Humboldt-Universität zu Berlin

Master thesis

Filtrations in Dyson-Schwinger equations

Olaf Krüger
Matrikel number: 529499
Email: okrueger@physik.hu-berlin.de

November 16, 2013

supervised by

1. Prof. Dr. Dirk Kreimer
2. Dr. Christian Bogner

Mathematisch-Naturwissenschaftliche Fakultät I - Institut für Physik
Newtonstraße 15
12489 Berlin

This thesis investigates the structure of the solution of a non-linear Dyson-Schwinger equation. In particular, a program was written in FORM, which computes the perturbative coefficients of the solution as elements in the Hopf algebra of words. Furthermore, the program performs a filtration of the coefficients, each filtrated term maps to a certain power of the external scale parameter in the log-expansion of the associated renormalized Feynman amplitude. Finally, the multiplicities of certain filtrated terms, corresponding to next-to-. . .-leading-log terms in the log-expansion, form some series, whose generating functions are determined.

Contents

1. Introduction

Quantum field theory provides a perturbative calculation of probability amplitudes of physical processes between subatomic particles. In particular, quantum electrodynamics and the standard model of particle physics are among the most successful physical theories. The usual way to compute a physical probability amplitude was introduced by Richard Feynman [1]. For a given quantum field theory, each term of the perturbative series can be replaced by a Feynman graph and Feynman rules translate these graphs to a Feynman amplitude. In general, Feynman amplitudes turn out to be divergent, which can be fixed by renormalization. This process extracts the physical interesting finite terms out of the divergent amplitude, which yields the physically observable renormalized Feynman amplitude. As turns out, the perturbative series of the renormalized Feynman amplitude is a function of an external scale parameter L, which includes, for example, the center of mass energy of the process and a collection θ of scattering angle parameters. Moreover, this function is a polynomial in L, which is called log-expansion of the renormalized Feynman amplitude [2, 3].

Dirk Kreimer and Alain Connes explored the Hopf algebraic structure of Feynman graphs [4]. They discovered that all divergent one-particle irreducible Feynman graphs (i.e. graphs, which are still connected after removal of any one internal edge) of any physical quantum field theory generate a Hopf algebra $\mathcal{H}$, called the Hopf algebra of Feynman graphs. Using this mathematical structure, Dyson-Schwinger equations can be introduced, which are fix-point equations for Feynman graphs. The point is that for any given quantum field theory, there always exists a Dyson-Schwinger equation, whose solution is simply related to the log-expansion of the probability amplitude of some physical process by application of renormalized Feynman rules. The structure of different Dyson-Schwinger equations and their classification was investigated by Loïc Foissy[5].

In the scope of this thesis, the perturbative coefficients of the solution of a physical Dyson-Schwinger equation were filtrated. Renormalized Feynman rules map each filtrated term to a certain power of the external scale parameter L in the log-expansion of the associated physical probability amplitude. Therefore, Chapter 2 gives all mathematical and physical background to Dyson-Schwinger equations in detail. Chapter 3 describes a canonical way to filtrate the coefficients of the solution of any physical Dyson-Schwinger equation. The filtration was done by a program, whose source code is given and explained in Appendix A. Chapter 4 discusses the resulting filtration for a representative non-linear Dyson-Schwinger equation. In particular, it was observed that the multiplicities of certain filtrated terms in different perturbative coefficients of the solution of the treated representative Dyson-Schwinger equation form some series. These series were investigated and their generating functions were determined. The required

calculations are collected in Appendix B. Finally, Chapter 5 concludes the thesis by a summary and gives an outlook for future work.

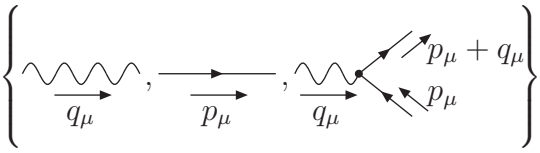
2. Preliminaries

This chapter is divided into six parts. Section 2.1 discusses the relation between Feynman graphs, Feynman rules, Feynman amplitudes and physical processes. Section 2.2 shows up occurring divergences in Feynman amplitudes due to integrations over internal loop momenta and how to erase them by regularization and renormalization. Section 2.3 gives the mathematical concept of a Hopf algebra. Therefore, Section 2.4 considers Feynman graphs as generators of such a Hopf algebra, which is called the Hopf algebra of Feynman graphs. This points out the relation between Feynman graphs in physics and the mathematical concept of a Hopf algebra. Section 2.5 introduces another mathematical structure, namely the grafting operator and its relation to Hochschild co-homology. All these mathematical tools are finally needed in Section 2.6, which discusses Dyson-Schwinger equations representing the equations of motion for Feynman amplitudes. Solving these equations gives important information about the combinatorial structure of Feynman amplitudes, which is indispensable for the calculation of probability amplitudes of physical processes.

2.1. From a physical process to Feynman graphs

In any quantum field theory, it is desired to calculate the probabilities of different interaction processes. The standard method is to draw some Feynman graph associated to the process and to calculate its Feynman amplitude using well known Feynman rules [1]. These rules are given by the considered theory. Each Feynman graph consists of edges (denoting propagating particles) and vertices (denoting interaction between particles). In order to differentiate between different particles of the theory, there are different kinds of edges. For example in quantum electrodynamics (Q.E.D.) [6], the standard notation is a wiggly line for the photon and a solid arrow line for the fermion. The arrow contains information, weather the fermion is a particle or an antiparticle. Additionally, each edge is labeled by the orientated 4-momentum of the associated particle s.t. 4-momentum conservation does hold at each vertex and therefore within each interaction. Furthermore, the set of edges splits into the set of internal and the set of external edges and only the latter correspond to physical observable particles, whereas the former are associated with non-physical particles. Shrinking the set of internal edges of a Feynman graph to one vertex yields its residue r and $\mathcal{R}$ denotes the set of all residues for a given theory.

For example in Q.E.D.,

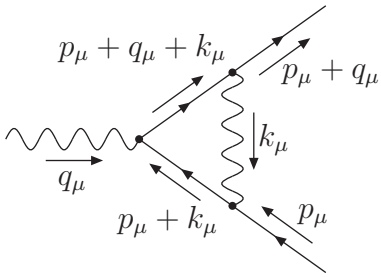
$\mathcal{R}_{\text{Q.E.D.}} =$ 

. (2.1)

The next steps require

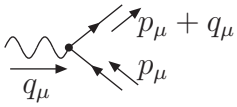
Definition 2.1.1. *A Feynman graph Γ is called one-particle irreducible (1PI), iff the graph remains connected after removal of any one internal edge.*

Consider for example the Q.E.D. one-loop vertex graph (i.e. a Feynman graph with three external edges)

$\Gamma =$ 

. (2.2)

It is 1PI and its residue is

$r(\Gamma) =$ 

. (2.3)

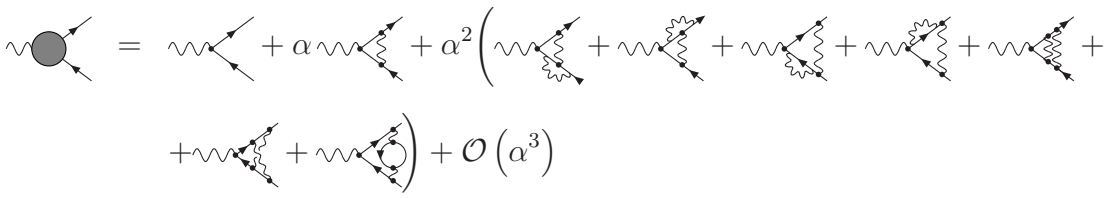
It is clear that a physical probability of an interaction process cannot depend on the internal edge structure of any associated Feynman graph, because these edges belong to unphysical particles. However, the residue r of a Feynman graph does contain all physical information about the associated process, in particular the kinds of interacting particles and their 4-momenta. It is therefore convenient to call an interaction process to be 'of a certain residue', which contains all external information. Finally, the Feynman amplitude $\phi(\Gamma)$ of a Feynman graph Γ with a certain residue is given by applying Feynman rules to it. These can be considered as a linear map $\phi : \Gamma \to \phi(\Gamma)$. At this point, it is necessary to consider all Feynman graphs as elements of a vector space, otherwise, the linearity of ϕ does not make sense. Thus, for two Feynman graphs Γ_1 and Γ_2, which are elements of the considered vector space, $\phi(\lambda_1\Gamma_1 + \lambda_2\Gamma_2) = \lambda_1\phi(\Gamma_1) + \lambda_2\phi(\Gamma_2)$, where $\lambda_1, \lambda_2 \in \mathbb{C}$. The structure of the vector space of Feynman graphs will be discussed in details in Section 2.4. The physically interesting point is that Feynman amplitudes are related to the probability amplitude of some interaction process by

Proposition 2.1.2. *The probability amplitude of some interaction process of a certain residue is proportional to the renormalized Feynman amplitude of the sum of all Feynman graphs with the respective residue. Furthermore, it can be expressed through the renormalized Feynman amplitudes of each sum of all 1PI Feynman graphs with a particular residue.*

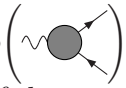
The proof of this powerful proposition is given in any textbook on quantum field theory [6]. However, this section does not explain the meaning of a 'renormalized' Feynman amplitude. This is postponed to the next section. An example for the meaning of the last sentence in Proposition 2.1.2 will be shown in Section 2.6. At this point, it is important to notice that only 1PI Feynman graphs need to be evaluated by Feynman rules in order to compute physical probability amplitudes.

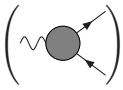
2.2. Divergences in Feynman amplitudes and the concept of renormalization

The problem of writing Proposition 2.1.2 without the word 'renormalized' becomes clear by a look at some Feynman rules. E.g., applying the Feynman rules of Q.E.D. to Γ in Eq. (2.2) results in a divergent integral over the loop momentum k_μ (see any textbook on quantum field theory [6]). Thus, the Feynman amplitude of this graph, $\phi(\Gamma)$ cannot be well defined. The solution to this dilemma is given by the regularization and renormalization of the theory. First, Feynman rules imply that for all Feynman graphs Γ, $\phi(\Gamma)$ is proportional to a loop counting parameter α to the power of the first Betti number (number of independent loops) of Γ. As long as α is smaller than 1, the quantum field theory is perturbative. It is more convenient to abstract α from the Feynman rules and to multiply each Feynman graph with the respective power of α directly in the vector space of Feynman graphs. Then, the sum of all 1PI Feynman graphs of a certain residue is denoted by a blob, e.g. the sum of all 1PI vertex graphs in Q.E.D. is



$$= \quad + \alpha \quad + \alpha^2 \Big(\quad + \quad + \quad + \quad + \quad + \quad + \quad + \quad \Big) + \mathcal{O}\left(\alpha^3\right) \tag{2.4}$$

where all 4-momenta labelings were dropped for convenience. Since this work is about the combinatorics of Feynman graphs and since the interest is not in Feynman amplitudes, these labels are not needed for the whole work. They can always be added in a simple way. Perturbativity of quantum field theory permits to cut off the sum at some order in α. This approximates the Feynman amplitudes. Sometimes the quantum field theory itself is then called 'theory at a certain loop order'. Now, calculating the Feynman amplitude of the sum in Eq. (2.4) yields an infinite number and is thus, not well defined. However, it is possible to write each divergent integral of $\phi\Big($  $\Big)$ as a finite integral depending on some dummy parameter s.t. taking the limit of that parameter to a certain value gives back the original divergent integral. The choice of such a parameter is not unique and is called regularization scheme. The process of handling these divergences in general as a limit of a dummy parameter is called regularization. Of course, nothing changed for the

Feynman amplitudes. They are still ill defined and are written in a different form with a divergent limit instead of divergent integrals. However, taking the limit of the dummy parameter yields different kinds of divergences. Some integrals will be logarithmically, some quadratically divergent and so on. In this way, all divergences can be classified. The crucial point to handle the infinities is in the renormalization of the theory. Therefore, it is important that each quantum field theory comes with a finite number of coupling constants, e.g. in Q.E.D., the electromagnetic charge. In addition, there are the wave functions and masses of the interacting particles. The Feynman amplitudes of the theory will all depend on these parameters. Now, it is convenient to consider them as 'bare' or unphysical parameters. This makes sense, because the Feynman amplitudes are then unphysical too, for example $\phi\Bigg($  $\Bigg)$ above. Remember, they are even ill defined. A useful concept is then

Definition 2.2.1. *A quantum field theory is called renormalizable, iff the number of different classes of divergent integrals in all Feynman amplitudes is at most equal to the number of bare parameters of the theory. Thus, this number remains finite, even at infinite loop order.*

For example, Q.E.D. and the standard model of particle physics are renormalizable [7]. Finally, the coupling constants, wave functions and masses of the quantum field theory can be redefined, s.t. they include the different divergences in the Feynman amplitudes. In particular, each of these parameters can include only one class of divergence and they are named 'renormalized' or physical parameters. One says, they 'absorb' the infinities, because after taking the limit of the dummy parameter, the Feynman amplitudes remain finite in terms of the renormalized coupling constants, masses and wave functions. They are then called renormalized Feynman amplitudes and are obtained by application of renormalized Feynman rules ϕ_R to Feynman graphs. The whole procedure is called renormalization and in fact, it requires a renormalizable theory, because each class of divergence in all Feynman amplitudes needs at least one bare parameter of the theory, in which it can be absorbed. In the end, it turns out that the renormalized Feynman amplitude ϕ_R of a sum of all 1PI Feynman graphs with a certain residue can be written in terms of the loop counting parameter α, an external scale parameter L and a collection of angle parameters, abbreviated by θ. Furthermore, this simplifies to a polynomial in L, i.e.

$$\phi_R(\alpha, L, \theta) = 1 + \sum_i \gamma_i(\alpha, \theta) L^i, \tag{2.5}$$

where $\gamma_i(\alpha, \theta)$ are functions in θ and polynomials in α. The parameter L is a logarithm of a kinetic invariant of the process (including the external 4-momenta) and Eq. (2.5) is called log-expansion of ϕ_R. For more details to renormalization and the derivation of Eq. (2.5), see [2, 3]. Here it is not necessary to know more about θ and L. This work investigates the structure of the sum in Eq. (2.5), which will be explained in Section 2.6 and moreover, in the next chapter. However, before doing this, it is necessary to

bring an order into the mess of possible Feynman graphs (up to a given loop order). Therefore, it is convenient to introduce the Hopf algebra of Feynman graphs, which will be done in the next two sections.

2.3. The concept of a Hopf algebra

This section is a general overview to the concept of Hopf algebras. It does not go into the very details. A more extensive work is Eric Panzers master thesis [8]. First, consider

Definition 2.3.1. *A vector space $\mathcal{A}$ together with an associative product $m : \mathcal{A}\otimes\mathcal{A} \to \mathcal{A}$,*

$$m \circ (\mathrm{id} \otimes m) = m \circ (m \otimes \mathrm{id}), \tag{2.6}$$

where id *is the identity map, and a unit $\mathbb{I} \in \mathcal{A}$ being the neutral element of m,*

$$\forall a \in \mathcal{A} : m(a \otimes \mathbb{I}) = m(\mathbb{I} \otimes a) = a \tag{2.7}$$

is called an (associative and unital) algebra $(\mathcal{A}, m, \mathbb{I})$.

This is well known in physics. Here and in the following, each vector space is meant over a field $\mathbb{K}$, which can be e.g. the real numbers $\mathbb{R}$ or the complex numbers $\mathbb{C}$. A slightly different, but very similar concept is

Definition 2.3.2. *A vector space $\mathcal{C}$ together with a co-associative product $\Delta : \mathcal{C} \to \mathcal{C}\otimes\mathcal{C}$,*

$$(\mathrm{id} \otimes \Delta) \circ \Delta = (\Delta \otimes \mathrm{id}) \circ \Delta \tag{2.8}$$

and a co-unit map $\hat{\mathbb{I}} : \mathcal{C} \to \mathbb{K}$,

$$(\mathrm{id} \otimes \hat{\mathbb{I}}) \circ \Delta = (\hat{\mathbb{I}} \otimes \mathrm{id}) \circ \Delta \cong \mathrm{id} \tag{2.9}$$

is called a (co-associative and co-unital) co-algebra $(\mathcal{C}, \Delta, \hat{\mathbb{I}})$.

In fact, unit and co-unit are unique [8]. It is not necessary to imagine any example of a co-algebra at this point. For convenience, an illustrating example is postponed to the next section, where also the following definitions will be included. The next step is to merge algebra and co-algebra into a bi-algebra. Therefore, let S_n be the usual permutation group of n elements and $\sigma \in S_n$ be any permutation of S_n. Then, for a given vector space V, it is convenient to define the permutation map

$$\tau_\sigma : v_1 \otimes \ldots \otimes v_n \to v_{\sigma_1} \otimes \ldots \otimes v_{\sigma_n} \quad \forall v_1, \ldots, v_n \in V, \tag{2.10}$$

which is needed in

Definition 2.3.3. *Let $\mathcal{B}$ be the vector space of an algebra $(\mathcal{B}, m, \mathbb{I})$ as well as the vector space of a co-algebra $(\mathcal{B}, \Delta, \hat{\mathbb{I}})$. Iff $m, \Delta, \mathbb{I}$ and $\hat{\mathbb{I}}$ hold*

$$\Delta \circ m = (m \otimes m) \circ \tau_{(2,3)} \circ (\Delta \otimes \Delta) \tag{2.11}$$

$$\Delta\mathbb{I} = \mathbb{I} \otimes \mathbb{I} \tag{2.12}$$

$$\hat{\mathbb{I}}(\mathbb{I}) = 1, \tag{2.13}$$

then, $(\mathcal{B}, m, \Delta, \mathbb{I}, \hat{\mathbb{I}})$ is called a bi-algebra.

Again, the concept of a bi-algebra becomes more clear in the next section. There is only one last step to define a Hopf algebra that is given in

Proposition 2.3.4. *Let $(\mathcal{B}, m_B, \Delta_B, \mathbb{I}_B, \hat{\mathbb{I}}_B)$ be a bi-algebra and $(\mathcal{A}, m_A, \mathbb{I}_A)$ be an algebra. The vector space V_c of linear maps $f : \mathcal{B} \to \mathcal{A}$, which are character-like, i.e.*

$$f(\mathbb{I}_B) = \mathbb{I}_A, \quad m_A \circ (f \otimes f) = f \circ m_B \tag{2.14}$$

together with the convolution product, defined as

$$f \star g = m_A \circ (f \otimes g) \circ \Delta_B \tag{2.15}$$

$\forall f, g \in V_c$, forms an algebra.

Proof. The convolution product is associative. For three linear maps $f, g, h \in V_c$,

$$\begin{aligned} f \star (g \star h) &= m_A \circ [f \otimes (m_A \circ (g \otimes h) \circ \Delta_B)] \circ \Delta_B \\ &= m_A \circ (\mathrm{id} \otimes m_A) \circ (f \otimes g \otimes h) \circ (\mathrm{id} \otimes \Delta_B) \circ \Delta_B \\ &= m_A \circ (m_A \otimes \mathrm{id}) \circ (f \otimes g \otimes h) \circ (\Delta_B \otimes \mathrm{id}) \circ \Delta_B \\ &= m_A \circ [(m_A \circ (f \otimes g) \circ \Delta_B) \otimes h] \circ \Delta_B \\ &= (f \star g) \star h, \end{aligned} \tag{2.16}$$

where the associativity of m_A (Eq. (2.6)) as well as the co-associativity of Δ_B (Eq. (2.8)) were used in the third line. Furthermore, the neutral element with respect to $\star$ is $\mathbb{I}_A \circ \hat{\mathbb{I}}_B$. Indeed,

$$\begin{aligned} f \star (\mathbb{I}_A \circ \hat{\mathbb{I}}_B) &= m_A \circ (f \otimes \mathbb{I}_A \circ \hat{\mathbb{I}}_B) \circ \Delta_B \\ &= m_A \circ (f \otimes f) \circ (\mathrm{id} \otimes \mathbb{I}_B) \circ (\mathrm{id} \otimes \hat{\mathbb{I}}_B) \circ \Delta_B \\ &= f \circ m_B \circ (\mathrm{id} \otimes \mathbb{I}_B) \\ &= f, \end{aligned} \tag{2.17}$$

where Eqs. (2.9, 2.14) were used in line three and Eq. (2.7) was used in line four. In full analogy, $(\mathbb{I}_A \circ \hat{\mathbb{I}}_B) \star f = f$ can be proved. Thus, $\left(V_c, \star, \mathbb{I}_A \circ \hat{\mathbb{I}}_B\right)$ forms an algebra. □

Lemma 2.3.5. *If furthermore, there exists a map $S : \mathcal{B} \to \mathcal{B}$, which fulfills*

$$m_B \circ (S \otimes \mathrm{id}) \circ \Delta_B = m_B \circ (\mathrm{id} \otimes S) \circ \Delta_B = \mathbb{I}_B \circ \hat{\mathbb{I}}_B, \tag{2.18}$$

then, $\left(V_c, \star, \mathbb{I}_A \circ \hat{\mathbb{I}}_B\right)$ forms a group, called the convolution group.

