathbb{K}$.
We check Eq. (34) directly by expanding both sides using definitions of Eqs. (25) and (28). Accordingly, the left-hand-side takes the form

$$\Delta(\Gamma_2 * \Gamma_1) = \sum_{\Gamma \in \Gamma_2 \blacktriangleleft \Gamma_1} \Delta(\Gamma) = \sum_{\substack{\Gamma \in \Gamma_2 \blacktriangleleft \Gamma_1 \\ (\Gamma'', \Gamma') \in \langle \Gamma \rangle}} \Gamma'' \otimes \Gamma' = \sum_{(\Gamma'', \Gamma') \in \langle \Gamma_2 \blacktriangleleft \Gamma_1 \rangle} \Gamma'' \otimes \Gamma'\ , \tag{36}$$

while the right-hand-side reads

$$\Delta\left(\Gamma_2\right) * \Delta\left(\Gamma_1\right) = \sum_{\substack{(\Gamma_2'',\Gamma_2')\in\langle\Gamma_2\rangle \\ (\Gamma_1'',\Gamma_1')\in\langle\Gamma_1\rangle}} \underbrace{(\Gamma_2''\otimes\Gamma_2') * (\Gamma_1''\otimes\Gamma_1')}_{(\Gamma_2''*\Gamma_1'')\otimes(\Gamma_2'*\Gamma_1')} = \sum_{\substack{(\Gamma_2'',\Gamma_2')\in\langle\Gamma_2\rangle \\ (\Gamma_1'',\Gamma_1')\in\langle\Gamma_1\rangle}} \sum_{\substack{\Gamma''\in\Gamma_2''\blacktriangleleft\Gamma_1'' \\ \Gamma'\in\Gamma_2'\blacktriangleleft\Gamma_1'}} \Gamma''\otimes\Gamma' \,. \tag{37}$$

A closer look at condition **`(CD1)`** and Eq. (16) shows a one-to-one correspondence between terms in the sums on the right-hand-sides of Eqs. (36) and (37), which proves Eq. (34).
Verification of Eq. (35) rests upon a simple observation, steaming from **`(C3)`**, **`(D3)`** and **`(CD1)`**, that composition of objects $\Gamma_2 \blacktriangleleft \Gamma_1$ yields the neutral element Ø only if both of them are void. Then, both sides of Eq. (35) are equal to 1 for $\Gamma_1 = \Gamma_2 = Ø$ and 0 otherwise, which ends the proof.

□

Finally, let us take a linear mapping

$$S : \mathscr{C} \quad\longrightarrow\quad \mathscr{C}\,, \tag{38}$$

defied as an alternating sum of multiple products over possible nontrivial decompositions of an object, *i.e.*

$$S(\Gamma) = \sum_{\substack{\Gamma\rightsquigarrow(\Gamma_n,\ldots,\Gamma_1) \\ \Gamma_n,\ldots,\Gamma_1\neq Ø}} (-1)^n\ \Gamma_n * \ldots * \Gamma_1 \tag{39}$$

for $\Gamma \neq Ø$ and $S(Ø) = Ø$. This mapping provides an *antipode* completing the construction of a Hopf algebra structure on $\mathscr{C}$ [Swe69, Abe80].

Proposition 3 (Hopf Algebra)
If furthermore condition **`(D5)`** *holds, S defined in Eq. (39) is the antipode which makes the bi-algebra $\mathscr{C}$ of Theorem 1 into a Hopf algebra $(\mathscr{C}, +, *, Ø, \Delta, \varepsilon, S)$.*

Proof:
A Hopf algebra consists of a bi-algebra $(\mathscr{C}, +, *, Ø, \Delta, \varepsilon)$ equipped with an antipode $S : \mathscr{C} \longrightarrow \mathscr{C}$. The latter is an endomorphism by definition satisfying the property

$$\mu \circ (Id \otimes S) \circ \Delta = \epsilon = \mu \circ (S \otimes Id) \circ \Delta\,, \tag{40}$$

where for better clarity multiplication was denoted by $\mu(\Gamma_2 \otimes \Gamma_1) = \Gamma_2 * \Gamma_1$. The mapping $\epsilon : \mathscr{C} \longrightarrow \mathscr{C}$, defined as $\epsilon = Ø\,\varepsilon$, is the projection on the subspace spanned by Ø, *i.e.*

$$\epsilon(\Gamma) = \begin{cases} \Gamma & \text{if} \quad \Gamma = Ø\,, \\ 0 & \text{otherwise}\,. \end{cases} \tag{41}$$