Proof. The inverse of the map $f \in V_c$ is given by $f^{\star -1} = f \circ S \in V_c$. Indeed,

$$\begin{aligned} f \circ S \star f &= m_A \circ (f \otimes f) \circ (S \otimes \mathrm{id}) \circ \Delta_B \\ &= f \circ m_B \circ (S \otimes \mathrm{id}) \circ \Delta_B \\ &= f \circ \mathbb{I}_B \circ \hat{\mathbb{I}}_B \\ &= \mathbb{I}_A \circ \hat{\mathbb{I}}_B, \end{aligned} \tag{2.19}$$

where Eqs. (2.14, 2.18) were used in lines two, three and four. $f \star f \circ S = \mathbb{I}_A \circ \hat{\mathbb{I}}_B$ can be proved in full analogy. □

As turns out, the convolution group is very convenient in renormalization of quantum field theories. For now, it is useful to introduce the concept of a Hopf algebra by

Definition 2.3.6. *Let* $(\mathcal{H}, m, \Delta, \mathbb{I}, \hat{\mathbb{I}})$ *be a bi-algebra. Iff there exists a map* S*, which fulfills*

$$m \circ (S \otimes \mathrm{id}) \circ \Delta = m \circ (\mathrm{id} \otimes S) \circ \Delta = \mathbb{I} \circ \hat{\mathbb{I}}, \tag{2.20}$$

then, $(\mathcal{H}, m, \Delta, \mathbb{I}, \hat{\mathbb{I}}, S)$ *forms a Hopf algebra. The map* S *is called antipode.*

Even if the sense of the antipode does not become clear right now (except for the existence of the convolution group), it plays a very important role in renormalization. However, at this point, only two properties of S will be mentioned, i.e. it fulfills

$$S \circ m = m \circ \tau_{(1,2)} \circ (S \otimes S) \tag{2.21}$$
$$\Delta \circ S = \tau_{(1,2)} \circ (S \otimes S) \circ \Delta. \tag{2.22}$$

After defining the concept of a Hopf algebra, consider as a final remark

Definition 2.3.7. *Let* $(\mathcal{H}, m, \Delta, \mathbb{I}, \hat{\mathbb{I}}, S)$ *be a Hopf algebra. A decomposition of* $\mathcal{H} = \bigoplus_{n \in \mathbb{N}_0} \mathcal{H}_n$ *is called a graduation, iff* $\forall n, n_1, n_2 \in \mathbb{N}_0$*,*

$$\Delta(\mathcal{H}_n) \subseteq \bigoplus_{i=0}^{n} \mathcal{H}_i \otimes \mathcal{H}_{n-i} \tag{2.23}$$
$$m(\mathcal{H}_{n_1} \otimes \mathcal{H}_{n_2}) \subseteq \mathcal{H}_{n_1+n_2} \tag{2.24}$$
$$S(\mathcal{H}_n) \subseteq \mathcal{H}_n \tag{2.25}$$

hold.

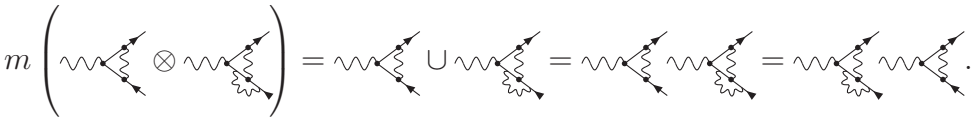
In any literature, most of the inductive proofs rely on the graduation of a Hopf algebra. However, there is no need for such a grading to exist. Hence, given a Hopf algebra, it remains to find a grading parameter in order to make the inductive proofs work. All the definitions made here will be illustrated in the next section.

2.4. An example: The Hopf algebra of Feynman graphs

The relation between the mathematical concept of a Hopf algebra introduced in the previous section and Feynman graphs of a physical quantum field theory will be explained here. Consider the vector space $\mathcal{H}$ of all divergent 1PI Feynman graphs and their disjoint unions. A Feynman graph is said to be divergent, if its Feynman amplitude is divergent. As shown in Definition 2.3.1, an associative product m as well as a unit $\mathbb{I}$ is needed, s.t. $(\mathcal{H}, m, \mathbb{I})$ forms an algebra. Define the product as the disjoint union of graphs, i.e.

$$\forall \Gamma_1, \Gamma_2 \in \mathcal{H}: \quad m(\Gamma_1 \otimes \Gamma_2) := \Gamma_1 \cup \Gamma_2 \tag{2.26}$$

which is often abbreviated as $\Gamma_1\Gamma_2$, because usual multiplication of Feynman graphs does not make sense. This product is commutative. For example,

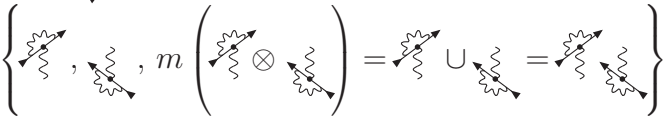


(2.27)

Associativity of m is trivial. The unit $\mathbb{I}$ of $\mathcal{H}$ is the empty graph $\emptyset$, which is naturally the neutral element with respect to multiplication m. Thus, Eq. (2.6) and Eq. (2.7) are fulfilled and $(\mathcal{H}, m, \mathbb{I})$ is clearly an algebra generated by all divergent 1PI Feynman graphs of a given theory and their disjoint unions. $\mathcal{H}$ also preserve a co-algebraic structure. The action of the co-product Δ on divergent 1PI Feynman graphs Γ can be defined by

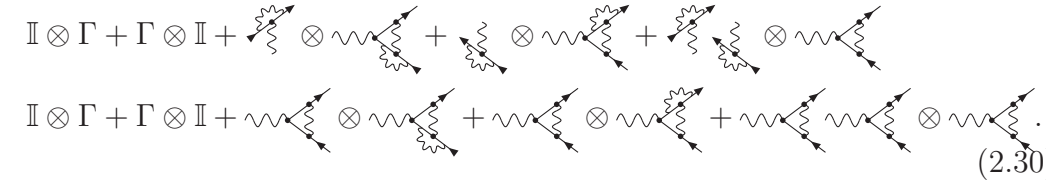
$$\Delta(\Gamma) \;=\; \mathbb{I}\otimes\Gamma + \Gamma\otimes\mathbb{I} + \sum_{\gamma\in\mathcal{P}(\Gamma)} \gamma\otimes\Gamma/\gamma. \tag{2.28}$$

Here, $\mathcal{P}(\Gamma)$ denotes the set of all proper sub-graphs γ of Γ, s.t. $\gamma = \prod_i \gamma_i$, where γ_i are divergent 1PI sub-graphs of Γ and the product is the disjoint union already introduced in $(\mathcal{H}, m, \mathbb{I})$. The word 'proper' excludes the case $\gamma = \Gamma$. Furthermore, Γ/γ denotes the Feynman graph Γ with the sub-graph γ shrunken to one vertex. This may need some explanation. Let $\Gamma =$  . Therefore,



(2.29)

and the co-product acting on Γ yields



(2.30)

In general, not every proper sub-graph of Γ needs to be divergent. Consider for example the graph $\gamma =$  . It is a proper sub-graph of Γ but applying the Feynman rules to γ yields a convergent integral. One already notices that something is wrong by looking at the residue of γ. It is $r(\gamma) =$  which is not contained in the set of Q.E.D. residues (Eq. (2.1)). In fact, the condition that only divergent sub-graphs enter $\mathcal{P}(\Gamma)$ ensures that $\Delta : \mathcal{H} \to \mathcal{H}\otimes\mathcal{H}$ in any renormalizable theory. This includes that Γ/γ is a divergent 1PI Feynman graph with a residue contained in $\mathcal{R}$ for any divergent 1PI graph γ. At this point, there is still the freedom to define the action of the co-product on the empty graph as well as on products of Feynman graphs. This will be chosen, s.t. Eqs. (2.11, 2.12) are fulfilled and $\mathcal{H}$ may achieve a bi-algebraic structure. This leads to

Proposition 2.4.1. *The co-product defined by Eqs. (2.11, 2.12, 2.28) is co-associative.*

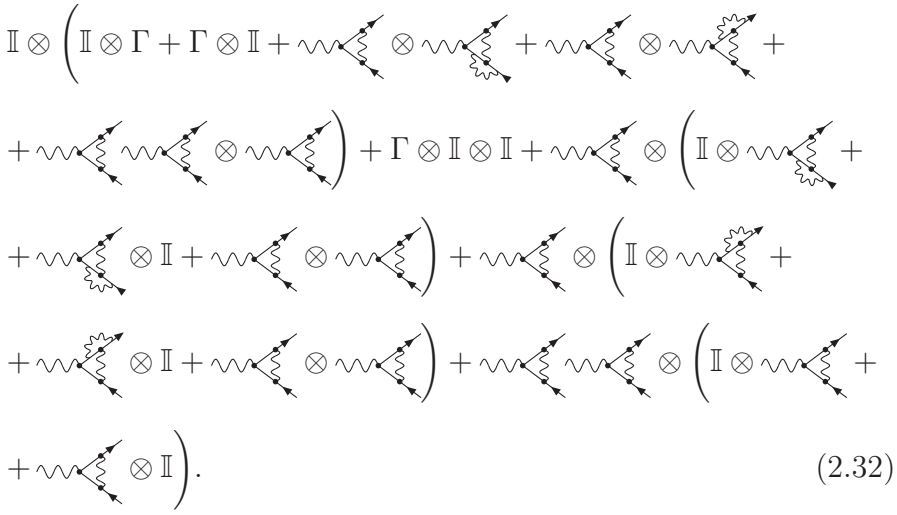
Proof. Eq. (2.8) needs to be checked. Let Γ be a divergent 1PI Feynman graph and assume that $\mathcal{P}(\Gamma)$ does not contain any product of sub-graphs. Then,

$$\begin{aligned}
(\mathrm{id}\otimes\Delta-\Delta\otimes\mathrm{id})\circ\Delta(\Gamma) &= (\mathrm{id}\otimes\Delta-\Delta\otimes\mathrm{id})\left(\mathbb{I}\otimes\Gamma+\Gamma\otimes\mathbb{I}+\sum_{\gamma\in\mathcal{P}(\Gamma)}\gamma\otimes\Gamma/\gamma\right)\\
&= \mathbb{I}\otimes\left(\mathbb{I}\otimes\Gamma+\Gamma\otimes\mathbb{I}+\sum_{\gamma\in\mathcal{P}(\Gamma)}\gamma\otimes\Gamma/\gamma\right)+\Gamma\otimes\mathbb{I}\otimes\mathbb{I}+\\
&\quad+\sum_{\gamma\in\mathcal{P}(\Gamma)}\gamma\otimes\left(\mathbb{I}\otimes\Gamma/\gamma+\Gamma/\gamma\otimes\mathbb{I}+\sum_{\beta\in\mathcal{P}(\Gamma/\gamma)}\beta\otimes(\Gamma/\gamma)/\beta\right)-\\
&\quad-\mathbb{I}\otimes\mathbb{I}\otimes\Gamma-\left(\mathbb{I}\otimes\Gamma+\Gamma\otimes\mathbb{I}+\sum_{\gamma\in\mathcal{P}(\Gamma)}\gamma\otimes\Gamma/\gamma\right)\otimes\mathbb{I}-\\
&\quad-\sum_{\gamma\in\mathcal{P}(\Gamma)}\left(\mathbb{I}\otimes\gamma+\gamma\otimes\mathbb{I}+\sum_{\beta\in\mathcal{P}(\gamma)}\beta\otimes\gamma/\beta\right)\otimes\Gamma/\gamma\\
&= \sum_{\gamma\in\mathcal{P}(\Gamma)}\sum_{\beta\in\mathcal{P}(\Gamma/\gamma)}\gamma\otimes\beta\otimes(\Gamma/\gamma)/\beta-\sum_{\gamma\in\mathcal{P}(\Gamma)}\sum_{\beta\in\mathcal{P}(\gamma)}\beta\otimes\gamma/\beta\otimes\Gamma/\gamma\\
&= 0, \qquad (2.31)
\end{aligned}$$

where the last equation follows directly by effecting the sums and rearranging the occurring terms. The generalization for arbitrary sets $\mathcal{P}(\Gamma)$ and for products of Feynman graphs is straightforward by use of Eq. (2.11). □

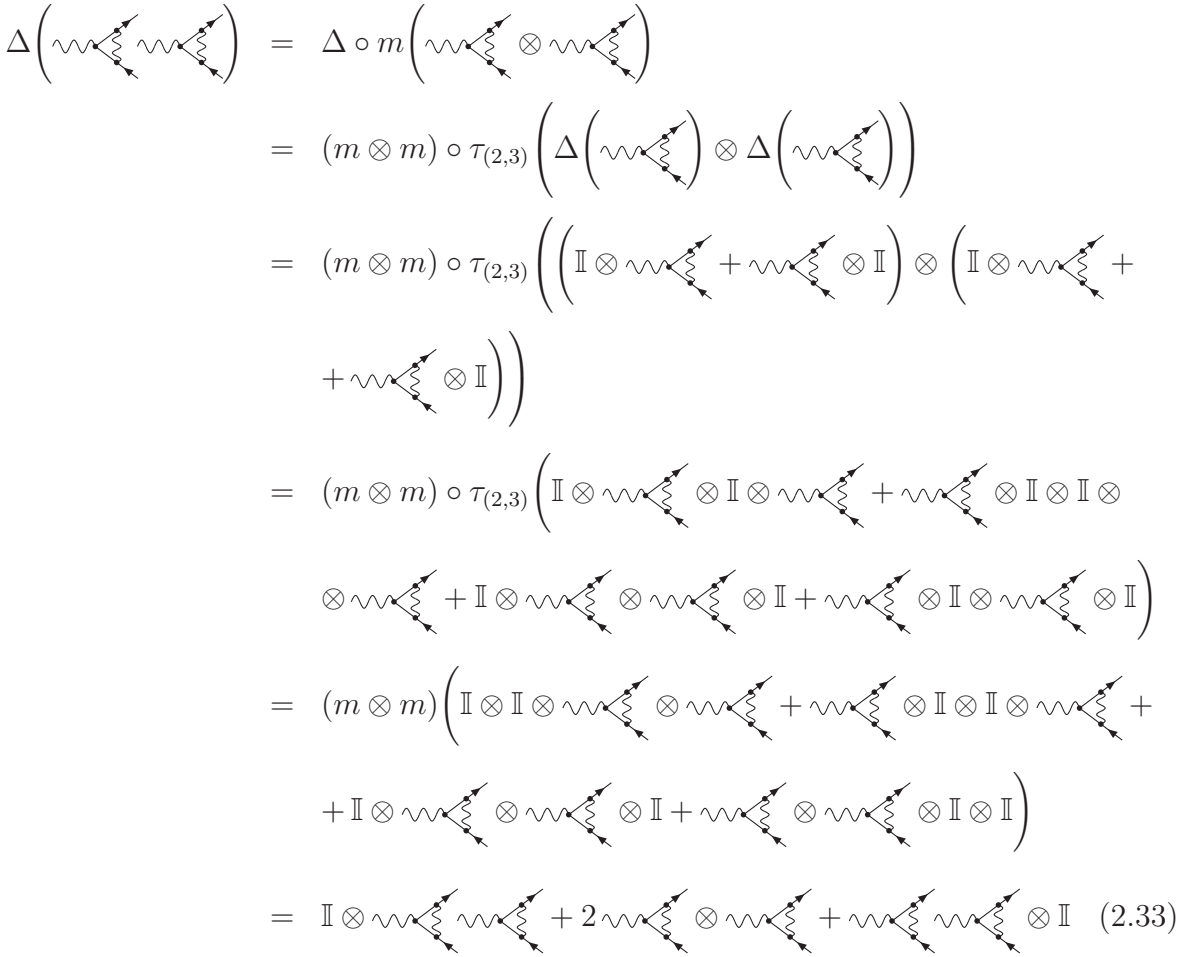
In order to make Proposition 2.4.1 and its proof more clear, consider for example the graph Γ above, whose co-product is given by Eq. (2.30). Therefore,

$(\mathrm{id}\otimes\Delta)\circ\Delta(\Gamma) =$



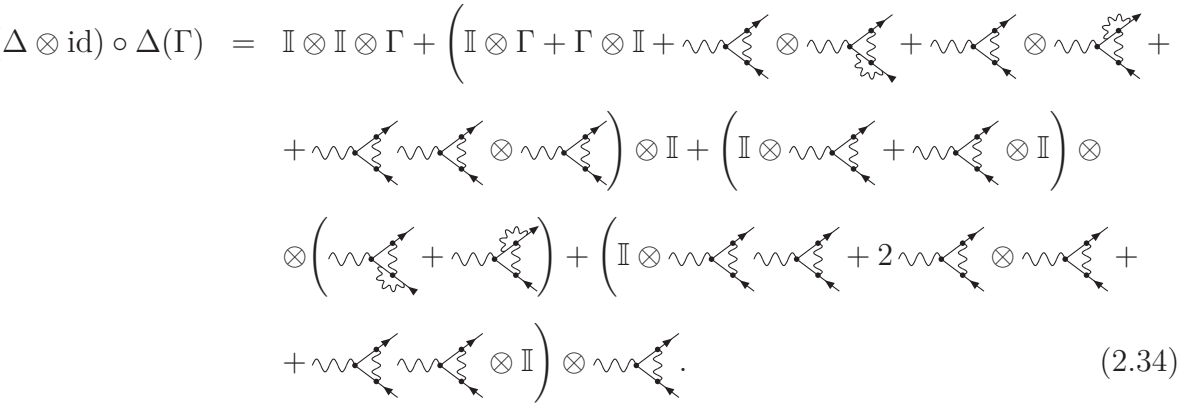
(2.32)

On the other hand, using Eq. (2.11) yields



$$
\begin{aligned}
\Delta\Big(\quad \Big) &= \Delta \circ m\Big(\quad \otimes \quad \Big) \\
&= (m \otimes m) \circ \tau_{(2,3)}\Big(\Delta\Big(\quad \Big) \otimes \Delta\Big(\quad \Big)\Big) \\
&= (m \otimes m) \circ \tau_{(2,3)}\Big(\Big(\mathbb{I} \otimes \quad + \quad \otimes \mathbb{I}\Big) \otimes \Big(\mathbb{I} \otimes \quad + \\
&\quad + \quad \otimes \mathbb{I}\Big)\Big) \\
&= (m \otimes m) \circ \tau_{(2,3)}\Big(\mathbb{I} \otimes \quad \otimes \mathbb{I} \otimes \quad + \quad \otimes \mathbb{I} \otimes \mathbb{I} \otimes \\
&\quad \otimes \quad + \mathbb{I} \otimes \quad \otimes \quad \otimes \mathbb{I} + \quad \otimes \mathbb{I} \otimes \quad \otimes \mathbb{I}\Big) \\
&= (m \otimes m)\Big(\mathbb{I} \otimes \mathbb{I} \otimes \quad \otimes \quad + \quad \otimes \mathbb{I} \otimes \mathbb{I} \otimes \quad + \\
&\quad + \mathbb{I} \otimes \quad \otimes \quad \otimes \mathbb{I} + \quad \otimes \quad \otimes \mathbb{I} \otimes \mathbb{I}\Big) \\
&= \mathbb{I} \otimes \quad + 2 \quad \otimes \quad + \quad \otimes \mathbb{I} \qquad (2.33)
\end{aligned}
$$

and thus,



$$
\begin{aligned}
(\Delta \otimes \mathrm{id}) \circ \Delta(\Gamma) &= \mathbb{I} \otimes \mathbb{I} \otimes \Gamma + \Big(\mathbb{I} \otimes \Gamma + \Gamma \otimes \mathbb{I} + \quad \otimes \quad + \quad \otimes \quad + \\
&\quad + \quad \otimes \quad \Big) \otimes \mathbb{I} + \Big(\mathbb{I} \otimes \quad + \quad \otimes \mathbb{I}\Big) \otimes \\
&\quad \otimes \Big(\quad + \quad \Big) + \Big(\mathbb{I} \otimes \quad + 2 \quad \otimes \quad + \\
&\quad + \quad \otimes \mathbb{I}\Big) \otimes \quad . \qquad (2.34)
\end{aligned}
$$

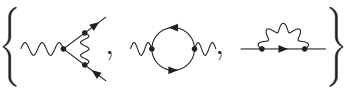
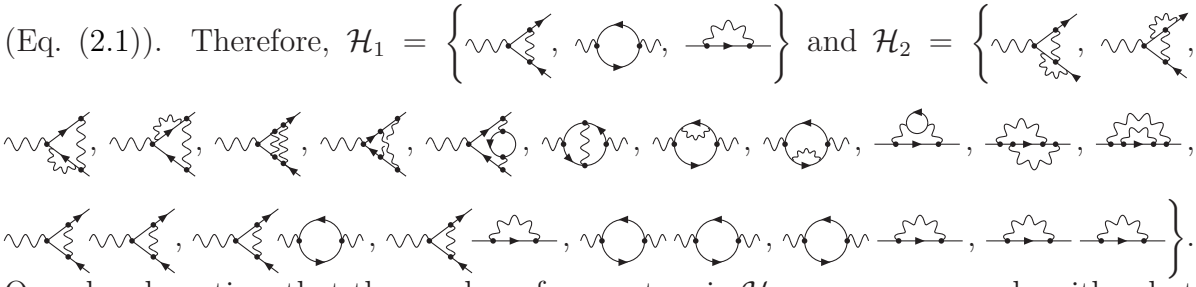
Easily can be seen that the R.H.S.s of Eqs. (2.32, 2.34) are the same. Thus, Δ is co-associative in this example. Now, the action of the co-unit map $\hat{\mathbb{I}}$ on any $\Gamma \in \mathcal{H}$ can be

defined by

$$\hat{\mathbb{I}}(\Gamma) = \begin{cases} 1, & \Gamma = \mathbb{I} \\ 0, & \text{else} \end{cases}. \tag{2.35}$$

It is trivial to show that this definition and the definitions of the co-product in Eqs. (2.11, 2.12, 2.28) fulfill Eq. (2.9). Together with the co-associativity of Δ shown in Proposition 2.4.1, $(\mathcal{H}, \Delta, \hat{\mathbb{I}})$ forms a co-algebra. Furthermore, the definitions of m, $\mathbb{I}$, Δ and $\hat{\mathbb{I}}$ include Eqs. (2.11, 2.12, 2.32). Thus, $(\mathcal{H}, m, \Delta, \mathbb{I}, \hat{\mathbb{I}})$ forms a bi-algebra as well. In order to define the Hopf algebra of Feynman graphs, only an antipode S needs to be introduced. Inserting the definitions of m, Δ, $\mathbb{I}$ and $\hat{\mathbb{I}}$ into Eq. (2.20) directly yields a recursive definition of S, i.e.

$$S(\mathbb{I}) = \mathbb{I}, \quad S(\Gamma) = -\Gamma - \sum_{\gamma \in \mathcal{P}(\Gamma)} S(\gamma)\, \Gamma/\gamma, \tag{2.36}$$

which, of course, exist. Its action on products of Feynman graphs is determined by Eq. (2.21). This work does not spend more time to investigate the antipode. For more details, see [4]. However, the necessary fact is that $(\mathcal{H}, m, \Delta, \mathbb{I}, \hat{\mathbb{I}}, S)$ forms a Hopf algebra. This Hopf algebra of Feynman graphs was discovered in 1999 [4]. Finally, the Hopf algebra of Feynman graphs admits a graduation as defined in Definition 2.3.7 by the First Betti number of Feynman graphs. This is meant, s.t. $\mathcal{H}_0 = \{\mathbb{I}\}$ and $\mathcal{H}_n$ is the vector space of divergent 1PI n-loop Feynman graphs and products of Feynman graphs with overall loop number n. For example, consider the set of residues in Q.E.D. (Eq. (2.1)). Therefore, $\mathcal{H}_1 =$  and $\mathcal{H}_2 =$  .
One already notices that the number of generators in $\mathcal{H}_n$ grows enormously with n but as can be easily seen, the graduation properties Eqs. (2.23, 2.24, 2.25) are automatically fulfilled. Thus, many properties related to renormalization can be inductively proven. Once, the Hopf algebra of Feynman graphs is introduced, the next section establishes the last important mathematical structure that is needed for renormalization of quantum field theories, namely the grafting operators B_+.

2.5. The grafting operators B_+

This section introduces the grafting operators B_+, in order to handle the sub-divergences of 1PI Feynman graphs mathematically. First, even at high loop order there exist divergent Feynman graphs without sub-divergences, such as the Q.E.D. 2-loop graph  . Thus, a useful concept is

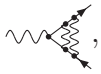
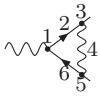
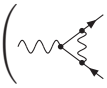
Definition 2.5.1. *The reduced co-product of a Hopf algebra is given by*

$$\tilde{\Delta}(\Gamma) = \Delta(\Gamma) - \mathbb{I} \otimes \Gamma - \Gamma \otimes \mathbb{I}. \tag{2.37}$$

If a Hopf algebra element Γ *fulfills* $\tilde{\Delta}(\Gamma) = 0$*, then, it is called primitive.*

Lemma 2.5.2. *In the Hopf algebra of Feynman graphs, all graphs without sub-divergences are primitive.*

Proof. Iff Γ has no sub-divergences, then, $\mathcal{P}(\Gamma) = \{\emptyset\}$. A closer look at the definition of the co-product in Eq. (2.28) finally gives $\tilde{\Delta}(\Gamma) = 0$. □

With these definitions it becomes clear that each 1PI divergent Feynman graph is built out of primitive ones. For example, $\Gamma =$  consists of the outer divergence $\Gamma_1 =$  and the sub-divergence $\Gamma_2 =$  . Then, the only thing left is a rule, how to build Γ out of the primitive graphs Γ_1 and Γ_2. Therefore, a family of grafting operators B_+^Γ can be introduced, where Γ denotes any primitive graph of $\mathcal{H}$. Arguments of these operators are (products of) Hopf algebra elements, and the action is insertion of the argument (or all elements of the possibly occurring product) into the primitive graph Γ, e.g. $B_+^{\Gamma_1}(\Gamma_2) = \Gamma$. The action on the unit is defined to be $B_+^{\Gamma_1}(\mathbb{I}) = \Gamma_1$. But this is not sufficient to describe the sub-divergences in a unique way. For example, the graph Γ_1 has three insertion places for Γ_1 itself. Thus, $B_+^{\Gamma_1}(\Gamma_1)$ could be  , as well as  or  . Furthermore, for $\Gamma_3 =$  , there are six possibilities for $B_+^{\Gamma_1}(\Gamma_1\Gamma_3)$. A possible solution to this dilemma is to label the insertion places of the primitive graph Γ and to assign a label according to the respective insertion place to the argument of B_+^Γ (or to each element of the possibly occurring product). Write, for example, $\Gamma_1 =$  . Therefore, $B_+^{\Gamma_1}\left(m\left(\right.\right.$  $3 \otimes$  $\left.\left.6\right)\right) =$  . The product m is meant over the usual vector space $\mathcal{H}$, the assigned numbers are to remember, where the Feynman graphs have to be inserted. Of course, the B_+^Γ operators can be recursively used. Together with the labels, it is possible to write each divergent 1PI Feynman graph and thus, each element of the Hopf algebra in terms of primitive ones and the grafting operators. This becomes very important in the next section, which discusses the equations of motion of Feynman amplitudes, namely Dyson-Schwinger equations, using the grafting operators and the primitive graphs of a quantum field theory. However, there is one last important property of the B_+^Γ operators introduced above, which is discussed in

Proposition 2.5.3. *The grafting operators B_+^Γ of the Hopf algebra of Feynman graphs fulfill*

$$\Delta \circ B_+^\Gamma(\cdot) = B_+^\Gamma(\cdot) \otimes \mathbb{I} + \left(\mathrm{id} \otimes B_+^\Gamma\right) \circ \Delta(\cdot), \tag{2.38}$$

where Γ is any primitive graph and $\cdot$ represents any element of the Hopf algebra of Feynman graphs.

Proof. Let Γ_1 be a primitive graph. Then, Eq. (2.38) yields

$$\Delta \circ B_+^{\Gamma_1}(\mathbb{I}) = \Gamma_1 \otimes \mathbb{I} + \mathbb{I} \otimes \Gamma_1, \tag{2.39}$$

which is true, because Γ_1 is primitive. Let, furthermore, Γ_2 be any 1PI Feynman graph, then,

$$\begin{aligned}
\Delta \circ B_+^{\Gamma_1}(\Gamma_2) &= \mathbb{I} \otimes B_+^{\Gamma_1}(\Gamma_2) + B_+^{\Gamma_1}(\Gamma_2) \otimes \mathbb{I} + \sum_{\gamma \in \mathcal{P}\left(B_+^{\Gamma_1}(\Gamma_2)\right)} \gamma \otimes B_+^{\Gamma_1}(\Gamma_2)/\gamma \\
&= \mathbb{I} \otimes B_+^{\Gamma_1}(\Gamma_2) + B_+^{\Gamma_1}(\Gamma_2) \otimes \mathbb{I} + \sum_{\gamma \in \{\Gamma_2\} \cup \mathcal{P}(\Gamma_2)} \gamma \otimes B_+^{\Gamma_1}(\Gamma_2)/\gamma \\
&= \mathbb{I} \otimes B_+^{\Gamma_1}(\Gamma_2) + B_+^{\Gamma_1}(\Gamma_2) \otimes \mathbb{I} + \Gamma_2 \otimes B_+^{\Gamma_1}(\Gamma_2)/\Gamma_2 + \sum_{\gamma \in \mathcal{P}(\Gamma_2)} \gamma \otimes B_+^{\Gamma_1}(\Gamma_2)/\gamma \\
&= B_+^{\Gamma_1}(\Gamma_2) \otimes \mathbb{I} + \mathbb{I} \otimes B_+^{\Gamma_1}(\Gamma_2) + \Gamma_2 \otimes \Gamma_1 + \sum_{\gamma \in \mathcal{P}(\Gamma_2)} \gamma \otimes B_+^{\Gamma_1}(\Gamma_2/\gamma) \\
&= B_+^{\Gamma_1}(\Gamma_2) \otimes \mathbb{I} + \left(\mathrm{id} \otimes B_+^{\Gamma_1}\right) \left(\mathbb{I} \otimes \Gamma_2 + \Gamma_2 \otimes \mathbb{I} + \sum_{\gamma \in \mathcal{P}(\Gamma_2)} \gamma \otimes \Gamma_2/\gamma\right) \\
&= B_+^{\Gamma_1}(\Gamma_2) \otimes \mathbb{I} + \left(\mathrm{id} \otimes B_+^{\Gamma_1}\right) \circ \Delta(\Gamma_2).
\end{aligned} \tag{2.40}$$

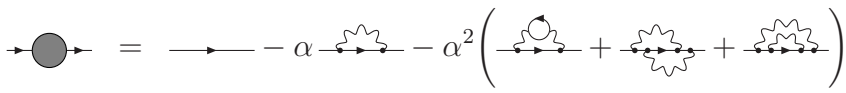
The generalization to products of Feynman graphs is straightforward by use of Eq. (2.11). □

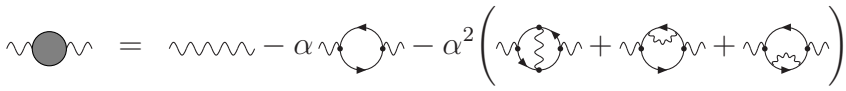
Operators fulfilling Eq. (2.38) are called Hochschild one-co-cycles. The global setting to this is Hochschild co-homology, but it would go beyond the main topic of the underlying work to go into the details of Hochschild co-homology. However, the one-co-cycle property of the grafting operators discussed in Proposition 2.5.3 turns out to be very convenient in the next section, as well as in Chapter 3, where it is used to show that there exists a Hopf algebra morphism from the Hopf algebra of Feynman graphs $\mathcal{H}$ to the Hopf algebra of words. The next section introduces Dyson-Schwinger equations.

2.6. Dyson-Schwinger equations for Feynman graphs

Dyson-Schwinger equations are fix-point equations for Feynman amplitudes on the level of Feynman graphs. For example, there exists a fix-point equation, whose solution is given by Eq. (2.4). In other words, the solution of a Dyson-Schwinger equation is the

sum of all 1PI Feynman graphs with a given residue for a given quantum field theory. The point is that every physical Dyson-Schwinger equation can be solved iteratively. Thus, instead of writing down the full sum of all 1PI Feynman graphs with a given residue (which is in fact, impossible), it is more convenient to investigate the structure of the respective Dyson-Schwinger equation. Therefore, the knowledge of Hopf algebras and grafting operators and their properties is needed, which was already discussed in the previous sections. This section discusses the Dyson-Schwinger equations of Q.E.D. Therefore, consider the sums of all 1PI Feynman graphs with residues ∿∿ and →, which will also be denoted by a blob,

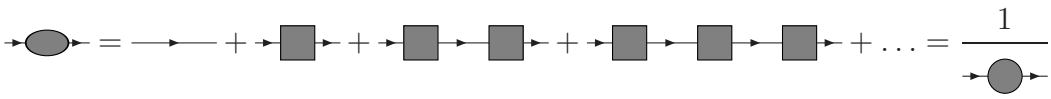

$$= - \alpha - \alpha^2 \left(+ + \right) + \mathcal{O}\left(\alpha^3\right), \quad (2.41)$$


$$= - \alpha - \alpha^2 \left(+ + \right) + \mathcal{O}\left(\alpha^3\right). \quad (2.42)$$

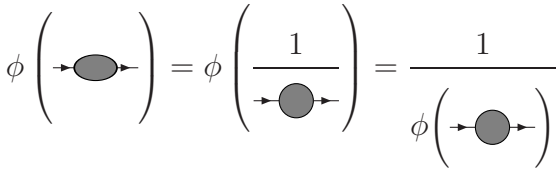
The occurring minus sign is a convention but it turns out to be very convenient. As introduced in Proposition 2.1.2, the probability amplitude of a physical process is proportional to the renormalized Feynman amplitude of the sum of all Feynman graphs with the respective residue. With the introduction of the square-shaped blob


$$= - , \quad (2.43)$$

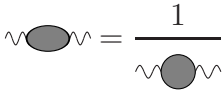
it is clear that the sum of all Feynman graphs (not necessarily 1PI) with residue → is given by


$$= + + + + \ldots = \frac{1}{\quad}, \quad (2.44)$$

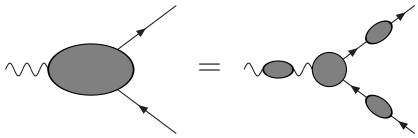
which is denoted by an egg-shaped blob. The last equation abbreviates the geometric series [6]. In fact, Feynman rules are character-like, which means that the abbreviation coincide with Feynman rules,


$$\phi\left(\quad\right) = \phi\left(\frac{1}{\quad}\right) = \frac{1}{\phi\left(\quad\right)}. \quad (2.45)$$

In full analogy, the sum of all Feynman graphs with residue ∿∿ is given by the egg-shaped blob


$$= \frac{1}{\quad}. \quad (2.46)$$

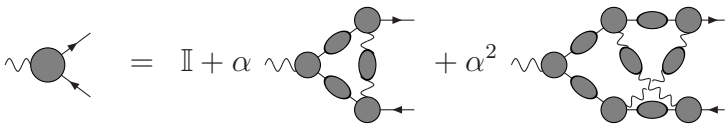
A more subtle point is that the sum of all vertex graphs, also denoted by an egg-shaped blob, can be written without any more vertex structures. It is



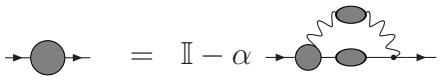
(2.47)

Finally, Eqs. (2.44, 2.46, 2.47) explain the relation between the sum of all Feynman graphs with a certain residue (egg-shaped blob) and the sum of all 1PI Feynman graphs with the respective residue (normal blobs). This was already noted in Proposition 2.1.2, but here, it becomes more clear. Using the character property of Feynman rules, each renormalized Feynman amplitude of the sum of all Feynman graphs with a certain residue is related to the renormalized Feynman amplitudes of each sum of all 1PI Feynman graphs with a particular residue in a simple way. This is the reason, why it is more convenient to investigate the sums of all 1PI Feynman graphs with a certain residue, e.g. in Q.E.D.,  ,  and  , instead of calculating the sums of all Feynman graphs with a certain residue, e.g. in Q.E.D.,  ,  and  . Now, before dealing with Dyson-Schwinger equations, observe that the occurring 0-loop graphs in Eqs. (2.4, 2.41, 2.42) are not elements of the Hopf algebra of Feynman graphs $\mathcal{H}$, because these are not divergent. However, they can be replaced by the unit $\mathbb{I}$ of $\mathcal{H}$, s.t. computing $\phi(\mathbb{I})$ gives the 0-loop Feynman amplitude of the respective residue. In connection, it is always clear, which amplitude is meant. Thus, each blob represents an element of $\mathcal{H}$. With all this knowledge, the blobs can be written as solutions of a system of Dyson-Schwinger equations, as described in

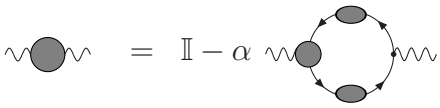
Proposition 2.6.1. *The system of Q.E.D. Dyson-Schwinger equations*



(2.48)



(2.49)

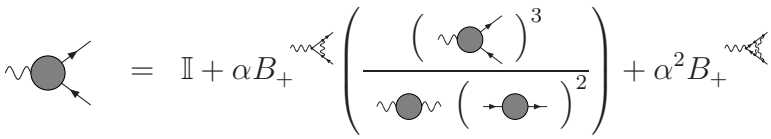


(2.50)

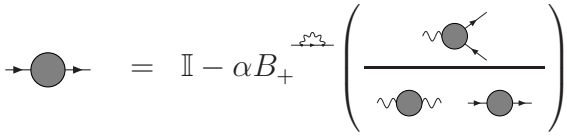
has a unique solution given by Eqs. (2.4, 2.41, 2.42) and thus, by the sums of all 1PI Feynman graphs with a particular residue.

For a proof, see [9, 10]. The reader may check that Proposition 2.6.1 is true at least, for the first loop orders. The essential property of Eqs. (2.48, 2.49, 2.50) is that only primitive graphs occur as outer divergences. This preserves that no Feynman graph is counted twice in the sums. As a matter of fact, there is only one 1-loop primitive in each propagator Dyson-Schwinger equation (Eqs. (2.49, 2.50)). On the other hand, there exists an infinite number of primitive vertex graphs in Eq. (2.48). Finally, Eqs. (2.48, 2.49, 2.50) can be written in a more compact form using the grafting operators introduced in Section 2.5, which is summarized in

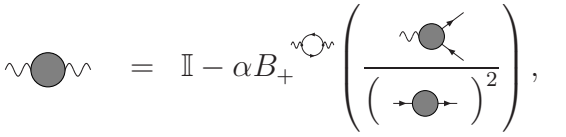
Proposition 2.6.2. *The system of Dyson-Schwinger equations introduced in Proposition 2.6.1 is equivalent to*



(2.51)



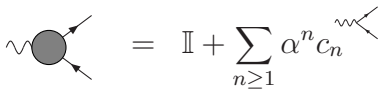
(2.52)



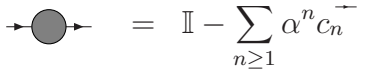
(2.53)

where the labels for the insertion places were dropped without loss of generality.

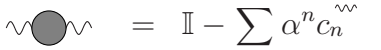
Proposition 2.6.2 follows immediately from Proposition 2.6.1. The possibility to drop the labels in the grafting operators is a consequence of inserting sums of all Feynman graphs with a certain residue. Terms occurring in the denominator of any B_+-argument have to be interpreted as the geometric series, which makes sense, because of the linearity of the B_+ operators. All occurring products are products in the Hopf algebra of Feynman graphs $\mathcal{H}$. The advantage of physical Dyson-Schwinger equations is in the structure of the coefficients of the solutions. Denote c_n^r as the sum of all divergent 1PI Feynman graphs with loop number n and residue r, e.g. in Q.E.D.,



(2.54)



(2.55)



(2.56)

Then, the c_n^r generate a sub Hopf algebra [9, 10]. The proof relies mainly on the one-co-cycle property of the grafting operators (Eq. (2.38)) and as turns out, the mentioned

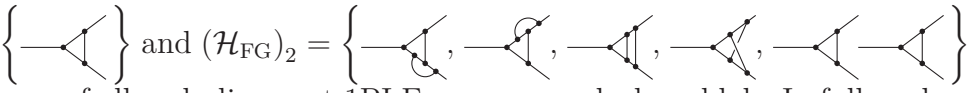
sub Hopf algebra always exists in physical quantum field theories. This is very useful in renormalization and ensures that all the c_n^r can be iteratively computed, which is very practical for the calculation of the Feynman amplitudes of the sums given in Eqs. (2.54, 2.55, 2.56). Because of the linearity of renormalized Feynman rules, it is sufficient to calculate $\phi_R\left(c_n^r\right)$ for each loop order n and given residue r. Finally, as discussed in Section 2.2, the Feynman amplitude of a full sum of divergent 1PI Feynman graphs with a certain residue yields a polynomial in the external scale parameter L (Eq. (2.5)). The arising question is now, which part of each coefficient c_n^r for a given residue r maps to which power of L in Eq. (2.5) by application of renormalized Feynman rules ϕ_R. This was meant by 'investigating the structure of Eq. (2.5)' at the end of Section 2.2. The answer to this question is given in the next chapter and turns out to be very useful for further work in renormalization [11]. However, for simplification reasons, this work will not investigate the coefficients c_n^r of the Q.E.D.-system in Eqs. (2.54, 2.55, 2.56). Instead, a single non-linear Dyson-Schwinger equation is considered. Its coefficients c_n do not have the r-index, because there will be only one residue. The procedure to divide c_n into those parts, which map into a certain power of L by applying renormalized Feynman rules ϕ_R is called filtration and will be explained in the next chapter. Then, if the filtration was done for this representative Dyson-Schwinger equation and the coefficients c_n of the solution, it can be applied to more complicated equations or systems of Dyson-Schwinger equations like the Q.E.D.-system given in Proposition 2.6.2 in future work.

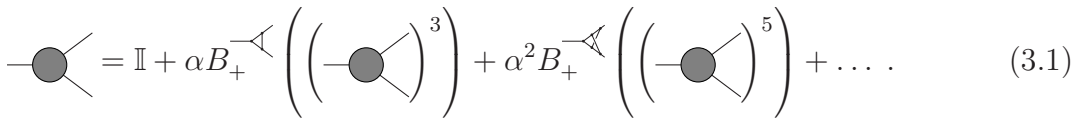
3. Filtrations in Dyson-Schwinger equations

This chapter is divided into three sections. First, Section 3.1 derives the considered representative Dyson-Schwinger equation from the Hopf algebra of Feynman graphs and a recursive formula for the coefficients c_n of its solution. Second, Section 3.2 introduces the Hopf algebra of words, which is the essential mathematical structure for the filtration of the coefficients c_n. Moreover, it will be shown that there exists a map Φ with $\Phi(c_n) = w_n$, where w_n is an element of the Hopf algebra of words. Then, renormalized Feynman rules can be applied to the words w_n as $\phi_R \circ \Phi^{-1}$ in order to calculate physical probability amplitudes. Finally, Section 3.3 explains in detail, how to filtrate the words w_n, s.t. each filtrated part of w_n maps to a certain power of the external scale parameter L (Eq. (2.5)) in the Feynman amplitude $\phi_R \circ \Phi^{-1}(w_n)$.

3.1. Derivation of the considered representative Dyson-Schwinger equation

3.1.1. Motivation: Scalar ϕ^3-theory

Consider a quantum field theory with only one kind of edge — and two residues $\mathcal{R} = \{$  $\}$. This theory is known as scalar ϕ^3-theory. Furthermore, assume that no propagator corrections are taken into account. This means that all 1PI vertex graphs with no propagator insertions together with their disjoint unions as well as the empty graph $\mathbb{I}$ generate a Hopf algebra $\mathcal{H}_{\mathrm{FG}}$, which will be called Hopf algebra of Feynman graphs as well. The definitions for the product m, the co-product Δ, the co-unit $\hat{\mathbb{I}}$, the antipode S and the one-co-cycles B_+^Γ, made in Sections 2.4-2.5 remain valid. $\mathcal{H}_{\mathrm{FG}}$ also admits a graduation by the first Betti number of graphs, s.t. $(\mathcal{H}_{\mathrm{FG}})_0 = \{\mathbb{I}\}$, $(\mathcal{H}_{\mathrm{FG}})_1 =$ and $(\mathcal{H}_{\mathrm{FG}})_2 =$  etc. Denote the sum of all such divergent 1PI Feynman graphs by a blob. In full analogy to the previous chapter, it is the solution of the Dyson-Schwinger equation



$$= \mathbb{I} + \alpha B_+ \left(\left(\quad \right)^3 \right) + \alpha^2 B_+ \left(\left(\quad \right)^5 \right) + \dots \,. \tag{3.1}$$

The purpose of this work is to solve Eq. (3.1) by the Ansatz

$$= \mathbb{I} + \sum_n \alpha^n c_n^{\prec} \tag{3.2}$$

and to compute the coefficients $c_n^{\prec}$. Furthermore, as discussed in Section 2.2, applying renormalized Feynman rules ϕ_R to Eq. (3.2) yields a polynomial in the external scale parameter L (Eq. (2.5)). Thus, the interesting question is, which part of each $c_n^{\prec}$ maps to which power of L by application of renormalized Feynman rules ϕ_R.