We will prove that S given in Eq. (39) satisfies the condition of Eq. (40). We start by considering an auxiliary linear mapping $\Phi : \mathsf{End}(\mathscr{C}) \longrightarrow \mathsf{End}(\mathscr{C})$ defined as

$$\Phi(f) = \mu \circ (Id \otimes f) \circ \Delta\,, \qquad \text{for} \quad f \in \mathsf{End}(\mathscr{C})\,. \tag{42}$$

Observe that under the assumption that Φ is invertible the first equality in Eq. (40) can be rephrased into the condition

$$S = \Phi^{-1}(\epsilon) \; . \tag{43}$$

Now, our objective is to show that Φ is invertible and calculate its inverse explicitly. By extracting identity we get $\Phi = Id + \Phi^+$ and observe that such defined Φ^+ can be written in the form

$$\Phi^+(f) = \mu \circ (\bar{\epsilon} \otimes f) \circ \Delta \; , \qquad \text{for} \quad f \in \mathsf{End}(\mathscr{C}) \; , \tag{44}$$

where $\bar{\epsilon} = Id - \epsilon$ is the complement of ϵ projecting on the subspace spanned by $\Gamma \neq \emptyset$, *i.e.*

$$\bar{\epsilon}(\Gamma) = \begin{cases} 0 & \text{if} \quad \Gamma = \emptyset \; , \\ \Gamma & \text{otherwise} \; . \end{cases} \tag{45}$$

We claim that he mapping Φ is invertible with the inverse given by[(iii)]

$$\Phi^{-1} = \sum_{n=0}^{\infty} (-\Phi^+)^n \; . \tag{46}$$

In order to check that the above sum is well defined one analyzes the sum term by term. It is not difficult to calculate n-th iteration of Φ^+ explicitly

$$\left(\Phi^+\right)^n (f)(\Gamma) = \sum_{\substack{\Gamma \rightsquigarrow (\Gamma_k, \ldots, \Gamma_1, \Gamma_0) \\ \Gamma_k, \ldots, \Gamma_1 \neq \emptyset}} \Gamma_k * \ldots * \Gamma_1 * f(\Gamma_0) \; . \tag{47}$$

We note that in the above formula products of multiple decompositions arise from repeated use of the property of Eq. (34); the exclusion of empty components in the decompositions (except the single one on the right hand side) comes from the definition of $\bar{\epsilon}$ in Eq. (45). The latter constraint together with condition **(D5)** asserts that the number of non-vanishing terms in Eq. (46) is always finite proving that Φ^{-1} is well defined. Finally, using Eqs. (46) and (47) one explicitly calculates S from Eq. (43) obtaining the formula of Eq. (39).

In conclusion, by construction the linear mapping S of Eq. (39) satisfies the first equality in Eq. (40); the second equality can be checked analogously. Therefore we have proved S to be an antipode thus making $\mathscr{C}$ into a Hopf algebra.

□

We remark that by a general theory of Hopf algebras, see [Swe69, Abe80], the property of Eq. (40) implies that S is an anti-morphism and it is unique. Moreover, if $\mathscr{C}$ is *commutative* or *co-commutative* S is an involution, *i.e.* $S \circ S = Id$. We should also observe that the definition of the antipode given in Eq. (39) admits construction by iteration

$$S(\Gamma) = - \sum_{\substack{(\Gamma'', \Gamma') \in \langle \Gamma \rangle \\ \Gamma' \neq \emptyset}} S(\Gamma'') * \Gamma' \; , \tag{48}$$

(iii) For a linear mapping $L = Id + L^+ : V \longrightarrow V$ its inverse can be constructed as $L^{-1} = \sum_{n=0}^{\infty}(-L^+)^n$ provided the sum is well defined. Indeed, one readily checks that $L \circ L^{-1} = (Id + L^+) \circ \sum_{n=0}^{\infty}(-L^+)^n = \sum_{n=0}^{\infty}(-L^+)^n + \sum_{n=0}^{\infty}(-L^+)^{n+1} = Id$, and similarly $L^{-1} \circ L = Id$.

and $S(\emptyset) = \emptyset$.

Finally, whenever composition/decomposition is compatible with the notion of size in class $\mathcal{C}$ we have a *grading* in the algebra $\mathscr{C}$ as explained in the following proposition:

Proposition 4 (Grading)

Suppose we have a bi-algebra structure $(\mathscr{C}, +, *, \emptyset, \Delta, \varepsilon)$ constructed as in Theorem 1. If condition **(CD2)** *holds, then $\mathscr{C}$ is a graded Hopf algebra with grading given by size in $\mathcal{C}$, i.e.*

$$\mathscr{C} = \bigoplus_{n\in\mathbb{N}} \mathscr{C}_n \,, \qquad \mathscr{C}_n = \mathsf{Span}\,(\mathcal{C}_n) \,, \tag{49}$$

where $\mathcal{C}_n = \{\Gamma \in \mathcal{C} : |\Gamma| = n\}$, and

$$* : \mathscr{C}_i \times \mathscr{C}_j \longrightarrow \mathscr{C}_{i+j} \,, \qquad \Delta : \mathscr{C}_k \longrightarrow \bigoplus_{i+j=k} \mathscr{C}_i \otimes \mathscr{C}_j \;. \tag{50}$$