3.1.2. Introduction of the analyzed Dyson-Schwinger equation

Eq. (3.1) gives rise to another, slightly different Dyson-Schwinger equation. Therefore, let $\left(\mathcal{H}, m, \Delta, \mathbb{I}, \hat{\mathbb{I}}, S\right)$ be a graded Hopf algebra, $\mathcal{H} = \bigoplus_{n\in\mathbb{N}_0} \mathcal{H}_n$, s.t. there is exactly one primitive element Γ_n in each $\mathcal{H}_n$. Furthermore, let $B_+^{\Gamma_n}$, $n \in \mathbb{N}$ be a collection of Hochschild one-co-cycles and α a perturbative parameter. This is all abstract, but an example is the Hopf algebra of decorated rooted trees [11]. Now, consider the non-linear Dyson-Schwinger equation

$$X = \mathbb{I} + \sum_{n\geq 1} \alpha^n B_+^{\Gamma_n}\left(X^{n+1}\right), \tag{3.3}$$

whose solution is denoted by X. The product in the arguments of the $B_+^{\Gamma_n}$ operators is the Hopf algebra product m. Eq. (3.3) has a unique solution [11], given by

$$X = \sum_{n\geq 0} \alpha^n c_n, \tag{3.4}$$

where the coefficients are recursively defined to be

$$c_0 = \mathbb{I}, \quad c_n = \sum_{m=1}^{n} B_+^{\Gamma_m}\left(\sum_{k_1+\ldots k_{m+1}=n-m, k_i\geq 0} c_{k_1}\ldots c_{k_{m+1}}\right). \tag{3.5}$$

The product in the brackets is abbreviated for the Hopf algebra product m. For a proof, insert the Ansatz Eqs. (3.4, 3.5) into Eq. (3.3). It is also worth to note that the coefficients c_n generate a sub Hopf algebra of $\mathcal{H}$ [11]. In the scope of this thesis the representative Dyson-Schwinger equation in Eq. (3.3) and its solution in Eqs. (3.4, 3.5) will be studied. However, it is a generalization of Eq. (3.1). Indeed, setting all terms of an odd power of α or terms 'containing' a Γ_n for an odd number n in the solution of Eq. (3.3) to zero as well as for even n, replacing Γ_n by the sum of all primitive graphs of $\mathcal{H}_{\mathrm{FG}}$ with loop number $n/2$ and α^2 by α gives the solution to Eq. (3.1). Furthermore, it is possible to extend the physical renormalized Feynman rules ϕ_R from $\mathcal{H}_{\mathrm{FG}}$ to the abstract Hopf algebra $\mathcal{H}$, s.t. application to X still yields a polynomial in the external scale parameter L, which matches $\phi_R(X_{\mathrm{FG}})$ after transforming X to X_{FG} in the way

described above. The only requirement for the renormalized Feynman rules ϕ_R acting on $\mathcal{H}$ is that they are character-like, i.e., $\forall h_1, h_2 \in \mathcal{H}$,

$$\phi_R \circ m\left(h_1 \otimes h_2\right) = \phi_R(h_1) \cdot \phi_R(h_2), \tag{3.6}$$

where the product $\cdot$ on the R.H.S. of Eq. (3.6) is usual multiplication. In the following, the coefficients $c_n \in \mathcal{H}$ of the solution of the Dyson-Schwinger equation in Eq. (3.3) will be filtrated, s.t. each term of the filtration maps to a certain power of L by application of renormalized Feynman rules $\phi_R\left(c_n\right)$. The extension to the solution of Eq. (3.1) is then, straightforward using the method described above. Furthermore, the following filtration can be applied to any other Dyson-Schwinger equation of a physical quantum field theory.

3.2. The Hopf algebra of words

The filtration of the coefficients $c_n \in \mathcal{H}$ does not need to be directly possible. Therefore, this section introduces the Hopf algebra of words $\mathcal{H}_W$. Let $\mathcal{H}_W$ be the vector space of words, $\mathcal{H}_L \subset \mathcal{H}_W$ the subspace of letters and $p : \mathcal{H}_L \otimes \mathcal{H}_L \to \mathcal{H}_L$ a commutative and associative map, which assigns a new letter for any two given letters. Commutativity and associativity mean that $\forall l_1, l_2, l_3 \in \mathcal{H}_L$,

$$p\left(l_1, l_2\right) = p\left(l_2, l_1\right), \quad p\left(l_1, p\left(l_2, l_3\right)\right) = p\left(p\left(l_1, l_2\right), l_3\right) =: p\left(l_1, l_2, l_3\right). \tag{3.7}$$

It is clear that, given any two words u_1 and u_2, it is always possible to create a new word u_3 by concatenating u_1 and u_2, which means $u_3 = u_1 u_2$. In the following, concatenated words never abbreviate any product. It is important to have this in mind in the following

Definition 3.2.1. *Denote the empty word by $\mathbb{I}_W$. Then, a product $m_W : \mathcal{H}_W \otimes \mathcal{H}_W \to \mathcal{H}_W$ can be recursively defined, s.t. $\forall l_1, l_2 \in \mathcal{H}_L, u_1, u_2, u \in \mathcal{H}_W$,*

$$\begin{aligned} m_W\left(l_1 u_1 \otimes l_2 u_2\right) &= l_1 m_W\left(u_1 \otimes l_2 u_2\right) + l_2 m_W\left(l_1 u_1 \otimes u_2\right) + \\ &\quad + p(l_1, l_2) m_W\left(u_1 \otimes u_2\right) && (3.8) \\ m_W(\mathbb{I}_W \otimes u) &= m_W(u \otimes \mathbb{I}_W) = u. && (3.9) \end{aligned}$$

This product is called shuffle product and for better readability, it is sometimes abbreviated by

$$m_W\left(u_1 \otimes u_2\right) =: u_1 \sqcup\!\sqcup u_2 \tag{3.10}$$

for any two words u_1 and u_2. In the following, both notations are used dependent on which one is more convenient.

For example, let $l_1, l_2, l_3 \in \mathcal{H}_L$. Then,

$$\begin{aligned} l_1 \sqcup\!\sqcup l_2 l_3 &= l_1 \mathbb{I}_W \sqcup\!\sqcup l_2 l_3 \\ &= l_1\left(\mathbb{I}_W \sqcup\!\sqcup l_2 l_3\right) + l_2\left(l_1 \mathbb{I}_W \sqcup\!\sqcup l_3 \mathbb{I}_W\right) + p\left(l_1, l_2\right)\left(\mathbb{I}_W \sqcup\!\sqcup l_3\right) \\ &= l_1 l_2 l_3 + l_2\left(l_1\left(\mathbb{I}_W \sqcup\!\sqcup l_3\right) + l_3\left(l_1 \sqcup\!\sqcup \mathbb{I}_W\right) + p\left(l_1, l_3\right)\left(\mathbb{I}_W \sqcup\!\sqcup \mathbb{I}_W\right)\right) + p\left(l_1, l_2\right) l_3 \\ &= l_1 l_2 l_3 + l_2 l_1 l_3 + l_2 l_3 l_1 + l_2 p\left(l_1, l_3\right) + p\left(l_1, l_2\right) l_3. && (3.11) \end{aligned}$$

Definition 3.2.1 assigns an algebra structure to $\mathcal{H}_W$ by

Proposition 3.2.2. $(\mathcal{H}_W, m_W, \mathbb{I}_W)$ *forms an algebra.*

Proof. Eq. (3.9) ensures that $\mathbb{I}_W$ is the neutral element of m_W, thus, Eq. (2.7) is fulfilled. It remains to show associativity (Eq. (2.6)). Therefore, commutativity of m_W is needed. This can be proved by induction over the length of words, i.e. the number of letters in a word. The induction start is given by Eq. (3.9). The induction step goes as follows. Assume that for any two words with overall length $\leq n \in \mathbb{N}_0$ the shuffle product is commutative. Then, for any $l_1, l_2 \in \mathcal{H}_L$ and $u_1, u_2 \in \mathcal{H}_W$ with overall length $n+1$,

$$\begin{aligned} l_1 u_1 \sqcup\!\sqcup l_2 u_2 &= l_1 \underbrace{(u_1 \sqcup\!\sqcup l_2 u_2)}_{\text{overall length } n} + l_2 \underbrace{(l_1 u_1 \sqcup\!\sqcup u_2)}_{\text{overall length } n} + p(l_1, l_2) \underbrace{(u_1 \sqcup\!\sqcup u_2)}_{\text{overall length } n-1} \\ &= l_2 (u_2 \sqcup\!\sqcup l_1 u_1) + l_1 (l_2 u_2 \sqcup\!\sqcup u_1) + p(l_2, l_1)(u_2 \sqcup\!\sqcup u_1) \\ &= l_2 u_2 \sqcup\!\sqcup l_1 u_1, \end{aligned} \tag{3.12}$$

where in the second equation, commutativity of p (Eq. (3.7)) was used. It is straightforward to show associativity. The induction start is again, given by Eq. (3.9). For the induction step assume that for any three words with overall length $\leq n \in \mathbb{N}_0$ the shuffle product is associative. Then, for any $l_1, l_2, l_3 \in \mathcal{H}_L$ and $u_1, u_2, u_3 \in \mathcal{H}_W$ with overall length $n+1$,

$$\begin{aligned} l_1 u_1 \sqcup\!\sqcup (l_2 u_2 \sqcup\!\sqcup l_3 u_3) &= l_1 u_1 \sqcup\!\sqcup (l_2 (u_2 \sqcup\!\sqcup l_3 u_3) + l_3 (l_2 u_2 \sqcup\!\sqcup u_3) + p(l_2, l_3)(u_2 \sqcup\!\sqcup u_3)) \\ &= l_1 (u_1 \sqcup\!\sqcup (l_2 u_2 \sqcup\!\sqcup l_3 u_3)) + l_2 (l_1 u_1 \sqcup\!\sqcup (u_2 \sqcup\!\sqcup l_3 u_3)) + \\ &\quad + l_3 (l_1 u_1 \sqcup\!\sqcup (l_2 u_2 \sqcup\!\sqcup u_3)) + p(l_2, l_3)(l_1 u_1 \sqcup\!\sqcup (u_2 \sqcup\!\sqcup u_3)) + \\ &\quad + p(l_1, l_2)(u_1 \sqcup\!\sqcup (u_2 \sqcup\!\sqcup l_3 u_3)) + p(l_1, l_3)(u_1 \sqcup\!\sqcup (l_2 u_2 \sqcup\!\sqcup u_3)) + \\ &\quad p(l_1, p(l_2, l_3))(u_1 \sqcup\!\sqcup (u_2 \sqcup\!\sqcup u_3)) \\ &= l_1 (u_1 \sqcup\!\sqcup (l_2 u_2 \sqcup\!\sqcup l_3 u_3)) + l_2 (u_2 \sqcup\!\sqcup (l_3 u_3 \sqcup\!\sqcup l_1 u_1)) + \\ &\quad + l_3 (u_3 \sqcup\!\sqcup (l_1 u_1 \sqcup\!\sqcup l_2 u_2)) + p(l_2, l_3)(l_1 u_1 \sqcup\!\sqcup (u_2 \sqcup\!\sqcup u_3)) + \\ &\quad + p(l_1, l_2)(l_3 u_3 \sqcup\!\sqcup (u_1 \sqcup\!\sqcup u_2)) + p(l_3, l_1)(l_2 u_2 \sqcup\!\sqcup (u_3 \sqcup\!\sqcup u_1)) + \\ &\quad p(l_1, l_2, l_3)(u_1 \sqcup\!\sqcup u_2 \sqcup\!\sqcup u_3), \end{aligned} \tag{3.13}$$

where in the last equation, commutativity of m_W and p as well as associativity of p and the induction hypothesis were used. In particular, the inner brackets of the shuffle product in the last line could be dropped because of the assumed associativity. Finally, the R.H.S. of Eq. (3.13) is symmetric in the pairs (l_1, u_1), (l_2, u_2), (l_3, u_3). This must also be fulfilled by the L.H.S. and thus,

$$l_1 u_1 \sqcup\!\sqcup (l_2 u_2 \sqcup\!\sqcup l_3 u_3) = l_3 u_3 \sqcup\!\sqcup (l_2 u_2 \sqcup\!\sqcup l_1 u_1) = (l_1 u_1 \sqcup\!\sqcup l_2 u_2) \sqcup\!\sqcup l_3 u_3, \tag{3.14}$$

where the last equation follows from commutativity of the shuffle product. This proves the induction step, and thus, associativity of m_W (Eq. (2.6)). □

In order to assign a co-algebraic structure to $\mathcal{H}_W$, a co-product as well as a co-unit are needed. These are introduced in

Definition 3.2.3. *The co-product Δ_W is defined as*

$$\Delta_W(u) = \sum_{u''u'=u} u' \otimes u'' \tag{3.15}$$

and the co-unit map $\hat{\mathbb{I}}_W$ is defined as

$$\hat{\mathbb{I}}_W(u) = \begin{cases} 1, & u = \mathbb{I}_W \\ 0, & \text{else} \end{cases} \tag{3.16}$$

for any word u.

For example, let $l_1, l_2, l_3 \in \mathcal{H}_L$. Then,

$$\Delta_W(l_1 l_2 l_3) = \mathbb{I}_W \otimes l_1 l_2 l_3 + l_3 \otimes l_1 l_2 + l_2 l_3 \otimes l_1 + l_1 l_2 l_3 \otimes \mathbb{I}_W. \tag{3.17}$$

In the literature, the co-product on words is usually defined as $\Delta_{\text{Lit.}} = \tau_{1,2} \circ \Delta_W$ [12] but this implicates a redefinition of the product m_W as well as different grafting operators than used later in this work. Thus, $\mathcal{H}_W$ admits a co-algebraic structure, as described in

Proposition 3.2.4. $\left(\mathcal{H}_W, \Delta_W, \hat{\mathbb{I}}_W\right)$ *forms a co-algebra.*

Proof. The co-product Δ_W is co-associative,

$$(\mathrm{id} \otimes \Delta_W) \circ \Delta_W(u) = \sum_{u'''u''u'=u} u' \otimes u'' \otimes u''' = (\Delta_W \otimes \mathrm{id}) \circ \Delta_W(u). \tag{3.18}$$

Co-unitarity is obvious. Thus, Eqs. (2.8, 2.9) are fulfilled. □

At this point, it is convenient to define the grafting operators B_+. In fact, there is no need for an underlying Hopf algebra structure in order to define them. The primitive elements of the co-algebra $(\mathcal{H}_W, \Delta_W, \mathbb{I}_W)$ are all letters $l \in \mathcal{H}_L$. This is obvious by a look at the co-product (Eq. (3.15)). Thus, the grafting operators can be defined in

Definition 3.2.5. *For any letter $l \in \mathcal{H}_L$, there is a linear map $B_+^l : \mathcal{H}_W \to \mathcal{H}_W$ which concatenates u to l, thus,*

$$B_+^l(\mathbb{I}_W) = l, \quad B_+^l(u) = lu. \tag{3.19}$$

These maps are the grafting operators of $\mathcal{H}_W$.

Proposition 3.2.6. *The grafting operators introduced in Definition 3.2.5 are Hochschild one-cycles.*

Proof. In fact,

$$\begin{aligned} \Delta_W \circ B_+^l(u) &= \sum_{u''u'=lu} u' \otimes u'' \\ &= lu \otimes \mathbb{I}_W + \left(\mathrm{id} \otimes B_+^l\right)\left(\sum_{u''u'=u} u' \otimes u''\right) \\ &= B_+^l(u) \otimes \mathbb{I}_W + \left(\mathrm{id} \otimes B_+^l\right) \circ \Delta_W(u). \end{aligned} \tag{3.20}$$

□

The reason to define the grafting operators at this point is that they can be used to rewrite the definition of the shuffle product (Eq. (3.8)) into

$$\begin{aligned} m_W\left(B_+^{l_1}(u_1)\otimes B_+^{l_2}(u_2)\right) &= B_+^{l_1}\circ m_W\left(u_1\otimes l_2u_2\right)+B_+^{l_2}\circ m_W\left(l_1u_1\otimes u_2\right)+ \\ &\quad + B_+^{p(l_1,l_2)}\circ m_W\left(u_1\otimes u_2\right). \end{aligned} \tag{3.21}$$

This turns out to be very convenient in the following

Proposition 3.2.7. $\left(\mathcal{H}_W, m_W, \Delta_W, \mathbb{I}_W, \hat{\mathbb{I}}_W\right)$ *forms a bi-algebra.*

Proof. Eqs. (2.12, 2.13) are fulfilled by the definitions of Δ_W and $\hat{\mathbb{I}}_W$ (Eqs. (3.15, 3.16)). It remains to show Eq. (2.11), which can be done by induction. For the induction start, let u be any word. Then,

$$\Delta_W\circ m_W\left(\mathbb{I}_W\otimes u\right)=\Delta_W(u)=\sum_{u''u'=u}u'\otimes u''. \tag{3.22}$$

On the other hand,

$$\begin{aligned} \left(m_W\otimes m_W\right)\circ\tau_{(2,3)}\circ\left(\Delta_W\otimes\Delta_W\right)\left(\mathbb{I}_W\otimes u\right) &= \left(m_W\otimes m_W\right)\left(\sum_{u''u'=u}\mathbb{I}_W\otimes u'\otimes\mathbb{I}_W\otimes u''\right) \\ &= \sum_{u''u'=u}u'\otimes u''. \end{aligned} \tag{3.23}$$

In full analogy,

$$\Delta_W\circ m_W\left(u\otimes\mathbb{I}_W\right)=\left(m_W\otimes m_W\right)\circ\tau_{(2,3)}\circ\left(\Delta_W\otimes\Delta_W\right)\left(u\otimes\mathbb{I}_W\right). \tag{3.24}$$

For the induction step, assume that Eq. (2.11) is fulfilled for any two words with overall length $\leq n\in\mathbb{N}_0$. Then, for any $l_1,l_2\in\mathcal{H}_L$ and $u_1,u_2\in\mathcal{H}_W$ with overall length $n+1$, the L.H.S. of Eq. (2.11) acting on $l_1u_1\otimes l_2u_2$ can be computed to be

$$\begin{aligned} \text{L.H.S.} &:= \Delta_W\circ m_W\left(B_+^{l_1}(u_1)\otimes B_+^{l_2}(u_2)\right) \\ &= \Delta_W\circ B_+^{l_1}\circ m_W\left(u_1\otimes B_+^{l_2}(u_2)\right)+\Delta_W\circ B_+^{l_2}\circ m_W\left(B_+^{l_1}(u_1)\otimes u_2\right)+ \\ &\quad +\Delta_W\circ B_+^{p(l_1,l_2)}\circ m_W\left(u_1\otimes u_2\right) \\ &= m_W\left(B_+^{l_1}(u_1)\otimes B_+^{l_2}(u_2)\right)\otimes\mathbb{I}_W+\left(\mathrm{id}\otimes B_+^{l_1}\right)\circ\Delta_W\circ m_W\left(u_1\otimes B_+^{l_2}(u_2)\right)+ \\ &\quad +\left(\mathrm{id}\otimes B_+^{l_2}\right)\circ\Delta_W\circ m_W\left(B_+^{l_1}(u_1)\otimes u_2\right)+ \\ &\quad +\left(\mathrm{id}\otimes B_+^{p(l_1,l_2)}\right)\circ\Delta_W\circ m_W\left(u_1\otimes u_2\right), \end{aligned} \tag{3.25}$$

where Eq. (3.21) was used in the second equation and the Hochschild one-co-cycle property of the grafting operators (Eq. (3.20)) was used in the third equation. On the R.H.S.