Proof: Note, that condition **(CD2)** implies **(D5)**, and hence $\mathscr{C}$ is a Hopf algebra by Theorem 3. Furthermore, condition **(CD2)** asserts a proper action of $*$ and Δ on subspaces $\mathscr{C}_n$ built of objects of the same size.

□

4.4 A special case: Monoid

Let us consider a simplified situation by taking a determinate composition law of the form

$$\blacktriangleleft : \mathcal{C} \times \mathcal{C} \quad \longrightarrow \quad \mathcal{C} \,, \tag{51}$$

which means that objects compose in a unique way. In other words, for each $\Gamma_2, \Gamma_1 \in \mathcal{C}$ the multiset $\Gamma_2 \blacktriangleleft \Gamma_1$ of Definition 1 is always a singleton. Observe that conditions **(C2)** and **(C3)** are equivalent to the requirement that $(\mathcal{C}, \blacktriangleleft)$ is a *monoid*. We note that the case of a commutative monoid was thoroughly investigated by S.A. Joni and G.-C. Rota in [JR79] and further developed by A. Joyal [Joy81].

In this context it is convenient to consider collections $\mathfrak{C} \subset \mathcal{C}$ such that each element of $\mathcal{C}$ can be constructed as a composition of a finite number of elements from $\mathfrak{C}$, *i.e.*

$$\mathcal{C} = \{\; \Gamma_n \blacktriangleleft \ldots \blacktriangleleft \Gamma_1 \;:\, \Gamma_n, \ldots, \Gamma_1 \in \mathfrak{C} \;\} \;. \tag{52}$$

We call $\mathfrak{C}$ a *generating class* if it is the smallest (in the sense of inclusion) subclass of $\mathcal{C}$ with this property. It has the advantage that when establishing a decomposition rule satisfying **(CD1)** one can specify it on the generating class $\mathfrak{C}$ in arbitrary way

$$\langle\,\cdot\,\rangle : \mathfrak{C} \quad \longrightarrow \quad \mathrm{MSET}\,(\mathcal{C} \times \mathcal{C}) \,, \tag{53}$$

and then consistently extend it using Eq. (16) to the whole class $\mathcal{C}$ by defining

$$\langle \Gamma_n \blacktriangleleft \ldots \blacktriangleleft \Gamma_1 \rangle \;=\; \biguplus_{\substack{(\Gamma''_n, \Gamma'_n) \in \langle \Gamma_n \rangle \\ \cdots \\ (\Gamma''_1, \Gamma'_1) \in \langle \Gamma_1 \rangle}} \{(\Gamma''_n \blacktriangleleft \ldots \blacktriangleleft \Gamma''_1 \,,\, \Gamma'_n \blacktriangleleft \ldots \blacktriangleleft \Gamma'_1)\} \;. \tag{54}$$

We note that from a practical point of view this way of introducing the decomposition rule is very convenient as it restricts the the number of objects to be scrutinized to a smaller class $\mathfrak{C}$ and automatically guarantees compatibility of composition and decomposition rules, *i.e.* **(CD1)** is satisfied by construction. Moreover, inspection of other properties is usually simpler in this context as well. For example, if composition preserves size Eq. (17), then it is enough to check Eq. (19) on $\mathfrak{C}$ and condition **(CD2)** automatically holds on the whole $\mathcal{C}$.

There is a canonical way in which decomposition can be introduced in this setting. Namely, one can define it on the generating elements $\Gamma \in \mathfrak{C}$ in a *primitive* way, *i.e.*

$$\langle \Gamma \rangle = \{(\emptyset, \Gamma), (\Gamma, \emptyset)\} \,. \tag{55}$$

Observe that such defined decomposition rule upon extension via Eq. (54) satisfies all the conditions **(D1)** – **(D5)**, which clears the way to the construction of a Hopf algebra. We note that objects having the property of Eq. (55) are usually called *primitive elements*, for which we have

$$\Delta(\Gamma) = \emptyset \otimes \Gamma + \Gamma \otimes \emptyset \,, \tag{56}$$

$$S(\Gamma) = -\Gamma \,. \tag{57}$$

5 Examples

Here, we will illustrate how the general framework developed in Sections 3 and 4 works in practice. We give a few examples of combinatorial objects which via composition/decomposition scheme lead to Hopf algebra structures. For other examples see *e.g.* [JR79, Car07, GL89, BDH$^+$10].