of Eq. (3.25), the induction hypothesis (Eq. (2.11)) can be used. Thus,

$$
\begin{aligned}
\text{L.H.S.} \;=\;& m_W\left(B_+^{l_1}(u_1)\otimes B_+^{l_2}(u_2)\right)\otimes \mathbb{I}_W + \\
&+\left(\mathrm{id}\otimes B_+^{l_1}\right)\circ(m_W\otimes m_W)\circ\tau_{2,3}\left(\Delta_W(u_1)\otimes\Delta_W\circ B_+^{l_2}(u_2)\right)+ \\
&+\left(\mathrm{id}\otimes B_+^{l_2}\right)\circ(m_W\otimes m_W)\circ\tau_{2,3}\left(\Delta_W\circ B_+^{l_1}(u_1)\otimes\Delta_W(u_2)\right)+ \\
&+\left(\mathrm{id}\otimes B_+^{p(l_1,l_2)}\right)\circ(m_W\otimes m_W)\circ\tau_{(2,3)}\left(\Delta_W(u_1)\otimes\Delta_W(u_2)\right) \\
=\;& m_W\left(B_+^{l_1}(u_1)\otimes B_+^{l_2}(u_2)\right)\otimes \mathbb{I}_W + \\
&+\left(\mathrm{id}\otimes B_+^{l_1}\right)\circ(m_W\otimes m_W)\circ\tau_{2,3}\left(\Delta_W(u_1)\otimes B_+^{l_2}(u_2)\otimes\mathbb{I}_W\right)+ \\
&+\left(\mathrm{id}\otimes B_+^{l_2}\right)\circ(m_W\otimes m_W)\circ\tau_{2,3}\left(B_+^{l_1}(u_1)\otimes\mathbb{I}_W\otimes\Delta_W(u_2)\right)+ \\
&+\Big(\left(\mathrm{id}\otimes B_+^{l_1}\right)\circ(m_W\otimes m_W)\circ\tau_{2,3}\circ\left(\mathrm{id}\otimes\mathrm{id}\otimes\mathrm{id}\otimes B_+^{l_2}\right)+ \\
&+\left(\mathrm{id}\otimes B_+^{l_2}\right)\circ(m_W\otimes m_W)\circ\tau_{2,3}\circ\left(\mathrm{id}\otimes B_+^{l_1}\otimes\mathrm{id}\otimes\mathrm{id}\right)+ \\
&+\left(\mathrm{id}\otimes B_+^{p(l_1,l_2)}\right)\circ(m_W\otimes m_W)\circ\tau_{(2,3)}\circ\left(\mathrm{id}^{\otimes 4}\right)\Big)\left(\Delta_W(u_1)\otimes\Delta_W(u_2)\right) \\
=\;& m_W\left(B_+^{l_1}(u_1)\otimes B_+^{l_2}(u_2)\right)\otimes \mathbb{I}_W + \sum_{u_1''u_1'=u_1} m_W\left(u_1'\otimes B_+^{l_2}(u_2)\right)\otimes \\
&\otimes B_+^{l_1}(u_1'') + \sum_{u_2''u_2'=u_2} m_W\left(B_+^{l_1}(u_1)\otimes u_2'\right)\otimes B_+^{l_2}(u_2'')+ \\
&+\sum_{u_1''u_1'=u_1}\sum_{u_2''u_2'=u_2} m_W(u_1'\otimes u_2')\otimes\Big(B_+^{l_1}\circ m_W\left(u_1''\otimes B_+^{l_2}(u_2'')\right) \\
&+B_+^{l_2}\circ m_W\left(B_+^{l_1}(u_1'')\otimes u_2''\right)+B_+^{p(l_1,l_2)}\circ m_W(u_1''\otimes u_2'')\Big) \\
=\;& m_W\left(B_+^{l_1}(u_1)\otimes B_+^{l_2}(u_2)\right)\otimes \mathbb{I}_W + \sum_{u_1''u_1'=u_1} m_W\left(u_1'\otimes B_+^{l_2}(u_2)\right)\otimes \\
&\otimes B_+^{l_1}(u_1'') + \sum_{u_2''u_2'=u_2} m_W\left(B_+^{l_1}(u_1)\otimes u_2'\right)\otimes B_+^{l_2}(u_2'')+ \\
&+\sum_{u_1''u_1'=u_1}\sum_{u_2''u_2'=u_2} m_W(u_1'\otimes u_2')\otimes m_W\left(B_+^{l_1}(u_1'')\otimes B_+^{l_2}(u_2'')\right),
\end{aligned}
\tag{3.26}
$$

where the Hochschild one-co-cycle property of the grafting operators (Eq. (3.20)) was used in the second equation and the definition of the co-product (Eq. (3.15)) as well as the definition of the product (Eq. (3.21)) were used in the third and fourth equations. On the other hand, for any $l_1, l_2 \in \mathcal{H}_L$ and $u_1, u_2 \in \mathcal{H}_W$ with overall length $n+1$, the

R.H.S. of Eq. (2.11) acting on $l_1 u_1 \otimes l_2 u_2$ can be calculated to be

$$\begin{aligned}
\text{R.H.S.} \;:=\; & (m_W \otimes m_W) \circ \tau_{2,3} \circ (\Delta_W \otimes \Delta_W) \left(B_+^{l_1}(u_1) \otimes B_+^{l_2}(u_2) \right) \\
= \; & (m_W \otimes m_W) \circ \tau_{2,3} \left(\Delta_W \circ B_+^{l_1}(u_1) \otimes \Delta_W \circ B_+^{l_2}(u_2) \right) \\
= \; & (m_W \otimes m_W) \circ \tau_{2,3} \left(B_+^{l_1}(u_1) \otimes \mathbb{I}_W \otimes B_+^{l_2}(u_2) \otimes \mathbb{I}_W \right) + \\
& + (m_W \otimes m_W) \circ \tau_{2,3} \left(\left(\left(\mathrm{id} \otimes B_+^{l_1} \right) \circ \Delta_W(u_1) \right) \otimes B_+^{l_2}(u_2) \otimes \mathbb{I}_W \right) + \\
& + (m_W \otimes m_W) \circ \tau_{2,3} \left(B_+^{l_1}(u_1) \otimes \mathbb{I}_W \otimes \left(\left(\mathrm{id} \otimes B_+^{l_2} \right) \circ \Delta_W(u_2) \right) \right) + \\
& + (m_W \otimes m_W) \circ \tau_{2,3} \circ \left(\mathrm{id} \otimes B_+^{l_1} \otimes \mathrm{id} \otimes B_+^{l_2} \right) (\Delta_W(u_1) \otimes \Delta_W(u_2)) \\
= \; & m_W \left(B_+^{l_1}(u_1) \otimes B_+^{l_2}(u_2) \right) \otimes \mathbb{I}_W + \sum_{u_1'' u_1' = u_1} m_W \left(u_1' \otimes B_+^{l_2}(u_2) \right) \otimes \\
& \otimes B_+^{l_1}(u_1'') + \sum_{u_2'' u_2' = u_2} m_W \left(B_+^{l_1}(u_1) \otimes u_2' \right) \otimes B_+^{l_2}(u_2'') + \\
& + \sum_{u_1'' u_1' = u_1} \sum_{u_2'' u_2' = u_2} m_W (u_1' \otimes u_2') \otimes m_W \left(B_+^{l_1}(u_1'') \otimes B_+^{l_2}(u_2'') \right), \qquad (3.27)
\end{aligned}$$

where Eqs. (3.20, 3.15) were used in the third and fourth equations. Comparison of Eqs. (3.26, 3.27) finally yields Eq. (2.11) for any two words with overall length $n+1$ which proves the induction step. □

In order to assign a Hopf algebra structure to the bi-algebra $\left(\mathcal{H}_W, m_W, \Delta_W, \mathbb{I}_W, \hat{\mathbb{I}}_W \right)$, an antipode S_W is needed. As discussed in Section 2.3, S_W must be chosen, s.t. it holds Eq. (2.20). Easily can be seen that this is fulfilled by the recursive definition

$$S_W(\mathbb{I}_W) = \mathbb{I}_W, \quad S_W(u) = -u - \sum_{u''u' = u,\, u', u'' \neq \mathbb{I}_W} m_W (S_W(u') \otimes u''). \qquad (3.28)$$

Thus, $\left(\mathcal{H}_W, m_W, \Delta_W, \mathbb{I}_W, \hat{\mathbb{I}}_W, S_W \right)$ finally forms a Hopf algebra, called the Hopf algebra of words [12]. The reason, why it was introduced is

Proposition 3.2.8. *$\exists$ a unique, linear, bijective map $\Phi : \mathcal{H} \to \mathcal{H}_W$ which maps $\mathbb{I}$ to $\mathbb{I}_W$,*

$$\Phi(\mathbb{I}) = \mathbb{I}_W \qquad (3.29)$$

and which respects product, co-product, co-unit and antipode,

$$\begin{aligned}
m_W \circ (\Phi \otimes \Phi) &= \Phi \circ m && (3.30)\\
\Delta_W \circ \Phi &= (\Phi \otimes \Phi) \circ \Delta && (3.31)\\
\hat{\mathbb{I}}_W \circ \Phi &= \Phi \circ \hat{\mathbb{I}} && (3.32)\\
S_W \circ \Phi &= \Phi \circ S. && (3.33)
\end{aligned}$$

Such a map Φ fulfilling Eqs. (3.29-3.33) is called Hopf algebra morphism.

The proof for this powerful proposition relies on the fact that the grading operators B_+ are Hochschild one-co-cycles in both Hopf algebras [13]. In fact, Φ can be given explicitly by Eqs. (3.29, 3.30) and the universal property of Hochschild one-co-cycles

$$B_+^{l_n} \circ \Phi = \Phi \circ B_+^{\Gamma_n}. \tag{3.34}$$

Applying the map Φ of Proposition 3.2.8 to both sides of the Dyson-Schwinger equation introduced in Eq. (3.3) yields

$$X_W = \Phi(X) = \mathbb{I}_W + \sum_{n\geq 1} \alpha^n B_+^{l_n} \left(X_W^{\sqcup\!\sqcup(n+1)} \right), \tag{3.35}$$

where Eqs. (3.29, 3.30, 3.34) were used. Then, X_W is an element of the Hopf algebra of words and can be given by application of Φ to Eqs. (3.4, 3.5). Thus,

$$X_W = \sum_{n\geq 0} \alpha^n w_n, \tag{3.36}$$

where the coefficients $w_n = \Phi(c_n)$ are explicitly given by

$$w_0 = \mathbb{I}_W, \quad w_n = \sum_{m=1}^{n} B_+^{l_m} \left(\sum_{k_1+\ldots k_{m+1}=n-m, k_i\geq 0} w_{k_1} \sqcup\!\sqcup \ldots \sqcup\!\sqcup w_{k_{m+1}} \right), \tag{3.37}$$

which follows directly from Eq. (3.5) using Eqs. (3.29, 3.30, 3.34). The first word coefficients are

$$\begin{aligned} w_1 &= B_+^{l_1}(w_0 \sqcup\!\sqcup w_0) = l_1, && (3.38)\\ w_2 &= B_+^{l_1}(2w_1 \sqcup\!\sqcup w_0) + B_+^{l_2}(w_0 \sqcup\!\sqcup w_0 \sqcup\!\sqcup w_0) \\ &= 2l_1 l_1 + l_2, && (3.39)\\ w_3 &= B_+^{l_1}(w_1 \sqcup\!\sqcup w_1 + 2w_2 \sqcup\!\sqcup w_0) + B_+^{l_2}(3w_1 \sqcup\!\sqcup w_0 \sqcup\!\sqcup w_0) + B_+^{l_3}(w_0 \sqcup\!\sqcup w_0 \sqcup\!\sqcup w_0 \sqcup\!\sqcup w_0) \\ &= l_1(l_1 \sqcup\!\sqcup l_1 + 2(2l_1 l_1 + l_2)) + 3l_2 l_1 + l_3 \\ &= 6l_1 l_1 l_1 + l_1 p(l_1, l_1) + 2l_1 l_2 + 3l_2 l_1 + l_3. && (3.40) \end{aligned}$$

The crucial point is that the Hopf algebra of words is filtratable (which is not necessary for $\mathcal{H}$). The next section describes in detail, how to filtrate the words w_n, s.t. application of renormalized Feynman rules $\phi_R \circ \Phi^{-1}$ to each filtrated term maps to a certain power of L in the log-expansion (Eq. (2.5)) of $\phi_R \circ \Phi^{-1}(X_W) = \phi_R(X)$.

3.3. Filtration of words

The filtration of words requires some additional mathematical background. Therefore Section 3.3.1 explains the concept of a Lie algebra and its lower central series. Section 3.3.2 introduces the universal enveloping algebra of a Lie algebra and Section 3.3.3 shows that the Hopf algebra of words is the dual of the universal enveloping algebra of some Lie algebra. Finally, Section 3.3.4 summarizes all these mathematical structures and deduces the filtration method for words, which will be illustrated in Section 3.3.5, using the example of the coefficient w_3 (Eq. (3.40)) of the solution of the Dyson-Schwinger equation in Eq. (3.35) .

3.3.1. Lie algebra and lower central series

Definition 3.3.1. *A vector space $\mathcal{L}$ over a field $\mathbb{K}$ together with a bi-linear map $[\cdot,\cdot] : \mathcal{L}\otimes\mathcal{L}\to\mathcal{L}$, which fulfills*

$$0 = [x_1,[x_2,x_3]] + [x_2,[x_3,x_1]] + [x_3,[x_1,x_2]], \tag{3.41}$$
$$0 = [x_1,x_1] \tag{3.42}$$

$\forall x_1,x_2,x_3\in\mathcal{L}$, is called Lie algebra $(\mathcal{L},[\cdot,\cdot])$. The map $[\cdot,\cdot]$ is called the Lie bracket, which is antisymmetric, because of bi-linearity and Eq. (3.42). Eq. (3.41) is called Jacobi identity.

Although, $(\mathcal{L},[\cdot,\cdot])$ is called a Lie algebra, the vector space $\mathcal{L}$ does not need to have an associative product as well as a neutral element of this product. Thus, $\mathcal{L}$ does not need to form an algebra. In particular, if it does not form an algebra, the Lie bracket of two Lie algebra elements $[x_1,x_2]$ cannot be the product of x_1 and x_2 minus the product of x_2 and x_1, because there is no product defined. However, a Lie algebra $(\mathcal{L},[\cdot,\cdot])$ (often abbreviated by $\mathcal{L}$) contains another very convenient structure, i.e. a lower central series as defined in

Definition 3.3.2. *Let $(\mathcal{L},[\cdot,\cdot])$ be a Lie algebra. The descending series of sub-algebras*

$$\mathcal{L} = \mathcal{L}_1 \trianglerighteq \mathcal{L}_2 \trianglerighteq \mathcal{L}_3 \trianglerighteq \dots \tag{3.43}$$

is called a lower central series of $\mathcal{L}$, iff $\mathcal{L}_{n+1}$ is generated by all Lie brackets $[x_1,x_2]$ with $x_1\in\mathcal{L}$ and $x_2\in\mathcal{L}_n$, $n\in\mathbb{N}$.

For example, $\mathcal{L}_2 = [\mathcal{L},\mathcal{L}]$ and $\mathcal{L}_3 = [\mathcal{L},[\mathcal{L},\mathcal{L}]]$. Now, denote those elements of $\mathcal{L}$, which cannot be written as a Lie bracket of any two elements of $\mathcal{L}$ by $x_1,x_2,x_3,\dots$. It is clear that $\{x_1,x_2,x_3,\dots\}$ does not form a basis of $\mathcal{L}$, as long as not all Lie brackets $[x_i,x_j]$, $x_i,x_j\in\{x_1,x_2,x_3,\dots\}$ vanish. Furthermore, $x_1,x_2,x_3,\dots$ together with all elements derived from (multiple) application of the Lie bracket on $x_1,x_2,x_3,\dots$ cannot form a basis of $\mathcal{L}$ because of the properties of $[\cdot,\cdot]$ defined in Definition 3.3.1. Indeed, there are for example six elements $[x_i,[x_j,x_k]]$ and six elements $[[x_i,x_j],x_k]$ for three different positive integers i,j,k, but only two out of these twelve are linearly independent, because of the antisymmetry of the Lie bracket as well as the Jacobi identity, (Eq. (3.41)). However, it is possible to define a basis of $\mathcal{L}$, e.g. the Hall basis [14]. This requires a lexicographical ordering between the elements of $\mathcal{L}$, which is usually given by $x_1 < x_2 < x_3 < \dots$, $x_i < [x_j,x_k], \forall i,j,k\in\mathbb{N}$ etc. It is not important to know the ordering in detail as long as it exists and it is strict. Then, the Hall basis elements of $\mathcal{L}$ are recursively defined by the following

Proposition 3.3.3. *If x and x' are basis elements of $\mathcal{L}$, then $[x,x']$ is a basis element, iff it holds the following two conditions*

$$x < x', \tag{3.44}$$
$$\text{if } x' = [x'',x'''], \text{ then, } x \geq x''. \tag{3.45}$$

A proof is given in [14]. For example with this definition, if $x_i < x_j < x_k$, only the two elements $[x_j, [x_i, x_k]]$ and $[x_k, [x_i, x_j]]$ out of the considered twelve elements above are Hall basis elements of $\mathcal{L}$.

3.3.2. Universal enveloping algebra

Definition 3.3.4. *Let $(\mathcal{L}, [\cdot,\cdot])$ be a Lie algebra. Then, an associative and unital algebra $(\mathcal{A}_\mathcal{L}, m_\mathcal{A}, \mathbb{I}_\mathcal{A})$ is called an enveloping algebra of $(\mathcal{L}, [\cdot,\cdot])$, iff there exists a Lie algebra homomorphism $\rho_\mathcal{A} : \mathcal{L} \to \mathcal{A}_\mathcal{L}$, which fulfills*

$$m_\mathcal{A} \circ (\rho_\mathcal{A} \otimes \rho_\mathcal{A}) (x \otimes x' - x' \otimes x) = \rho_\mathcal{A} ([x, x']) \tag{3.46}$$

$\forall x, x' \in \mathcal{L}$.

There may be plenty of enveloping algebras for a given Lie algebra. However, there is only one unique universal enveloping algebra, which is introduced in

Definition 3.3.5. *Let $(\mathcal{U}(\mathcal{L}), m_\mathcal{U}, \mathbb{I}_\mathcal{U})$ be an enveloping algebra and denote the Lie algebra homomorphism from $\mathcal{L}$ to $\mathcal{U}(\mathcal{L})$ by $\rho_\mathcal{U}$. $(\mathcal{U}(\mathcal{L}), m_\mathcal{U}, \mathbb{I}_\mathcal{U})$ is called universal enveloping algebra of $(\mathcal{L}, [\cdot,\cdot])$, iff for each enveloping algebra $(\mathcal{A}_\mathcal{L}, m, \mathbb{I})$ and Lie algebra homomorphism $\rho_\mathcal{A}$, $\exists$ a unique algebra homomorphism $\rho_{\mathcal{U}\to\mathcal{A}} : \mathcal{U}(\mathcal{L}) \to \mathcal{A}_\mathcal{L}$, which fulfills*

$$\rho_\mathcal{A} = \rho_{\mathcal{U}\to\mathcal{A}} \circ \rho_\mathcal{U}. \tag{3.47}$$

This property of $(\mathcal{U}(\mathcal{L}), m_\mathcal{U}, \mathbb{I}_\mathcal{U})$ is called universal property.

The important point of the universal enveloping algebra is its universality. The knowledge of $(\mathcal{U}(\mathcal{L}), m_\mathcal{U}, \mathbb{I}_\mathcal{U})$ is sufficient to construct any other enveloping algebra $(\mathcal{A}_\mathcal{L}, m_\mathcal{A}, \mathbb{I}_\mathcal{A})$ of the Lie algebra $(\mathcal{L}, [\cdot,\cdot])$ through an algebra homomorphism $\rho_{\mathcal{U}\to\mathcal{A}}$. A powerful hypothesis is

Proposition 3.3.6. *For any Lie algebra $(\mathcal{L}, [\cdot,\cdot])$, the universal enveloping algebra $(\mathcal{U}(\mathcal{L}), m_\mathcal{U}, \mathbb{I}_\mathcal{U})$ exists and is unique up to isomorphism.*

Proof. The existence can be shown by a direct construction. Therefore, define the tensor algebra $(T(\mathcal{L}), \otimes, 1)$ to be the algebra of tensors of any rank on the vector space $\mathcal{L}$ with associative tensor product $\otimes$ and unit $1 \in \mathbb{K}$. It always exists and its underlying vector space is

$$T(\mathcal{L}) = \bigoplus_{n\geq 0} \mathcal{L}^{\otimes n} = \mathbb{K} \oplus \mathcal{L} \oplus (\mathcal{L} \otimes \mathcal{L}) \oplus (\mathcal{L} \otimes \mathcal{L} \otimes \mathcal{L}) \oplus \dots . \tag{3.48}$$

The unit 1 is meant, s.t. $1 \otimes x \cong x \ \forall x \in T(\mathcal{L})$. Now, consider the subspace

$$I = \{x_1' \otimes (x_1 \otimes x_2 - x_2 \otimes x_1 - [x_1, x_2]) \otimes x_2' \,|\, x_1, x_2 \in \mathcal{L}\,;\, x_1', x_2' \in T(\mathcal{L})\} \subset T(\mathcal{L}). \tag{3.49}$$

I is a two-sided ideal, which means that $\forall x \in I, x' \in T(\mathcal{L}) : x \otimes x', x' \otimes x \in I$. Given this ideal I, an equivalence relation can be defined by

$$x \sim x', \text{ iff } (x - x') \in I \tag{3.50}$$

for $x, x' \in T(\mathcal{L})$. This equivalence relation divides the elements of $T(\mathcal{L})$ into equivalent classes, i.e. sets

$$[x] = \{x' \in T(\mathcal{L}) |\, x' \sim x\}\,. \tag{3.51}$$

For example, for $x_1, x_2, x_3 \in \mathcal{L}$,

$$\begin{aligned}
[0] &= \{0, x_1 \otimes x_2 - x_2 \otimes x_1 - [x_1, x_2]\,, \ldots\}\,, && (3.52)\\
[[x_1, x_2]] &= \{[x_1, x_2]\,, x_1 \otimes x_2 - x_2 \otimes x_1, \ldots\}\,, && (3.53)\\
[x_1 \otimes x_2 \otimes x_3] &= \{x_1 \otimes x_2 \otimes x_3, [x_1, x_2] \otimes x_3 + x_2 \otimes x_1 \otimes x_3, \ldots\}\,, \text{ etc.} && (3.54)
\end{aligned}$$

These equivalent classes also form a vector space, denoted by $\mathcal{U}(\mathcal{L}) = T(\mathcal{L})/I$. It is called the quotient space of $T(\mathcal{L})$ and the ideal I, which generates the equivalence relation in Eq. (3.50). Therefore, addition of two equivalent classes is totally well defined by addition of the classes representatives, i.e.

$$[x] + [x'] = [x + x'] \tag{3.55}$$

$\forall x, x' \in T(\mathcal{L})$. Furthermore, $\mathcal{U}(\mathcal{L})$ forms an algebra as well. The product $m_{\mathcal{U}}$ of two equivalent classes is well defined by the tensor product of the classes representatives, which means

$$m_{\mathcal{U}}([x] \otimes [x']) = [x \otimes x'] \tag{3.56}$$

$\forall x, x' \in T(\mathcal{L})$. It is associative, because the tensor product itself is associative. The neutral element $\mathbb{I}_{\mathcal{U}}$ is the equivalent class [1]. Moreover, there exists a canonical Lie algebra homomorphism $\rho_{\mathcal{U}} : \mathcal{L} \to \mathcal{U}(\mathcal{L})$, which fulfills Eq. (3.46). It can be defined by $\rho_{\mathcal{U}}(x) = [x]$ for any $x \in \mathcal{L}$ and indeed, $\forall x, x' \in \mathcal{L}$,

$$\begin{aligned}
m_{\mathcal{U}} \circ (\rho_{\mathcal{U}} \otimes \rho_{\mathcal{U}})(x \otimes x' - x \otimes x') &= m_{\mathcal{U}}([x] \otimes [x']) - m_{\mathcal{U}}([x] \otimes [x'])\\
&= [x \otimes x'] - [x' \otimes x]\\
&= [x \otimes x' - x' \otimes x]\\
&= [[x, x']]\\
&= \rho_{\mathcal{U}}([x, x']). && (3.57)
\end{aligned}$$

Thus, $(\mathcal{U}(\mathcal{L}), m_{\mathcal{U}}, \mathbb{I}_{\mathcal{U}})$ is an enveloping algebra of $(\mathcal{L}, [\cdot, \cdot])$. It is even the universal enveloping algebra, because it is constructed directly from the Tensor algebra $T(\mathcal{L})$, which fulfills the universal property as well [15]. The uniqueness of the universal enveloping algebra can be shown as follows. Assume that there are two universal enveloping algebras $\mathcal{U}_1(\mathcal{L})$ and $\mathcal{U}_2(\mathcal{L})$. Then, by the universal property, there exist two unique algebra homomorphisms $\rho_{\mathcal{U}_1(\mathcal{L}) \to \mathcal{U}_2(\mathcal{L})} : \mathcal{U}_1(\mathcal{L}) \to \mathcal{U}_2(\mathcal{L})$ and $\rho_{\mathcal{U}_2(\mathcal{L}) \to \mathcal{U}_1(\mathcal{L})} : \mathcal{U}_2(\mathcal{L}) \to \mathcal{U}_1(\mathcal{L})$. But then, $\rho_{\mathcal{U}_1(\mathcal{L}) \to \mathcal{U}_2(\mathcal{L})} = \rho^{-1}_{\mathcal{U}_2(\mathcal{L}) \to \mathcal{U}_1(\mathcal{L})}$ is an isomorphism of algebras. Thus, the universal enveloping algebra is unique up to isomorphism. □

At this point, it is important to note that the product $\otimes$ of the tensor algebra $T(\mathcal{L})$ of a Lie algebra is nothing else but a concatenation product. Indeed, represent the elements of the Lie algebra $\mathcal{L}$ by an ordered alphabet $x_1, x_2, x_3, \ldots, [x_1, x_2], \ldots$. Thus, tensor products can be expressed by 'words', e.g. $x_1 \otimes x_2$ by $x_1 x_2$ and so on. The neutral element 1 is characterized by the 'empty word'. This goes through the composition of equivalent classes, which means for the universal enveloping algebra $\mathcal{U}(\mathcal{L})$ that $[x_1 \otimes x_2]$ can be represented by $[x_1 x_2]$ and so on. Writing the product of the tensor algebra as the concatenation product of words turns out to be very convenient, as becomes clear in the following

Proposition 3.3.7. *There exists a co-product $\Delta_{\mathcal{U}} : \mathcal{U}(\mathcal{L}) \to \mathcal{U}(\mathcal{L}) \otimes \mathcal{U}(\mathcal{L})$, a co-unit $\hat{\mathbb{I}}_{\mathcal{U}} : \mathcal{U}(\mathcal{L}) \to \mathbb{K}$ and an antipode $S_{\mathcal{U}} : \mathcal{U}(\mathcal{L}) \to \mathcal{U}(\mathcal{L})$, s.t. $\left(\mathcal{U}(\mathcal{L}), m_{\mathcal{U}}, \mathbb{I}_{\mathcal{U}}, \Delta_{\mathcal{U}}, \hat{\mathbb{I}}_{\mathcal{U}}, S_{\mathcal{U}}\right)$ forms a Hopf algebra.*

The proof is given in [15]. Indeed, if the tensor product of the tensor algebra is not expressed by the concatenation product, it would cause confusion with the tensor product in the image of the co-product. However, these two tensor products are not the same.

3.3.3. Hopf algebra duality

The final mathematical structure needed in the next section is the concept of dual Hopf algebras, which will be sketched in this section. Consider two Hopf algebras $(\mathcal{H}_1, m_1, \mathbb{I}_1, \Delta_1, \hat{\mathbb{I}}_1, S_1)$ and $(\mathcal{H}_2, m_2, \mathbb{I}_2, \Delta_2, \hat{\mathbb{I}}_2, S_2)$ over the same field $\mathbb{K}$. Moreover, let $\mathcal{H}_2$ be the vector space of functionals $G : \mathcal{H}_1 \to \mathbb{K}$ acting on elements $y \in \mathcal{H}_1$. Therefore, define a bi-linear form $\langle \cdot, \cdot \rangle : \mathcal{H}_1 \otimes \mathcal{H}_2 \to \mathbb{K}$, s.t.

$$\langle y, G \rangle = G(y) \in \mathbb{K} \tag{3.58}$$

$\forall y \in \mathcal{H}_1, G \in \mathcal{H}_2$. This can be extended, s.t. $\forall y, y', \ldots \in \mathcal{H}_1$ and $\forall G, G', \ldots \in \mathcal{H}_2$,

$$\langle y \otimes y' \otimes \ldots, G \otimes G' \otimes \ldots \rangle = G(y) \cdot G'(y') \cdot \ldots \in \mathbb{K}, \tag{3.59}$$

where the product $\cdot$ on the R.H.S. is usual multiplication. Bi-linearity means that $\forall y, y' \in \mathcal{H}_1$ and $\forall G, G' \in \mathcal{H}_2$ as well as for $\lambda \in \mathbb{K}$,

$$\langle y + y', G \rangle = G(y) + G(y'), \tag{3.60}$$

$$\langle y, G + G' \rangle = G(y) + G'(y), \tag{3.61}$$

$$\langle \lambda y, G \rangle = \langle y, \lambda G \rangle = \lambda G(y). \tag{3.62}$$

The definition of Hopf algebra duality requires a pairing between the elements of $\mathcal{H}_1$ and $\mathcal{H}_2$. This is a one-to-one assignment $\mathcal{H}_1 \to \mathcal{H}_2$. In other words, for each element $y \in \mathcal{H}_1$, there is a dual element $G \in \mathcal{H}_2$. In order to relate $\mathcal{H}_1$ to $\mathcal{H}_2$ in Hopf duality, the dual element of $\mathbb{I}_1$ must be $\mathbb{I}_2$. As turns out, this is the co-unit $\hat{\mathbb{I}}_1$. Furthermore, it is also possible to assign a 'dual' to the co-unit map $\hat{\mathbb{I}}_1$, which must be $\hat{\mathbb{I}}_2$ in order to

make $\mathcal{H}_1$ and $\mathcal{H}_2$ be dual Hopf algebras. As turns out, this is the map, which evaluates a given map at the unit $\mathbb{I}_1$, denoted by $eval_{\mathbb{I}_1}$. Thus,

$$\langle \mathbb{I}_1, \mathbb{I}_2 \rangle = \langle \mathbb{I}_1, \hat{\mathbb{I}}_1 \rangle = \hat{\mathbb{I}}_1(\mathbb{I}_1) = 1, \quad \langle \hat{\mathbb{I}}_1, \hat{\mathbb{I}}_2 \rangle = \langle \hat{\mathbb{I}}_1, eval_{\mathbb{I}_1} \rangle = \hat{\mathbb{I}}_1(\mathbb{I}_1) = 1. \tag{3.63}$$

The final requirement of Hopf algebra duality is that the 'dual' of the product m_1 is the co-product Δ_2 and the other way around, which means that

$$\langle m_1 (y \otimes y'), G \rangle = \langle y \otimes y', \Delta_2(G) \rangle \tag{3.64}$$
$$\langle \Delta_1(y), G \otimes G' \rangle = \langle y, m_2(G \otimes G') \rangle \tag{3.65}$$

$\forall y, y' \in \mathcal{H}_1, \forall G, G' \in \mathcal{H}_2$. This thesis does not investigate dual Hopf algebras in more details, see the work of Milnor and Moore for more information [15].

3.3.4. Filtration method

The usefulness of the mathematical structures introduced in the previous three sections is that there exists some Lie algebra, whose universal enveloping algebra is the dual of the Hopf algebra of words. This powerful statement is proven within the scope of the Milnor-Moore theorem [15]. Establish a Lie algebra $\mathcal{L}$, s.t. for each word $u \in \mathcal{H}_W$, there is one dual element $[x]$ in the universal enveloping algebra $\mathcal{U}(\mathcal{L})$. For example, denote the dual of the letter $l_i \in \mathcal{H}_W$, $i \in \mathbb{N}$ by $[x_i] \in \mathcal{U}(\mathcal{L})$, where $x_i \in \mathcal{L}$. In full analogy, denote the dual of the letter $p(l_i, l_j) \in \mathcal{H}_W$, $l_i, l_j \in \mathbb{N}$ by $[p(x_i, x_j)] \in \mathcal{U}(\mathcal{L})$, where $p(x_i, x_j) \in \mathcal{L}$ denotes a Lie algebra element. The dual of $\mathbb{I}_W$ is $\mathbb{I}_{\mathcal{U}} = \hat{\mathbb{I}}_W$ and the 'dual' of $\hat{\mathbb{I}}_W$ is $\hat{\mathbb{I}}_{\mathcal{U}} = eval_{\mathbb{I}_W}$. Words consisting of more than one letter can also be put in duality. The dual of the word $l_i l_j$, where $l_i, l_j \in \mathcal{H}_L$ is denoted by $[x_i x_j]$, where $x_i x_j$ is the concatenation of two Lie algebra elements and thus, $x_i x_j \in \mathcal{L} \otimes \mathcal{L}$. The dual elements of further words can be obtained straightforward by concatenation of Lie algebra elements. Then, the bi-linear form can be defined as

$$\langle u, [x] \rangle = [x](u) = \begin{cases} 1, & [x] \text{ is dual to } u \\ 0, & \text{else} \end{cases} \tag{3.66}$$

for any word $u \in \mathcal{H}_W$ and $[x] \in \mathcal{U}(\mathcal{L})$. Indeed, $\langle \cdot, \cdot \rangle$ fulfills Eqs. (3.63-3.65). This remains without proof, see [13] for more details. Thus, $\mathcal{U}(\mathcal{L})$ is the dual Hopf algebra of $\mathcal{H}_W$. The crucial point is that the renormalized Feynman rules of Section 3.2, $\phi_R \circ \Phi^{-1}$ respect the dual Hopf algebra structure. This is explained in the following

Proposition 3.3.8. *Let u be an element of the Hopf algebra of words $\mathcal{H}_W$ and $[x] \in \mathcal{U}(\mathcal{L})$ be its dual. If the element $x \in T(\mathcal{L})$ is also an element of the Lie algebra $\mathcal{L} \subset T(\mathcal{L})$ itself, then, renormalized Feynman rules $\phi_R \circ \Phi^{-1}$ map u to the L^1-term in the log-expansion (Eq. (2.5)).*

This is the most important proposition in this thesis, its proof is given in [16]. Consider for example the element $l_i l_j - l_j l_i \in \mathcal{H}_W$, where $l_i \neq l_j \in \mathcal{H}_L$. Because of the bi-linearity of $\langle \cdot, \cdot \rangle$, its dual is $[x_i x_j] - [x_j x_i] = [x_i x_j - x_j x_i] = [[x_i, x_j]] \in \mathcal{U}(\mathcal{L})$ and in fact,

$[x_i, x_j] \in \mathcal{L} \subset T(\mathcal{L})$. Thus, $\phi_R \circ \Phi^{-1}(l_i l_j - l_j l_i) \propto L$, although the separate application of renormalized Feynman rules to both words $l_i l_j$ and $l_j l_i$ may yield different powers of L. On the other hand, renormalized Feynman rules are character-like in any Hopf algebra. Thus, Eq. (3.6) translates into the Hopf algebra of words to

$$\phi_R \circ \Phi^{-1}(m_W(u \otimes u')) = \phi_R \circ \Phi^{-1}(u \sqcup\!\sqcup u') = \phi_R \circ \Phi^{-1}(u) \cdot \phi_R \circ \Phi^{-1}(u') \tag{3.67}$$

$\forall u, u' \in \mathcal{H}_W$. This yields to

Proposition 3.3.9. *Let $u_1, u_2, \ldots, u_n \in \mathcal{H}_W$ be elements of the Hopf algebra of words, s.t. their duals are $[x_1], [x_2], \ldots, [x_n] \in \mathcal{U}(\mathcal{L})$. If furthermore $x_1, x_2, \ldots, x_n \in \mathcal{L} \subset T(\mathcal{L})$, then, application of renormalized Feynman rules $\phi_R \circ \Phi^{-1}$ to the shuffle product $u_1 \sqcup\!\sqcup u_2 \sqcup\!\sqcup \ldots \sqcup\!\sqcup u_n$ maps to the L^n-term in the log-expansion (Eq. (2.5)).*

The proof is obvious using Proposition 3.3.8 and Eq. (3.67). Consider for example the word $l_i l_j$, where $l_i \neq l_j \in \mathcal{H}_L$. It can be written as

$$l_i l_j = \underbrace{\frac{1}{2} l_i \sqcup\!\sqcup l_j}_{\propto L^2} + \underbrace{\frac{1}{2}(l_i l_j - l_j l_i)}_{\propto L} - \underbrace{\frac{1}{2} p(l_i, l_j)}_{\propto L} \tag{3.68}$$

and is therefore split into terms mapping to the particular powers of L in the leading log-expansion (Eq. (2.5)). At this point, it is convenient to consider elements $u \in \mathcal{H}_W$, whose duals $[x] \in \mathcal{U}(\mathcal{L})$ fulfill that $x \in \mathcal{L} \subset T(\mathcal{L})$, as letter elements in $\mathcal{H}_L$. For example, denote the element $l_i l_j - l_j l_i$ for $i < j \in \mathbb{N}$, $l_i, l_j \in \mathcal{H}_L$ as the concatenation Lie bracket $[l_i, l_j] \in \mathcal{H}_L$. Then, Eq. (3.68) reads

$$l_i l_j = \underbrace{\frac{1}{2} l_i \sqcup\!\sqcup l_j}_{\propto L^2} + \underbrace{\frac{1}{2}[l_i, l_j]}_{\propto L} - \underbrace{\frac{1}{2} p(l_i, l_j)}_{\propto L}. \tag{3.69}$$

The introduction of concatenation Lie brackets at the level of words may cause problems choosing a basis. Therefore, the discussion of the lower central series of a Lie algebra $\mathcal{L}$ and its Hall basis (Section 3.3.1) can be translated to the Lie algebra $(\mathcal{H}_L, [\cdot,\cdot])$. This only requires a lexicographical ordering of the letters, which can be given by $l_1 < l_2 < \ldots < p(l_1, l_1) < p(l_1, l_2) < \ldots < [l_1, l_2] < \ldots$. It is not important to know about the ordering in details as long as it exists and is strict. Propositions 3.3.8 and 3.3.9, as well as the introduction of the Lie algebra $(\mathcal{H}_L, [\cdot,\cdot])$ provide a canonical way to filtrate the words w_n (Eq. (3.37)), which are the solution coefficients of the Dyson-Schwinger equation in Eq. (3.35). This will be illustrated using the example of the coefficient w_3 (Eq. (3.40)) in the next section.

3.3.5. An example: w_3

Recall Eq. (3.40), i.e.

$$w_3 = 6 l_1 l_1 l_1 + l_1 p(l_1, l_1) + 2 l_1 l_2 + 3 l_2 l_1 + l_3. \tag{3.70}$$

First, take the longest word on the R.H.S., i.e. $l_1 l_1 l_1$. It is already given in the lexicographically proper order. Now, consider the shuffle product

$$\begin{aligned} l_1 \sqcup\!\sqcup l_1 \sqcup\!\sqcup l_1 &= l_1 \sqcup\!\sqcup (2l_1 l_1 + p(l_1, l_1)) \\ &= 2l_1 l_1 l_1 + 2l_1 (2l_1 l_1 + p(l_1, l_1)) + 3p(l_1, l_1) l_1 + l_1 p(l_1, l_1) + p(l_1, l_1, l_1) \\ &= 6l_1 l_1 l_1 + 3l_1 p(l_1, l_1) + 3p(l_1, l_1) l_1 + p(l_1, l_1, l_1). \end{aligned} \tag{3.71}$$

Inserting it into Eq. (3.70) yields

$$w_3 = l_1 \sqcup\!\sqcup l_1 \sqcup\!\sqcup l_1 - 2l_1 p(l_1, l_1) - 3p(l_1, l_1) l_1 - p(l_1, l_1, l_1) + 2l_1 l_2 + 3l_2 l_1 + l_3. \tag{3.72}$$

Second, take all words with length two on the R.H.S. and bring them into the lexicographically proper order, using the concatenation Lie bracket. Thus,

$$w_3 = l_1 \sqcup\!\sqcup l_1 \sqcup\!\sqcup l_1 - 5l_1 p(l_1, l_1) + 3[l_1, p(l_1, l_1)] - p(l_1, l_1, l_1) + 5l_1 l_2 - 3[l_1, l_2] + l_3. \tag{3.73}$$

On the other hand, consider the shuffle products

$$\begin{aligned} l_1 \sqcup\!\sqcup p(l_1, l_1) &= l_1 p(l_1, l_1) + p(l_1, l_1) l_1 + p(l_1, l_1, l_1) \\ &= 2l_1 p(l_1, l_1) - [l_1, p(l_1, l_1)] + p(l_1, l_1, l_1), \end{aligned} \tag{3.74}$$

$$\begin{aligned} l_1 \sqcup\!\sqcup l_2 &= l_1 l_2 + l_2 l_1 + p(l_1, l_2) \\ &= 2l_1 l_2 - [l_1, l_2] + p(l_1, l_2), \end{aligned} \tag{3.75}$$

which are brought to the lexicographically proper order, using the concatenation Lie bracket. Inserting Eqs. (3.74, 3.75) into Eq. (3.73) finally yields

$$\begin{aligned} w_3 &= l_1 \sqcup\!\sqcup l_1 \sqcup\!\sqcup l_1 - \frac{5}{2}(l_1 \sqcup\!\sqcup p(l_1, l_1) + [l_1, p(l_1, l_1)] - p(l_1, l_1, l_1)) + 3[l_1, p(l_1, l_1)] - \\ &\quad - p(l_1, l_1, l_1) + \frac{5}{2}(l_1 \sqcup\!\sqcup l_2 + [l_1, l_2] - p(l_1, l_2)) - 3[l_1, l_2] + l_3 \\ &= \underbrace{l_1 \sqcup\!\sqcup l_1 \sqcup\!\sqcup l_1}_{\propto L^3} - \underbrace{\frac{5}{2} l_1 \sqcup\!\sqcup p(l_1, l_1)}_{\propto L^2} + \underbrace{\frac{5}{2} l_1 \sqcup\!\sqcup l_2}_{\propto L^2} + \underbrace{\frac{1}{2}[l_1, p(l_1, l_1)]}_{\propto L} - \underbrace{\frac{1}{2}[l_1, l_2]}_{\propto L} - \\ &\quad - \underbrace{\frac{5}{2} p(l_1, l_2)}_{\propto L} + \underbrace{\frac{3}{2} p(l_1, l_1, l_1)}_{\propto L} + \underbrace{l_3}_{\propto L}. \end{aligned} \tag{3.76}$$

The illustrated filtration method is completely canonical and finally, w_3 is given in terms, which map to a certain power of L in the log-expansion (Eq. (2.5)) by application of renormalized Feynman rules $\phi_R \circ \Phi^{-1}$. This method can be applied to each w_n (Eq. (3.37)), as well as for the coefficients of any other Dyson-Schwinger equation of a physical quantum field theory. One only has to pay attention when introducing Lie brackets. Each new created Lie bracket needs to be a Hall basis element (see Proposition 3.3.3).

4. Results

Upon giving the physical and mathematical background in the previous two chapters, the results are presented now. Therefore, a program was written in `FORM` [17], which computes the coefficients w_n (Eq. (3.37)) of the Dyson-Schwinger equation in Eq. (3.35). Finally, the program performs the filtration explained in Section 3.3.4. Section 4.1 explains the usage of the program and gives some exemplary output, i.e. w_n in filtrated terms for $n = 1 \ldots 4$. The program code, as well as explanations and program tests are presented in Appendix A. Finally, Section 4.2 gives an analytical formula for the multiplicities of full shuffle terms in the coefficients w_n.

4.1. Some program calls

The output is given in text form. It is displayed in a terminal after running the main program by `form Main.frm`. For example, for $n = 1$, it gives back $w_1 = l_1$ (see Listing 4.1), which matches Eq. (3.38). The first non-trivial case is $n = 2$. The program gives the answer $w_2 = l_1 \sqcup\!\sqcup\, l_1 + l_2 - p\,(l_1, l_1)$ (see Listing 4.2). In the output, the shuffle product is abbreviated by a function `SH`, which can also have more than two arguments, e.g., $\texttt{SH}\,(l_1, l_1, l_2)$ represents $l_1 \sqcup\!\sqcup\, l_1 \sqcup\!\sqcup\, l_2$. It is trivial to check that the answer of the program is correct. Indeed,

$$w_2 = l_1 \sqcup\!\sqcup\, l_1 + l_2 - p\,(l_1, l_1) = 2l_1 l_1 + l_2, \tag{4.1}$$

which exactly matches Eq. (3.39). Furthermore, w_2 is completely filtrated. $l_1 \sqcup\!\sqcup\, l_1$ maps to the L^2-term in the log-expansion (Eq. (2.5)) by application of renormalized Feynman rules $\phi_R \circ \Phi^{-1}$ and the other two letters map to terms linear in L. This was explained in detail in Section 3.3.4. Two more examples are given for $n = 3, 4$ in Listings 4.3 and 4.4. A function c abbreviates the concatenation Lie bracket, e.g., $c\,(l_1, l_2) = [l_1, l_2] = l_1 l_2 - l_2 l_1$ as explained in Section 3.3.4. The case w_3 was already discussed in Section 3.3.5 and as can be easily seen, the output of the program in Listing 4.3 matches the filtration by hand (Eq. (3.76)). The crosscheck for the word w_4 in filtrated terms is not collected here, but each term of the output maps to a certain power of L in the log-expansion. The higher n, the more complicated is the analysis that the program returns the correct answers. Therefore, the code was piecewise tested in Appendix A. Finally, Tab. 4.1 denotes some running times of the program on a local office machine. Even for $n = 9$ it does not take longer than half a minute. However, for $n > 9$, `FORM` could not handle the big output terms using a normal `MaxTermSize`. Using a higher `MaxTermSize`, this problem could be solved for some more values of n but even the output becomes too long in order to see any information about the filtration of the coefficients w_n. However,

n	7	8	9
Running time in seconds	1	5	24

Table 4.1.: Running time in seconds for different values of n

by several looks at the first coefficients w_n, $n = 1, 2, 3, 4, \ldots$, an analytical formula was discovered for the multiplicities of full shuffle terms in the filtration of w_n. This will be explained in the next section.