5.1 Words

Let $\mathfrak{A} = \{l_1, l_2, ..., l_n\}$ be a finite set of *letters* – an *alphabet*. We will consider a combinatorial class $\mathcal{A}$ consisting of (finite) words built of the alphabet $\mathfrak{A}$, *i.e.* $\mathcal{A} = \mathfrak{A}^* = \{\, \emptyset, l_1, l_1 l_1, l_1 l_2, \ldots, l_{i_1} ... l_{i_k}, ... \}$ where $\emptyset$ is an empty word. Size of a word will be defined as its lengths (number of letters): $|\, l_{i_1} ... l_{i_k}| = k$ and $|\emptyset| = 0$. Algebraic structure in $\mathcal{A}$ can be introduced in a few ways as explained below [Lot83, Reu93].

5.1.1 Free algebra of words

The simplest composition rule for words is given by concatenation, *i.e.*[(iv)]

$$l_{i_1} ... l_{i_m} \blacktriangleleft\ l_{j_1} ... l_{j_n} = l_{i_1} ... l_{i_m} l_{j_1} ... l_{j_n} \,. \tag{58}$$

Observe that $(\mathcal{A}, \blacktriangleleft)$ is a monoid and $\mathfrak{A}$ is a generating class. We define decomposition of generators (letters) in the primitive way, *i.e.*

$$\langle l_i \rangle = \{(\emptyset, l_i), (l_i, \emptyset)\} \,, \tag{59}$$

and extend it to the whole class $\mathcal{A}$ using Eq. (54). One checks that each decomposition of a word comes down to the choice of a subword which gives the first component of a splitting (the reminder constitutes the second one), *i.e.*

$$\langle l_{i_1} ... l_{i_k} \rangle = \biguplus_{\substack{j_1 < ... < j_m \\ j_{m+1} < ... < j_k}} \{(l_{i_{j_1}} ... l_{i_{j_m}}, l_{i_{j_{m+1}}} ... l_{i_{j_k}})\} \tag{60}$$

(iv) We adopt the convention that a sequence of letters indexed by the empty set is the empty word $\emptyset$.

Note such defined composition/decomposition rule is compatible with size and hence condition **(CD2)** holds. Application of the scheme discussed in previous sections provide us with the mappings

$$l_{i_1}\ldots l_{i_m} * \, l_{j_1}\ldots l_{j_n} = l_{i_1}\ldots l_{i_m} l_{j_1}\ldots l_{j_n} \ , \tag{61}$$

$$\Delta(l_{i_1}\ldots l_{i_k}) = \sum_{\substack{j_1<\ldots<j_m \\ j_{m+1}<\ldots<j_k}} l_{i_{j_1}}\ldots l_{i_{j_m}} \otimes \, l_{i_{j_{m+1}}}\ldots l_{i_{j_k}} \ , \tag{62}$$

$$\varepsilon(l_{i_1}\ldots l_{i_k}) = 0 \ , \qquad \varepsilon(\emptyset) = 1 \ , \tag{63}$$

$$S(l_{i_1}\ldots l_{i_k}) = (-1)^k \, l_{i_k}\ldots l_{i_1} \ , \tag{64}$$

which make $\mathscr{A}$ into a graded co-commutative Hopf algebra. It is called a *free algebra*. Note that if the alphabet consists of more than one letter then multiplication is non-commutative.

In conclusion, we observe that if the alphabet consists of one letter only $\mathfrak{A} = \{x\}$, then the construction starts from the the class of words $\mathcal{P} = \{\emptyset, x, xx, xxx, \ldots\}$ and leads to the *algebra of polynomials* in one variable $\mathscr{P} = \mathbb{K}[x] = \left\{ \sum_{i=0}^{n} \alpha_i \, x^i \, : \alpha_i \in \mathbb{K} \right\}$. In this case, we have

$$x^i * x^j = x^{i+j} \ , \quad \Delta(x^n) = \sum_{i=0}^{n} \binom{n}{i} \, x^i \otimes x^{n-i} \ , \quad \varepsilon(x^n) = \delta_{n,0} \ , \quad S(x^n) = (-1)^n \, x^n \ . \tag{65}$$

5.1.2 Symmetric algebra

Now, let an alphabet $\mathfrak{A} = \{l_1, l_2, \ldots, l_n\}$ be endowed with a linear order $l_1 < l_2 < \ldots < l_n$. We will consider words arranged in a non-decreasing order and define the pertaining class $\mathcal{S}$ in the form

$$\mathcal{S} = \{l_{i_1}\ldots l_{i_n} : i_1 \leqslant \ldots \leqslant i_n \in \mathbb{N}\} \ . \tag{66}$$