Listing 4.1: Output for w_1

```
FORM 4.0 (Jul  2 2013) 64-bits
    #define n "1"
    #include Declarations.h
    #-
    Off statistics;
    #call Word
    #call Sorting
    Print;

   [w_n] =
      l1;

    .end
  0.00 sec out of 0.00 sec
```

Listing 4.2: Output for w_2

```
FORM 4.0 (Jul  2 2013) 64-bits
    #define n "2"
    #include Declarations.h
    #-
    Off statistics;
    #call Word
    #call Sorting
    Print;

   [w_n] =
      SH(l1,l1) + l2 - p(l1,l1);

    .end
  0.00 sec out of 0.00 sec
```

Listing 4.3: Output for w_3

```
FORM 4.0 (Jul  2 2013) 64-bits
    #define n "3"
    #include Declarations.h
```

```
    #-
    Off statistics;
    #call Word
    #call Sorting
    Print;

   [w_n] =
       - 5/2*SH(l1,p(l1,l1)) + SH(l1,l1,l1) + 5/2*SH(l1,l2) + l3 +
       3/2*p(l1,l1,l1) - 5/2*p(l1,l2) + 1/2*c(l1,p(l1,l1))
       - 1/2*c(l1,l2);

    .end
  0.01 sec out of 0.01 sec
```

Listing 4.4: Output for w_4

```
FORM 4.0 (Jul  2 2013) 64-bits
    #define n "4"
    #include Declarations.h
    #-
    Off statistics;
    #call Word
    #call Sorting
    Print;

   [w_n] =
       4/3*SH(p(l1,l1),p(l1,l1)) + 14/3*SH(l1,p(l1,l1,l1))
       - 23/3*SH(l1,p(l1,l2)) + 3*SH(l1,c(l1,p(l1,l1))) -
       3*SH(l1,c(l1,l2)) - 13/3*SH(l1,l1,p(l1,l1)) + SH(l1,l1,l1,l1)
       + 13/3*SH(l1,l1,l2) + 3*SH(l1,l3) - 17/6*SH(l2,p(l1,l1)) +
       3/2*SH(l2,l2) + l4 - 3*p(l1,c(l1,p(l1,l1))) + 3*p(l1,c(l1,l2))
       - 8/3*p(l1,l1,l1,l1) + 37/6*p(l1,l1,l2) - 3*p(l1,l3)
       - 3/2*p(l2,l2) + c(l1,p(l1,l2)) - 1/3*c(l1,c(l1,p(l1,l1)))
       + 1/3*c(l1,c(l1,l2)) - c(l1,l3) + 3/2*c(l2,p(l1,l1));

    .end
  0.02 sec out of 0.02 sec
```

4.2. One discovered analytical formula

Denote the multiplicity of the full shuffle product $l_1^{\sqcup\!\sqcup n_1} \sqcup\!\sqcup\, l_2^{\sqcup\!\sqcup n_2} \sqcup\!\sqcup\, l_3^{\sqcup\!\sqcup n_3} \sqcup\!\sqcup \ldots$ in the word $w_{n_1+2n_2+3n_3+\ldots}$ by $f_{n_1,n_2,n_3\ldots}$, e.g. $w_8 = f_{1,0,1,1}\, l_1 \sqcup\!\sqcup l_3 \sqcup\!\sqcup l_4 + \ldots$, etc. As discussed in Section 3.3.4, a full shuffle product maps to the $L^{n_1+n_2+\cdots}$-term in the log-expansion (Eq. (2.5)) by application of renormalized Feynman rules. To the renormalized Feynman amplitude $\phi_R \circ \Phi^{-1}(X_W)$, it contributes a term proportional to $\alpha^{n_1+2n_2+3n_3\cdots} L^{n_1+n_2+n_3+\ldots}$

(see Eq.(3.36)). Such a term is called $\underbrace{\text{next-to-}\ldots\text{next-to-}}_{(n_2+2n_3+3n_4+\ldots)\text{times}}$leading-log term of the log-expansion (Eq. (2.5)). Some resulting values $f_{n_1,n_2,\ldots}$ given by the program are collected in Tab. 4.2. As discussed in Appendix B, for given values $n_2, n_3, \ldots$, the multiplicities $f_{0,n_2,n_3,\ldots}$, $f_{1,n_2,n_3,\ldots}$, $f_{2,n_2,n_3,\ldots}$, etc. form a series with generating functions $\mathcal{F}_{n_2,n_3,\ldots}$. For the particular values in Tab. 4.2, the associated generating functions $\mathcal{F}_{n_2,n_3,\ldots}$ were computed in Appendix B. Recall Eq. (B.10), i.e.

$$f_{n_1,n_2,n_3,\ldots} = \frac{1}{(n_1+n_2+\ldots)!}\frac{\mathrm{d}^{n_1+n_2+\ldots}}{\mathrm{d}z^{n_1+n_2+\ldots}}\mathcal{F}_{n_2,n_3,\ldots}(z)\Big|_{z=0}. \tag{4.2}$$

Furthermore, Eq. (B.13) states that $\mathcal{F}(z) = \frac{1}{1-z}$, which is the geometric series in z. Thus, using Eq. (4.2) yields

$$f_{n_1,n_{i>1}=0} = 1 \quad \forall n_1 \in \mathbb{N}_0, \tag{4.3}$$

which is reflected in Tab. 4.2. The associated full shuffle products $l_1^{\sqcup\!\sqcup n_1}$ to these multiplicities are the only terms in w_{n_1}, which map to leading-log terms in $\phi_R \circ \Phi^{-1}(X_W)$. Now, consider Eq. (B.20) for $i = 2, 3$, i.e.

$$\mathcal{F}_1(z) = \frac{1}{(1-z)^2}\ln\left(\frac{1}{1-z}\right) \tag{4.4}$$

$$\mathcal{F}_{0,1}(z) = \frac{1}{(1-z)^3} - \frac{1}{(1-z)^2}. \tag{4.5}$$

The full shuffle products $l_1^{\sqcup\!\sqcup n_1} \sqcup\!\sqcup\, l_2$ and $l_1^{\sqcup\!\sqcup n_1} \sqcup\!\sqcup\, l_3$ map to next-to-leading-log and next-to-next-to-leading-log terms in $\phi_R \circ \Phi^{-1}(X_W)$ respectively and their associated multiplicities can be calculated by Eq. (4.2) to be

$$f_{0,1} = \frac{\mathrm{d}}{\mathrm{d}z}\mathcal{F}_1(z)\Big|_{z=0} = \frac{1}{(1-z)^3}\left(2\ln\left(\frac{1}{1-z}\right)+1\right)\Big|_{z=0} = 1, \tag{4.6}$$

$$f_{1,1} = \frac{1}{2!}\frac{\mathrm{d}^2}{\mathrm{d}z^2}\mathcal{F}_1(z)\Big|_{z=0} = \frac{1}{2}\frac{1}{(1-z)^4}\left(6\ln\left(\frac{1}{1-z}\right)+5\right)\Big|_{z=0} = \frac{5}{2}, \tag{4.7}$$

$$f_{2,1} = \frac{1}{3!}\frac{\mathrm{d}^3}{\mathrm{d}z^3}\mathcal{F}_1(z)\Big|_{z=0} = \frac{1}{6}\frac{1}{(1-z)^5}\left(24\ln\left(\frac{1}{1-z}\right)+26\right)\Big|_{z=0} = \frac{13}{3}, \tag{4.8}$$

$$f_{3,1} = \frac{1}{4!}\frac{\mathrm{d}^4}{\mathrm{d}z^4}\mathcal{F}_1(z)\Big|_{z=0} = \frac{1}{24}\frac{1}{(1-z)^6}\left(120\ln\left(\frac{1}{1-z}\right)+154\right)\Big|_{z=0} = \frac{77}{12}, \tag{4.9}$$

$$f_{4,1} = \frac{1}{5!}\frac{\mathrm{d}^5}{\mathrm{d}z^5}\mathcal{F}_1(z)\Big|_{z=0} = \frac{1}{120}\frac{1}{(1-z)^7}\left(720\ln\left(\frac{1}{1-z}\right)+1044\right)\Big|_{z=0} = \frac{87}{10}, \tag{4.10}$$

$$f_{5,1} = \frac{1}{6!}\frac{\mathrm{d}^6}{\mathrm{d}z^6}\mathcal{F}_1(z)\Big|_{z=0} = \frac{1}{720}\frac{1}{(1-z)^8}\left(5040\ln\left(\frac{1}{1-z}\right)+8028\right)\Big|_{z=0} = \frac{223}{20}, \tag{4.11}$$

$$f_{0,0,1} = \frac{\mathrm{d}}{\mathrm{d}z}\mathcal{F}_{0,1}(z)\Big|_{z=0} = \left(\frac{3}{(1-z)^4} - \frac{2}{(1-z)^3}\right)\Big|_{z=0} = 1, \tag{4.12}$$

$$f_{1,0,1} = \frac{1}{2!}\frac{\mathrm{d}^2}{\mathrm{d}z^2}\mathcal{F}_{0,1}(z)\Big|_{z=0} = \frac{1}{2}\left(\frac{12}{(1-z)^5} - \frac{6}{(1-z)^4}\right)\Big|_{z=0} = 3, \tag{4.13}$$

$$f_{2,0,1} = \frac{1}{3!}\frac{\mathrm{d}^3}{\mathrm{d}z^3}\mathcal{F}_{0,1}(z)\Big|_{z=0} = \frac{1}{6}\left(\frac{60}{(1-z)^6} - \frac{24}{(1-z)^5}\right)\Big|_{z=0} = 6, \tag{4.14}$$

$$f_{3,0,1} = \frac{1}{4!}\frac{\mathrm{d}^4}{\mathrm{d}z^4}\mathcal{F}_{0,1}(z)\Big|_{z=0} = \frac{1}{24}\left(\frac{360}{(1-z)^7} - \frac{120}{(1-z)^6}\right)\Big|_{z=0} = 10, \tag{4.15}$$

$$f_{4,0,1} = \frac{1}{5!}\frac{\mathrm{d}^5}{\mathrm{d}z^5}\mathcal{F}_{0,1}(z)\Big|_{z=0} = \frac{1}{120}\left(\frac{2520}{(1-z)^8} - \frac{720}{(1-z)^7}\right)\Big|_{z=0} = 15, \tag{4.16}$$

$$f_{5,0,1} = \frac{1}{6!}\frac{\mathrm{d}^6}{\mathrm{d}z^6}\mathcal{F}_{0,1}(z)\Big|_{z=0} = \frac{1}{720}\left(\frac{20160}{(1-z)^9} - \frac{5040}{(1-z)^8}\right)\Big|_{z=0} = 21. \tag{4.17}$$

All the calculated values are confirmed by the program (see Tab. 4.2). It is slightly more complicated to compute the multiplicities $f_{n_1,1,1}$ of the full shuffle products $l_1^{\sqcup\!\sqcup n_1} \sqcup\!\sqcup\, l_2 \sqcup\!\sqcup\, l_3$, which map to next-to-next-to-next-to-leading-log terms in $\phi_R \circ \Phi^{-1}(X_W)$. Therefore, consider Eq. (B.25) for $i = 3$, i.e.

$$\mathcal{F}_{1,1}(z) = \left(\frac{3}{(1-z)^4} - \frac{2}{(1-z)^3}\right)\ln\left(\frac{1}{1-z}\right) - \frac{1}{(1-z)^3} + \frac{1}{(1-z)^2}. \tag{4.18}$$

Furthermore,

$$\frac{\mathrm{d}}{\mathrm{d}z}\mathcal{F}_{1,1}(z) = \left(\frac{12}{(1-z)^5} - \frac{6}{(1-z)^4}\right)\ln\left(\frac{1}{1-z}\right) + \frac{3}{(1-z)^5} - \frac{5}{(1-z)^4} + \frac{2}{(1-z)^3} \tag{4.19}$$

and the associated multiplicities (Eq. (4.2)) are

$$\begin{aligned} f_{0,1,1} = \frac{1}{2!}\frac{\mathrm{d}^2}{\mathrm{d}z^2}\mathcal{F}_{1,1}(z)\Big|_{z=0} &= \frac{1}{2}\left(\frac{60}{(1-z)^6} - \frac{24}{(1-z)^5}\right)\ln\left(\frac{1}{1-z}\right)\Big|_{z=0} + \\ &+ \frac{1}{2}\left(\frac{27}{(1-z)^6} - \frac{26}{(1-z)^5} + \frac{6}{(1-z)^4}\right)\Big|_{z=0} = \frac{7}{2}, \end{aligned} \tag{4.20}$$

$$\begin{aligned} f_{1,1,1} = \frac{1}{3!}\frac{\mathrm{d}^3}{\mathrm{d}z^3}\mathcal{F}_{1,1}(z)\Big|_{z=0} &= \frac{1}{6}\left(\frac{360}{(1-z)^7} - \frac{120}{(1-z)^6}\right)\ln\left(\frac{1}{1-z}\right)\Big|_{z=0} + \\ &+ \frac{1}{6}\left(\frac{222}{(1-z)^7} - \frac{154}{(1-z)^6} + \frac{24}{(1-z)^5}\right)\Big|_{z=0} = \frac{46}{3}, \end{aligned} \tag{4.21}$$

$$\begin{aligned} f_{2,1,1} = \frac{1}{4!}\frac{\mathrm{d}^4}{\mathrm{d}z^4}\mathcal{F}_{1,1}(z)\Big|_{z=0} &= \frac{1}{24}\left(\frac{2520}{(1-z)^8} - \frac{720}{(1-z)^7}\right)\ln\left(\frac{1}{1-z}\right)\Big|_{z=0} + \\ &+ \frac{1}{24}\left(\frac{1914}{(1-z)^8} - \frac{1044}{(1-z)^7} + \frac{120}{(1-z)^6}\right)\Big|_{z=0} = \frac{165}{4}, \end{aligned} \tag{4.22}$$

$f_0 = 1$	$f_1 = 1$	$f_2 = 1$	$f_3 = 1$	$f_4 = 1$	$f_5 = 1$
$f_{0,1} = 1$	$f_{1,1} = \frac{5}{2}$	$f_{2,1} = \frac{13}{3}$	$f_{3,1} = \frac{77}{12}$	$f_{4,1} = \frac{87}{10}$	$f_{5,1} = \frac{223}{20}$
$f_{0,0,1} = 1$	$f_{1,0,1} = 3$	$f_{2,0,1} = 6$	$f_{3,0,1} = 10$	$f_{4,0,1} = 15$	$f_{5,0,1} = 21$
$f_{0,1,1} = \frac{7}{2}$	$f_{1,1,1} = \frac{46}{3}$	$f_{2,1,1} = \frac{165}{4}$	$f_{3,1,1} = \frac{877}{10}$	$f_{4,1,1} = \frac{1617}{10}$	...

Table 4.2.: Some multiplicities of full shuffle terms calculated by the program

$$f_{3,1,1} = \frac{1}{5!}\frac{\mathrm{d}^5}{\mathrm{d}z^5}\mathcal{F}_{1,1}(z)\Big|_{z=0} = \frac{1}{120}\left(\frac{20160}{(1-z)^9} - \frac{5040}{(1-z)^8}\right)\ln\left(\frac{1}{1-z}\right)\Big|_{z=0} + \\ + \frac{1}{120}\left(\frac{17832}{(1-z)^9} - \frac{8028}{(1-z)^8} + \frac{720}{(1-z)^7}\right)\Big|_{z=0} = \frac{877}{10}, \tag{4.23}$$

$$f_{4,1,1} = \frac{1}{6!}\frac{\mathrm{d}^6}{\mathrm{d}z^6}\mathcal{F}_{1,1}(z)\Big|_{z=0} = \frac{1}{720}\left(\frac{181440}{(1-z)^{10}} - \frac{40320}{(1-z)^9}\right)\ln\left(\frac{1}{1-z}\right)\Big|_{z=0} + \\ + \frac{1}{720}\left(\frac{180648}{(1-z)^{10}} - \frac{69264}{(1-z)^9} + \frac{5040}{(1-z)^8}\right)\Big|_{z=0} = \frac{1617}{10}. \tag{4.24}$$

These values are also confirmed by the program (see Tab. 4.2). The main result is that it is always possible to find the generating functions for the discussed multiplicities of full shuffle products in the filtrated coefficients w_n (Eq. (3.37)) of the Dyson-Schwinger equation in Eq. (3.35). The way to find them is described in detail in Appendix B. It should not be too complicated to extend the described method to find the generating functions of full shuffle products to any other Dyson-Schwinger equation of a physical quantum field theory. Furthermore, it might be possible to find them for all remaining shuffle products in the coefficients w_n, which also map to next-to-...-leading-log terms in $\phi_R \circ \Phi^{-1}$. However, this is not clear yet and remains to be studied in future work.

5. Conclusions

This work shows that the log-expansion of any physical probability amplitude for a given quantum field theory is simply related to the solution of a certain Dyson-Schwinger equation by application of renormalized Feynman rules. Furthermore, the structure of the solution of a representative non-linear Dyson-Schwinger equation was investigated. In particular, a program was written in `FORM` [17], which solves the Dyson-Schwinger equation recursively in the Hopf algebra of words. Additionally, it performs a filtration of the perturbative coefficients into shuffle products of letters, s.t. each shuffle product maps to a certain power of the external scale parameter L in the log-expansion of the associated physical probability amplitude. Finally, it was observed that the multiplicities of certain filtrated terms in different perturbative coefficients of the solution of the treated representative Dyson-Schwinger equation form some series, whose generating functions were determined. For example, the multiplicities of those shuffle products, which map to leading-log terms in the log-expansion by application of renormalized Feynman rules, form a series with the generating function $\mathcal{F}(z) = \frac{1}{1-z}$. Furthermore, the multiplicities of all full shuffle products mapping to next-to-...leading-log terms in the log-expansion of the renormalized Feynman amplitude form some series and it was shown, how to determine the associated generating functions.

For future work, it would be desirable to apply the written program to other Dyson-Schwinger equations and to extend it to systems of Dyson-Schwinger equations, for example, the Q.E.D.-system discussed in Section 2.6. It should also be possible to translate the calculation of the generating functions for full shuffle terms (which belong to leading-log, next-to-leading-log terms, etc. in the log-expansion) to other (systems of) Dyson-Schwinger equations and to compare with the obtained terms of the program. Another interesting theoretical aspect is to find some analytical formulas for the remaining shuffle products in order to obtain information about the complete structure of the next-to-...-leading-log terms in the log-expansion. Finally, it would also be interesting to consider a linear Dyson-Schwinger equation because, as is trivial to observe, there are no antisymmetric parts of letters in the solution of a linear Dyson-Schwinger equation, thus, there are no occurring concatenation Lie brackets in the resulting filtration.

A. Program code and tests

The program code is written in FORM [17]. In the following, all parts of the main program Main.frm are explained and tested in full detail.

A.1. Declarations.h

This is the preamble with all declarations, see Listing A.1. For a given preprocessor variable 'n', it contains the letters l1,...,l'n' as well as the words w0,...,w'n'. These are defined to be usual functions, where the latter are contained in a set w. The functions L, R, l, r are used for left or right boundaries respectively. A function starting with a d denotes a dummy function. Thus, dl and dr are dummy functions, used as left and right boundaries. df is used as a 'final' dummy function. p, c and sh are functions for the map p defined in Eq. (3.7), the concatenation Lie bracket introduced in Section 3.3.4 and the shuffle product of words (Eq. (3.8)) respectively. A leading c in a function name denotes a commuting function, e.g. csh for commuting shuffle product or cd for commuting dummy function. A function in capital letters is a symmetric function, used for P (map p) and SH (shuffle product). Finally, i is used as an index, x, y and z for symbols and f and g as usual functions. Pay attention, when changing the order of declarations. This order determines the lexicographical ordering of letters. In particular, the Lie bracket function c must be declared after all other letters used in order to have the right Hall basis in the result.

Listing A.1: Program code for Declarations.h

```
#-
Index i;
Symbols x,y,z;
Functions P(symmetric),SH(symmetric);
Functions f,g,L,R,l,r,d,dl,dr,df;
Functions l1,...,l'n',p,c,sh;
Functions w0,...,w'n';
CFunctions csh,cd;
Set w : w0,...,w'n';
#+
```

A.2. Linearity.prc

This procedure linearizes the shuffle product, s.t. after running it, the arguments of the function `sh` are only words. Sums of words and multiplication of words by a number are performed outside the shuffle product. For example, `sh(l1+2*l2,l3)` turns into `sh(l1,l3)+2*sh(l2,l3)` after running `Linearity.prc`. Therefore, the first seven lines of the code move the summation outside the shuffle product. Lines eight to twelve move the multiplicities out of `sh`. Listing A.2 shows the procedure code. A simple, but successful test is illustrated in Listing A.3. Note that concatenation of letters is represented by a usual product in the program code, e.g. `l1*l2` represents the word $l_1 l_2$.