In this case, simple concatenation of words is not a legitimate composition rule and one has to amend it by additional reordering of letters

$$l_{i_1}\ldots l_{i_m} \blacktriangleleft \, l_{i_{m+1}}\ldots l_{i_{m+n}} = \, l_{i_{\sigma(1)}}\ldots l_{i_{\sigma(m+n)}} \ , \tag{67}$$

where σ is a unique permutation of $\{1, 2, \ldots, m+n\}$ such that $i_{\sigma(1)} \leqslant \ldots \leqslant i_{\sigma(m+n)}$. Clearly, $(\mathcal{S}, \blacktriangleleft)$ is a monoid generated by $\mathfrak{A}$. The simplest choice of primitive decomposition for the generators $\langle l_i \rangle = \{(\emptyset, l_i), (l_i, \emptyset)\}$ extends to the whole class as follows

$$\langle l_{i_1}\ldots l_{i_k} \rangle = \biguplus_{\substack{j_1<\ldots<j_m \\ j_{m+1}<\ldots<j_k}} \{(l_{i_{j_1}}\ldots l_{i_{j_m}}, l_{i_{j_{m+1}}}\ldots l_{i_{j_k}})\} \tag{68}$$

Observe apparent similarity of Eqs. (58) and (60) to Eqs. (67) and (68), with the only difference that words in the latter two are ordered. Construction of the *symmetric algebra* $\mathscr{S}$ follows the proposed scheme, and the mappings

$$l_n^{i_n}\ldots l_1^{i_1} * \, l_n^{j_n}\ldots l_1^{j_1} = \, l_n^{i_n+j_n}\ldots l_1^{i_1+j_1} \ , \tag{69}$$

$$\Delta(l_{i_1}...\,l_{i_k}) = \sum_{\substack{j_1<...<j_m \\ j_{m+1}<...<j_k}} l_{i_{j_1}}...\,l_{i_{j_m}} \otimes\, l_{i_{j_{m+1}}}...\,l_{i_{j_k}} \;, \tag{70}$$

$$\varepsilon(l_{i_1}...\,l_{i_k}) = 0 \;, \qquad \varepsilon(\text{Ø}) = 1 \;, \tag{71}$$

$$S(l_{i_1}...\,l_{i_k}) = (-1)^k\, l_{i_1}...\,l_{i_k} \;, \tag{72}$$

define a graded Hopf algebra structure which is both commutative and co-commutative. Note that in Eq. (69) the repeating letters were grouped together and denoted as powers. As a byproduct of this notation one immediately observes that the symmetric algebra $\mathscr{S}$ is isomorphic to the *algebra of polynomials* in many commuting variables $\mathbb{K}[x_1, ..., x_n]$.

5.1.3 Shuffle algebra

We will consider the class of words $\mathcal{A}$ and define composition as any *shuffle* which mixes letters of the words preserving their relative order. For example, for two words "shuffle" and "mix": "shmiufxfle" and "mixshuffle" are allowed compositions whilst "shufflemxi" is not. Note that there are always several possible shuffles for given two (nonempty) words, and hence the use of multiset construction in definition of the composition rule

$$l_{i_1}...\,l_{i_m} \blacktriangleleft\, l_{i_{m+1}}...\,l_{i_{m+n}} = \biguplus_{\substack{\sigma(1)<...<\sigma(m) \\ \sigma(m+1)<...<\sigma(m+n)}} \{l_{i_{\sigma(1)}}...\,l_{i_{\sigma(m)}} l_{i_{\sigma(m+1)}}...\,l_{i_{\sigma(m+n)}}\} \;, \tag{73}$$

where the index set runs over all permutations σ of the set $\{1, 2, ..., m+n\}$ which preserve the relative order of $1, 2, ..., m$ and $m+1, m+2, ...m+n$ respectively. One checks that a compatible decomposition rule is given by cutting a word into two parts and exchanging the prefix with the suffix, *i.e.*

$$\langle l_{i_1}...\,l_{i_k}\rangle = \biguplus_{j=0,...,k} \{(l_{i_{j+1}}...\,l_{i_k}, l_{i_1}...\,l_{i_j})\} \;. \tag{74}$$

Note that this is the instance of a non-monoidal composition law. Following the scheme of Section 4 we arrive at the Hopf algebra structure given by the mappings

$$l_{i_1}...\,l_{i_m} * \,l_{i_1}...\,l_{i_n} = \sum_{\substack{\sigma(1)<...<\sigma(m) \\ \sigma(m+1)<...<\sigma(m+n)}} l_{i_{\sigma(1)}}...\,l_{i_{\sigma(m)}} l_{i_{\sigma(m+1)}}...\,l_{i_{\sigma(m+n)}} \;, \tag{75}$$

$$\Delta(l_{i_1}...\,l_{i_k}) = \sum_{j=0}^{k} \; l_{i_{j+1}}...\,l_{i_k} \otimes\, l_{i_1}...\,l_{i_j} \;, \tag{76}$$

$$\varepsilon(l_{i_1}...\,l_{i_k}) = 0 \;, \qquad \varepsilon(\text{Ø}) = 1 \;, \tag{77}$$

$$S(l_{i_1}...\,l_{i_k}) = (-1)^k\, l_{i_k}...\,l_{i_1} \;, \qquad S(\text{Ø}) = \text{Ø} \;. \tag{78}$$

We remark that for such constructed *Shuffle algebra* the multiplication of Eq. (75) is commutative and the co-product of Eq. (76) is not co-commutative.

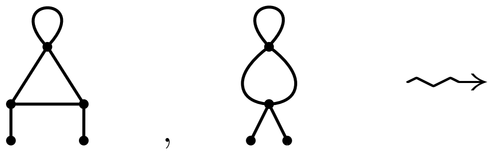
5.2 Graphs

Let us consider a class of undirected *graphs* $\mathcal{G}$ which one graphically represents as a collection of vertices connected by edges (we exclude isolated vertices). More formally a graph is defined as a mapping $\Gamma : E \longrightarrow V^{(2)}$ prescribing how the edges E are attached to vertices V, where $V^{(2)}$ is a set of unordered pairs of vertices (not necessarily distinct); for a rigorous definition see [Ore90, Wil96, Die05]. Let the size of a graph be the number of its edges $|\Gamma| = |E|$.

An obvious composition rule in $\mathcal{G}$ consist in taking two graphs $\Gamma_2, \Gamma_1 \in \mathcal{G}$ and drawing them one next to another, *i.e.*

$$\Gamma_2 \blacktriangleleft \Gamma_1 = \Gamma_2 \Gamma_1 \ , \tag{79}$$

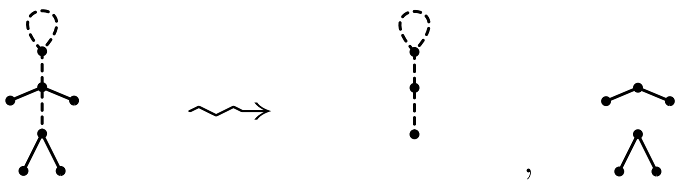
where formally $\Gamma_2\Gamma_1 : E_2 \uplus E_1 \longrightarrow V_2^{(2)} \uplus V_1^{(2)}$, such that $\Gamma_2\Gamma_1|_{E_2} = \Gamma_2$ and $\Gamma_2\Gamma_1|_{E_1} = \Gamma_1$.
For example:



Γ_2 Γ_1 $\Gamma_2 \ \Gamma_1$

Such defined composition rule satisfies conditions **(C1) – (C4)** and establishes a commutative algebra $\mathscr{G}$ (with void graph Ø being the identity).

Observe that for a given graph $\Gamma \in \mathcal{G}$ each subset of its edges $L \subset E$ induces a subgraph $\Gamma|_L : L \longrightarrow V^{(2)}$ which is defined by restriction of Γ to the subset L. Likewise, the remaining part of the edges $R = E - L$ gives rise to a subgraph $\Gamma|_R$. Thus, by considering ordered partitions of the set of edges into two subsets $L + R = E$, *i.e.* $L \cup R = E$ and $L \cap R = \emptyset$, we end up with pairs $(\Gamma|_L, \Gamma|_R)$ of disjoint graphs. For example:



Γ $\Gamma|_L$, $\Gamma|_R$

is a decompositions of a graph $\Gamma \rightsquigarrow (\Gamma|_L, \Gamma|_R)$ induced by the choice of the dashed edges $L \subset E$ which make the left component in the splitting. This suggests the following definition of the decomposition rule

$$\langle \Gamma \rangle = \biguplus_{L+R=E} \{(\Gamma|_L, \Gamma|_R)\} \ . \tag{80}$$

One checks that conditions **(D1) – (D5)** and **(CD2)** hold, and we obtain a graded Hopf algebra $\mathscr{G}$ with the grading given by the number of edges. Its structure is given by

$$\Gamma_2 * \Gamma_1 = \Gamma_2 \Gamma_1 \ , \tag{81}$$

$$\Delta(\Gamma) = \sum_{L+R=E} \Gamma|_L \otimes \Gamma|_R \ , \tag{82}$$

$$\varepsilon(\Gamma) = 0 \ , \qquad \varepsilon(\emptyset) = 1 \ , \tag{83}$$

$$S(\Gamma) = \sum_{\substack{L_1+\ldots+L_n=E \\ L_1,\ldots,L_n \neq \emptyset}} (-1)^n \ \Gamma|_{L_1} \ldots \Gamma|_{L_n} \ , \qquad S(\emptyset) = \emptyset \ . \tag{84}$$

So defined *algebra of graphs* is both commutative and co-commutative.

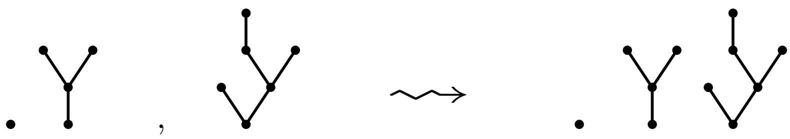
5.3 Trees and Forests

A *rooted tree* is a graph without cycles with one distinguished vertex, called the root. Let $\mathcal{T}$ denote the class of rooted trees. A *forest* is a collection of rooted trees and the pertaining combinatorial class has the specification $\mathcal{F} = \mathrm{MSET}(\mathcal{T})$. Size of a tree (forest) is defined as the number of vertices.

We will consider class $\mathcal{F}$ and define composition of forests as a multiset union (like for graphs), *i.e.*

$$\Gamma_2 \blacktriangleleft \Gamma_1 = \Gamma_2 \Gamma_1 \ . \tag{85}$$

For example:



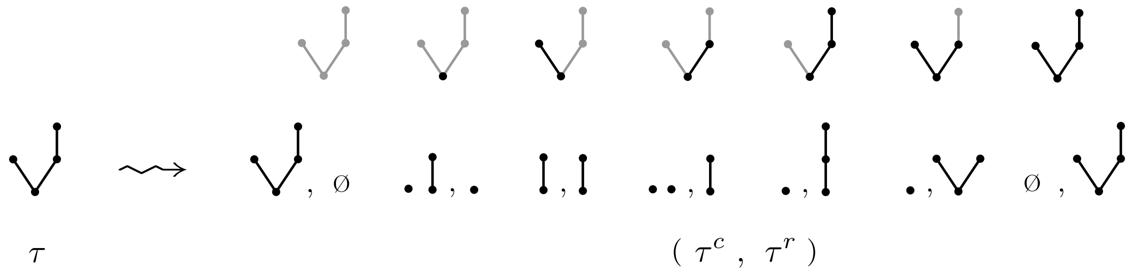
Γ_2 , Γ_1 $\rightsquigarrow$ $\Gamma_2 \ \Gamma_1$

Note that $(\mathcal{F}, \blacktriangleright)$ is a (commutative) monoid generated by the rooted trees $\mathcal{T}$.

For a given a tree $\tau \in \mathcal{T}$ one distinguishes subtrees $\tau^r \subset \tau$ which share the same root with τ, called proper subtrees (the empty tree Ø is considered as a proper subtree as well). Observe that the latter obtains by trimming τ to the required shape τ^r, and the branches which are cut off form a forest of trees denoted by τ^c (with the roots next to the cutting). Decomposition of a tree is defined as any splitting $\tau \rightsquigarrow (\tau^c, \tau^r)$ into a pair consisting of a proper subtree taken in the second component and the remaining forest in the first one. In other words

$$\langle \tau \rangle = \biguplus_{\tau^r \subset \tau} \{(\tau^c, \tau^r)\} \ , \tag{86}$$

where the disjoint union ranges over proper subtrees τ^r of τ, and τ^c is a forest of trees which 'complements' τ^r to τ. For example:



enumerates all possible decompositions of a tree (at the top row proper subtrees τ^r are drawn in black while the completing forests τ^c are drawn in gray).
Since trees $\mathcal{T}$ generate forests $\mathcal{F}$, we extend the decomposition rule to any forest $\Gamma = \tau_n \dots \tau_1 \in \mathcal{F}$ using Eq. (54) and obtain

$$\langle \tau_n \dots \tau_1 \rangle = \biguplus_{\tau_n^r \subset \tau_n\,,\,\dots\,,\,\tau_1^r \subset \tau_1} \{(\tau_n^c \dots \tau_1^c, \tau_n^r \dots \tau_1^r)\}\ , \tag{87}$$

which comes down to trimming some of the branches off the whole forest and gathering them in the first component $\Gamma^c = \tau_n^c \dots \tau_1^c$ whilst keeping the rooted parts in the second one $\Gamma^r = \tau_n^r \dots \tau_1^r$. Hence, we will briefly write

$$\langle \Gamma \rangle = \biguplus_{\Gamma^r \subset \Gamma} \{(\Gamma^c, \Gamma^r)\}\ . \tag{88}$$

Following the construction of Section 4 one obtains a graded Hopf algebra $\mathscr{F}$ with the grading given by the number of vertices. The required mappings take the form

$$\Gamma_2 * \Gamma_1 = \Gamma_2\ \Gamma_2\ , \tag{89}$$

$$\Delta(\Gamma) = \sum_{\Gamma^r \subset \Gamma} \Gamma^c \otimes \Gamma^r\ , \tag{90}$$

$$\varepsilon(\Gamma) = 0\ , \qquad \varepsilon(\emptyset) = 1\ , \tag{91}$$

$$S(\Gamma) = -\sum_{\Gamma^r \subset \Gamma} S(\Gamma^c)\ \Gamma^r\ , \qquad S(\emptyset) = \emptyset\ . \tag{92}$$

Such constructed *algebra of forests* $\mathscr{F}$ is commutative but not co-commutative. We remark that this Hopf algebra was first introduced by J. C. Butcher [But72, Bro04] and recently it was rediscovered by A. Connes and D. Kreimer [Kre98, CK98] in the context of renormalization in quantum field theory.

Acknowledgements

I wish to thank Gerard Duchamp, Philippe Flajolet, Andrzej Horzela, Karol A. Penson and Allan I. Solomon for important discussions on the subject. Most of this research was carried out in the Mathematisches Forschungsinstitut Oberwolfach (Germany) and the Laboratoire d'Informatique de l'Université Paris–Nord in Villetaneuse (France) whose warm hospitality is greatly appreciated. The author acknowledges support from the Agence Nationale de la Recherche under the programme no. ANR-08-BLAN-0243-2 and the Polish Ministry of Science and Higher Education grant no. N202 061434.

References

[Abe80] E. Abe. *Hopf Algebras*. Cambridge University Press, 1980.

[BDH+10] P. Blasiak, G. H. E. Duchamp, A. Horzela, K. A. Penson, and A. I. Solomon. Combinatorial Algebra for second-quantized Quantum Theory. *Adv. Theor. Math. Phys.*, 14(4), 2010. Article in Press, arXiv:1001.4964 [math-ph].

[BLL98] F. Bergeron, G. Labelle, and P. Leroux. *Combinatorial Species and Tree-like Structures*. Cambridge University Press, 1998.

[Bou89] N. Bourbaki. *Algebra I: Chapters 1–3*. Springer-Verlag, 1989.

[Bro04] C. Brouder. Trees, Renormalization and Differential Equations differential equations. *BIT Num. Math.*, 44:425–438, 2004.

[But72] J. C. Butcher. An Algebraic Theory of Integration Methods. *Math. Comput.*, 26:79–106, 1872.

[Car07] P. Cartier. A Primer of Hopf Algebrasrimer of hopf algebras. In *Frontiers in Number Theory, Physics, and Geometry II*, pages 537–615. Springer, 2007.

[CK98] A. Connes and D. Kreimer. Hopf Algebras, Renormalization and Noncommutative Geometry. *Commun. Math. Phys.*, 199:203–242, 1998.

[Die05] R. Diestel. *Graph Theory*. Springer-Verlag, 3rd edition, 2005.

[FS09] P. Flajolet and R. Sedgewick. *Analytic Combinatorics*. Cambridge University Press, 2009.

[GJ83] I. P. Goulden and D. M. Jackson. *Combinatorial Enumeration*. John Wiley & Sons, 1983.

[GL89] R. Grossman and R. G. Larson. Hopf-Algebraic Structure of Families of Trees. *J. Algebra*, 126(1):184–210, 1989.

[Joy81] A. Joyal. Une théorie combinatoire des séries formelles. *Adv. Math.*, 42(1):1–82, 1981.

[JR79] S. A. Joni and G. C. Rota. Coalgebras and Bialgebras in Combinatorics. *Stud. Appl. Math.*, 61:93–139, 1979.

[Kre98] D. Kreimer. On the Hopf algebra structure of perturbative quantum field theory. *Adv. Theor. Math. Phys.*, 2:303–334, 1998.

[Lot83] M. Lothaire. *Combinatorics on Words*. Addison-Wesley, 1983.

[Ore90] O. Ore. *Graphs and their Uses*. Mathematical Association of America, 2nd edition, 1990.

[Reu93] C. Reutenauer. *Free Lie Algebras*. Oxford University Press, 1993.

[Swe69] M. E. Sweedler. *Hopf Algebras*. Benjamin, 1969.

[Wil96] R. J. Wilson. *Introduction to Graph Theory*. Addison-Wesley, 4th edition, 1996.