Listing A.2: Program code for Linearity.prc

```
#procedure Linearity
  id sh(?aa) = L*sh(?aa)*R;
  Chainout,sh;
  SplitArg,sh;
  repeat;
    id sh(x?,?aa,y?) = sh(x,?aa)+sh(y);
  endrepeat;
  id sh(x?)=l*sh*x*r;
  id sh*f?(?aa) = sh(f(?aa));
  repeat;
    id l*sh(x?)*f?!{r}(?aa)=l*sh(x*f(?aa));
  endrepeat;
  id l=1;
  id r=1;
  Chainin,sh;
  id L=1;
  id R=1;
#endprocedure
```

Listing A.3: Test for Linearity.prc

```
FORM 4.0 (Jul  2 2013) 64-bits
    #define n "9"
    #include Declarations.h
    #-
    Off statistics;
    Local expr = sh(l1+2*l2*l8,4*l3)*sh(3*l7,7*l6,l4*l4+5*l5);
    #call Linearity
    Print;

   expr = 168*sh(l2*l8,l3)*sh(l7,l6,l4*l4) + 840*sh(l2*l8,l3)*
      sh(l7,l6,l5) + 84*sh(l1,l3)*sh(l7,l6,l4*l4)
      + 420*sh(l1,l3)*sh(l7,l6,l5);

    .end
```

A.3. Psort.prc

In the first four lines, this procedure applies the associativity of p (Eq. (3.7)). The rest of the code implements commutativity of p, also defined in Eq. (3.7) (see Listing A.4). A test is postponed to the next section.

Listing A.4: Program code for Psort.prc

```
#procedure Psort
  repeat;
    id p(?aa,p(?ab),?ac) = p(?aa,?ab,?ac);
  endrepeat;
  id p(?aa) = L*P(?aa)*R;
  Chainout,P;
  id P(x?) = p(x);
  Chainin,p;
  id L = 1;
  id R = 1;
#endprocedure
```

A.4. Shffl.prc

This procedure calculates the shuffle product of words defined in Eqs. (3.8, 3.9) (see Listing A.5). Therefore, the first four lines turn each shuffle product into nested shuffle products using its associativity (Eq. (2.6)), s.t. each function `sh` has only two arguments afterward. Then, lines five to eight bring `FORM` to the innermost possible shuffle products (see the `argument` command of `FORM`, [17]). In the final `do`-loop is one more `endargument` than `argument` command, thus, it performs its contents from the innermost to the outermost shuffle product. The main content of this `do`-loop is to alternatively execute the procedure `Linearity.prc` (line 13) and the definition of the shuffle product in Eq. (3.8) (line 14 to 17 in Listing A.5). The final two statements in the `do`-loop (lines 22 and 23) represent Eq. (3.9). Two successful tests for the procedures `Psort.prc` and `Shffl.prc` are summarized in Listing A.6. Indeed,

$$\begin{aligned}
l_1 l_2 \sqcup\!\sqcup l_1 l_3 &= l_1 \left(l_2 \sqcup\!\sqcup l_1 l_3 + l_1 l_2 \sqcup\!\sqcup l_3\right) + p\left(l_1, l_1\right)\left(l_2 \sqcup\!\sqcup l_3\right) \\
&= l_1 l_2 l_1 l_3 + 2 l_1 l_1 \left(l_2 \sqcup\!\sqcup l_3\right) + l_1 l_3 l_1 l_2 + l_1 p\left(l_2, l_1\right) l_3 + l_1 p\left(l_1, l_3\right) l_2 + \\
&\quad + p\left(l_1, l_1\right)\left(l_2 l_3 + l_3 l_2 + p\left(l_2, l_3\right)\right) \\
&= 2 l_1 l_1 l_2 l_3 + 2 l_1 l_1 l_3 l_2 + 2 l_1 l_1 p\left(l_2, l_3\right) + l_1 l_2 l_1 l_3 + l_1 l_3 l_1 l_2 + l_1 p\left(l_1, l_2\right) l_3 + \\
&\quad + l_1 p\left(l_1, l_3\right) l_2 + p\left(l_1, l_1\right) l_2 l_3 + p\left(l_1, l_1\right) l_3 l_2 + p\left(l_1, l_1\right) p\left(l_2, l_3\right), \qquad \text{(A.1)} \\
l_1 \sqcup\!\sqcup l_1 \sqcup\!\sqcup l_2 &= \left(2 l_1 l_1 + p\left(l_1, l_1\right)\right) \sqcup\!\sqcup l_2 \\
&= 2 l_1 \left(l_1 \sqcup\!\sqcup l_2\right) + 2 l_2 l_1 l_1 + 2 p\left(l_1, l_2\right) l_1 + p\left(l_1, l_1\right) l_2 + l_2 p\left(l_1, l_1\right) + p\left(l_1, l_1, l_2\right) \\
&= 2 l_1 l_1 l_2 + 2 l_1 l_2 l_1 + 2 l_1 p\left(l_1, l_2\right) + 2 l_2 l_1 l_1 + l_2 p\left(l_1, l_1\right) + p\left(l_1, l_1\right) l_2 + \\
&\quad + p\left(l_1, l_1, l_2\right) + 2 p\left(l_1, l_2\right) l_1. \qquad \text{(A.2)}
\end{aligned}$$

Listing A.5: Program code for Shffl.prc

```
#procedure Shffl
  repeat;
    id sh(x?,?aa,y?,z?) = sh(x,?aa,sh(y,z));
  endrepeat;
  #do k = 1, {'n'+1}
    argument;
  #enddo
  ;
  #do k = 1, {'n'+1}
    id f?(?aa) = d(f(?aa));
    Chainin,d;
    endargument;
    #call Linearity
    repeat;
      id sh(d(f?(?aa),?ab),d(g?(?ac),?ad)) = f(?aa)*sh(d(?ab),d(g(?
         ac),?ad)) + g(?ac)*sh(d(f(?aa),?ab),d(?ad)) + p(f(?aa),g(?
         ac))*sh(d(?ab),d(?ad));
      id sh(?aa,d,?ab) = sh(?aa,?ab);
    endrepeat;
    argument;
      Chainout,d;
      id d(x?) = x;
    endargument;
    id sh = 1;
    id sh(x?) = x;
  #enddo
  ;
#endprocedure
```

Listing A.6: Test for Shffl.prc

```
FORM 4.0 (Jul  2 2013) 64-bits
    #define n "4"
    #include Declarations.h
    #-
    Off statistics;
    Local expr = sh(l1*l2,l1*l3);
    Local expr2 = sh(l1,l1,l2);
    #call Shffl
    #call Psort
    .sort
    Print;

   expr =
      2*l1*l1*l2*l3 + 2*l1*l1*l3*l2 + 2*l1*l1*p(l2,l3) + l1*l2*l1*l3
      + l1*l3*l1*l2 + l1*p(l1,l2)*l3 + l1*p(l1,l3)*l2
```

```
16        + p(l1,l1)*l2*l3 + p(l1,l1)*l3*l2 + p(l1,l1)*p(l2,l3);

18     expr2 =
          2*l1*l1*l2 + 2*l1*l2*l1 + 2*l1*p(l1,l2) + 2*l2*l1*l1
20        + l2*p(l1,l1) + p(l1,l1)*l2 + p(l1,l1,l2) + 2*p(l1,l2)*l1;

22      .end
    0.00 sec out of 0.00 sec
```

A.5. Nthword.prc

This procedure applies Eq. (3.37) once for every occurring word w_n (see Listing A.7). Therefore, the occurring functions `f` abbreviate

$$\mathtt{f}(n-m, m+1) = \sum_{k_1+\ldots k_{m+1}=n-m, k_i \geq 0} w_{k_1} \sqcup\!\sqcup \ldots \sqcup\!\sqcup w_{k_{m+1}}. \tag{A.3}$$

In lines three to nine the shuffle product is written via a dummy function `d`, e.g. $w_1 \sqcup\!\sqcup w_2$ is written as `d*w1*d*w2`. This is translated into `sh(w1,w2)` in lines 10 to 13. A test is postponed to the next section.

Listing A.7: Program code for Nthword.prc

```
1 #procedure Nthword(n)
    id w'n'= <l1*f({'n'-1},2)>+...+<l'n'*f(0,{'n'+1})>;
3   repeat;
      #do k = 0, {'n'-1}
5        if (match(f('k',1))) id f('k',1) = d*w'k';
         id f('k',i?) = <d*w0*f('k',i-1)>+...+<d*w'k'*f(0,i-1)>;
7     #enddo
       ;
9   endrepeat;
    id d*w0 = 1;
11  id d*f?w = csh(f);
    Chainin,csh;
13  id csh(?aa) = sh(?aa);
  #endprocedure
```

A.6. Word.prc

The procedure, which computes the word w_n via the recursive application of Eq. (3.37) is `Word.prc`, see Listing A.8. It only contains calls of the procedures introduced in Appendices A.2 to A.5. Thus, there is no need for further explanation of the program code. A test for `Word.prc` and `Nthword.prc` is given in Listing A.9. It returns the word w_3 which matches the result given in Eq. (3.40).

Listing A.8: Program code for Word.prc

```
#procedure Word
  Local [w_n] = w'n';
  #call Nthword('n');
  .sort
  #do k = 1, 'n'
    argument;
    #do l = 1, 'n'
      #call Nthword('l')
    #enddo
    ;
  #enddo
  ;
  #do k = 1, 'n'
    endargument;
    #call Linearity
    #call Shffl
  #enddo
  ;
  .sort
  #call Psort
  .sort
#endprocedure
```

Listing A.9: Test for Word.prc

```
FORM 4.0 (Jul  2 2013) 64-bits
    #define n "3"
    #include Declarations.h
    #-
    Off statistics;
    #call Word
    Print;

   [w_n] =
      6*l1*l1*l1 + 2*l1*l2 + l1*p(l1,l1) + 3*l2*l1 + l3;

    .end
  0.00 sec out of 0.00 sec
```

A.7. Jacobi.prc

This procedure implements the Jacobi identity (Eq. (3.41)), s.t. only Hall basis Lie brackets enter the result for w_n. The conditions for Hall basis elements are given in Proposition 3.3.3. These are automatically fulfilled by the following method. Each time,

the procedure `Sorting.prc` creates a concatenation Lie bracket $[l_1, l_2]$ with lexicographical ordering $l_1 < l_2 \in \mathcal{H}_L$, it will call `Jacobi.prc` in order to transform every occurring $[l_1, [l_2, l_3]]$-term with lexicographical ordering $l_1 < l_2 < l_3 \in \mathcal{H}_L$ into $[l_2, [l_1, l_3]] - [l_3, [l_1, l_2]]$ (see Eq. (3.41)). In the second line of the program code (see Listing A.10), each occurring $[l_1, [l_2, l_3]]$-term is transformed into a dummy function `d(a,b,c)`. The `PolyFun` command [17] in the third line realizes that every term of the expression becomes positive, the signs and multiplicities are conserved in a commutative dummy function `cd`. Line six performs a cyclic permutation of the arguments of the `d`-functions in the expression `[dummy]`, which is a copy of the `[w_n]`-expression. Afterwards, the lexicographically smallest letter is at position one. Lines 9 to 13 construct an expression, which includes only terms with basis Lie brackets of the originally considered expression `[w_n]`. Lines 16 to 18 create an expression, which includes only the terms with non-basis Lie brackets of `[w_n]`. Furthermore, in line 18, these terms are transformed into Hall basis terms using the Jacobi identity (Eq. (3.41)). Finally, lines 20 to 25 write `[w_n]` in terms of only Hall basis terms including the back-transformation from `d(a,b,c)` into $[l_1, [l_2, l_3]]$. In particular, line 23 ensures that the procedure works even in the case that two of the three arguments of the `d`-function are equal. Listing A.11 gives a short and successful test of the procedure. This can be easily verified using the Jacobi identity (Eq. (3.41)). Furthermore, the output does not contain any term $[l_1, [l_2, l_3]]$ with lexicographical ordering $l_1 < l_2 < l_3 \in \mathcal{H}_L$, which would be a non-basis term.

Listing A.10: Program code for Jacobi.prc

```
1 #procedure Jacobi
    id c(x?,c(y?,z?)) = d(x,y,z);
3   .sort (PolyFun = cd);
    Skip,[w_n];
5   Local [dummy] = [w_n];
      Cyclesymmetrize,d;
7   .sort
    Skip,[w_n],[dummy];
9   Local [basic-terms] = [w_n] - [dummy];
    .sort
11  Skip,[w_n],[dummy];
      if (coefficient <0) discard;
13  .sort
    Drop,[dummy];
15  Skip,[w_n],[basic-terms];
    Local [non-basic-terms] = [w_n] - [basic-terms];
17    id d(x?,y?,z?) = d(y,x,z) - d(z,x,y);
    .sort
19  Drop,[w_n];
    Skip,[basic-terms],[non-basic-terms];
21  Local [w_n] = [basic-terms] + [non-basic-terms];
      id cd(x?) = x;
23    id d(x?,y?,y?) = 0;
```

```
    id d(x?,y?,z?) = c(x,c(y,z));
  .sort
  Drop,[basic-terms],[non-basic-terms];
  .sort
#endprocedure
```

Listing A.11: Test for Jacobi.prc

```
FORM 4.0 (Jul  2 2013) 64-bits
    #define n "3"
    #include Declarations.h
    #-
    Off statistics;
    Local [w_n] = 2*c(l1,c(l2,l3)) - c(l2,c(l1,l3));
    #call Jacobi
    Print;
    .end

   [w_n] =
      c(l2,c(l1,l3)) - 2*c(l3,c(l1,l2));

  0.00 sec out of 0.00 sec
```

A.8. Sorting.prc

This procedure performs the filtration of a word `[w_n]` as described in Section 3.3.4 (see Listing A.12). It consists of a loop over the length of words (`'k'`-loop), beginning with the biggest possible length n down to length two. Lines four and five transform each word of the considered length into a dummy function `d` with the letters of the word as its arguments. Furthermore, each word is split into three dummy functions `dl`, `dr` and `d`, s.t. `dl` (u_1) `*dr` (u_3) `*d` (u_2) represents the word $u = u_1u_2u_3$. This splitting starts in line six. The following loop from line 8 to line 31 represents a bubble sort. First, lines eleven and twelve bring the last two arguments of the `d`-functions of the expression `[dummy]`, which is a copy of the `[w_n]`-expression, into the lexicographically proper order. In line 15, a new expression is created, which is the difference between the previous two expressions. This expression, namely `[w_n-dummy]`, vanishes, if in each term of `[w_n]` the last two arguments of the `d`-function are already lexicographically ordered. If it is non-zero, lines 19 to 23 replace the occurring differences by a concatenation Lie bracket function `c`, which denotes a new letter, and reframe the words, including the newly created `c`-functions out of `dl` (u_1) `*dr` (u_3) `*d` (u_2). The lengths of these words are lowered by one and thus, they are not considered before the next `k`-loop starts. Lines 26 to 29 rebuild `[w_n]` out of `[w_n-dummy]` and `[dummy]`. In the resulting expression the last two arguments of the `d`-functions are lexicographically ordered. Thus, the last arguments are moved to the first positions of the `dr`-functions. If there is only one more argument in a `d`-function left, then it is fixed to the last position of the `dl`-function and the content of

the `dr`-function is removed to the `d`-function in order to start a new period of the bubble sort. After the full bubble sort in lines 8 to 32 was performed, the considered expression `[w_n]` only consists of words with length lowered by one using the new commutator functions plus one `dl`-function, which represents a word with the full length considered in the current `k`-loop. The letters of this word are lexicographically ordered. In line 36 a full shuffle `sh` of the `dl`-function arguments is subtracted of the expression, whereas the same full shuffle is added, but denoted by `SH`. The capital function `SH` is fixed and will be carried to the output without further modification. It represents a shuffle product of letters and maps to $L^{'\mathtt{n}'-'\mathtt{k}'}$ by application of renormalized Feynman rules $\phi_R \circ \Phi^{-1}$, where `'n'-'k'` is the number of arguments in the function `SH`. The minor function `sh` is calculated using `Shffl.prc` and `Psort.prc` in lines 38 and 40. The point is that a second performance of lines 3 to 43 yields the expression `[w_n]` in terms of a sum of words with lengths lowered by one, plus a full shuffle `SH`, plus two functions `dl`, which are the same except for the sign and thus, cancel. The cancellation can be easily seen by a look at Eq. (B.3). The last step is to call `Jacobi.prc` in line 44 as explained in the previous section. After the full `'k'`-loop over the length of words is performed, the expression `[w_n]` is filtrated. A test of the procedure is depicted in Listing A.13. Indeed,

$$
\begin{aligned}
[\mathtt{w\_n}] &= -\frac{1}{2}l_1 \sqcup\!\sqcup [l_3,l_4] - \frac{1}{2}l_3 \sqcup\!\sqcup [l_1,l_4] + \frac{1}{2}l_4 \sqcup\!\sqcup [l_1,l_3] + \frac{1}{2}p\left(l_1,[l_3,l_4]\right) + \\
&\quad + \frac{1}{2}p\left(l_3,[l_1,l_4]\right) - \frac{1}{2}p\left(l_4,[l_1,l_3]\right) \\
&= -\frac{1}{2}l_1\,[l_3,l_4] - \frac{1}{2}\,[l_3,l_4]\,l_1 - \frac{1}{2}l_3\,[l_1,l_4] - \frac{1}{2}\,[l_1,l_4]\,l_3 + \frac{1}{2}l_4\,[l_1,l_3] + \frac{1}{2}\,[l_1,l_3]\,l_4 \\
&= -\frac{1}{2}l_1l_3l_4 + \frac{1}{2}l_1l_4l_3 - \frac{1}{2}l_3l_4l_1 + \frac{1}{2}l_4l_3l_1 - \frac{1}{2}l_3l_1l_4 + \frac{1}{2}l_3l_4l_1 - \frac{1}{2}l_1l_4l_3 + \\
&\quad + \frac{1}{2}l_4l_1l_3 + \frac{1}{2}l_4l_1l_3 - \frac{1}{2}l_4l_3l_1 + \frac{1}{2}l_1l_3l_4 - \frac{1}{2}l_3l_1l_4 \\
&= +l_4l_1l_3 - l_3l_1l_4. \qquad \text{(A.4)}
\end{aligned}
$$

Furthermore, the terms in the output of Listing A.13 are well filtrated as described in Section 3.3.4.

Listing A.12: Program code for Sorting.prc

```
#procedure Sorting
  #do k = 0, {'n'-2}
    #do l = 1, 2
      id <l1?(?aa1)>*...*<l{'n'-'k'}?(?aa{'n'-'k'})> = <d(l1(?aa1))
          >*...*<d(l{'n'-'k'}(?aa{'n'-'k'}))>;
      Chainin,d;
      id d(?aa) = dl*dr*d(?aa);
      .sort
      #do m = 1, {{'n'-'k'}*{'n'-'k'-1}/2+1}
        Skip,[w_n];
        Local [dummy] = [w_n];
          id d(?aa,x?,y?) = d(?aa)*cd(x)*cd(y);
```

```
            id d(?aa)*cd(x?)*cd(y?) = d(?aa,x,y);
          .sort
          Skip,[w_n],[dummy];
          Local [w_n-dummy] = [w_n]-[dummy];
          .sort
          Drop,[w_n];
          Skip,[dummy];
            if (coefficient<0) discard;
            id dl(?aa)*dr(?ab)*d(?ac,x?,y?) = df(?aa,?ac,c(x,y),?ab);
            Chainout,df;
            id df(x?) = x;
            Antisymmetrize c;
          .sort
          Skip,[w_n-dummy],[dummy];
          Local [w_n] = [w_n-dummy]+[dummy];
            id dr(?aa)*d(?ab,x?,y?) = dr(y,?aa)*d(?ab,x);
            id dl(?aa)*dr(?ab)*d(x?) = dl(?aa,x)*dr*d(?ab);
          .sort
          Drop,[w_n-dummy],[dummy];
        #enddo
        ;
        id dr = 1;
        id d = 1;
        .sort
        id dl(?aa) = dl(?aa) - 1/fac_({'n'-'k'})*sh(?aa) + 1/fac_({'n
            '-'k'})*SH(?aa);
        .sort
        #call Shffl
        .sort
        #call Psort
        .sort
      #enddo
      ;
      #call Jacobi
    #enddo
    ;
#endprocedure
```

Listing A.13: Test for Sorting.prc

```
FORM 4.0 (Jul 2 2013) 64-bits
    #define n "4"
    #include Declarations.h
    #-
    Off statistics;
    Local [w_n] = l4*l1*l3 - l3*l1*l4;
    #call Sorting
```

```
    .sort
    Print;

  [w_n] =
      - 1/2*SH(l1,c(l3,l4)) - 1/2*SH(l3,c(l1,l4))
      + 1/2*SH(l4,c(l1,l3)) + 1/2*p(l1,c(l3,l4))
      + 1/2*p(l3,c(l1,l4)) - 1/2*p(l4,c(l1,l3));

    .end
 0.01 sec out of 0.01 sec
```

A.9. Main.frm

This is the main program. It calls the procedures `Word.prc`, which creates the word w_n for a given n, and `Sorting.prc`, which performs the filtration described in Section 3.3.4. These procedures are explained in Sections A.6 and A.8, thus, no further explanation is needed. Program tests are performed in Section 4.1. The program code is depicted in Listing A.14 for the representing case `'n'`$= 5$.

Listing A.14: Program code for Main.frm

```
#define n "3"
#include Declarations.h
Off statistics;
#call Word
#call Sorting
Print;
.end
```

B. Generating functions for the multiplicities of full shuffle products in w_n

This last part of the work describes a way to calculate the generating functions for all multiplicities $f_{n_1,n_2,\ldots}$ of full shuffle products in the words w_n, as introduced in Section 4.2. First, consider a full shuffle product of $n_1 + n_2 + \ldots$ letters, $l_1^{\sqcup\!\sqcup n_1} \sqcup\!\sqcup\, l_2^{\sqcup\!\sqcup n_2} \sqcup\!\sqcup \ldots$, $n_i \in \mathbb{N}_0$ in a filtrated word $w_{n_1+2n_2+3n_3+\ldots}$ (Eq. (3.37)). Performing the shuffle product (Eq. (3.8)) yields $(n_1+n_2+\ldots)!$ words with length $n_1+n_2+\ldots$ consisting of n_1 letters l_1, n_2 letters l_2 etc. in any order, plus words with length smaller than $n_1 + n_2 + \ldots$. Here, the length of a word denotes the number of its letters. It is further possible to transform $l_1^{\sqcup\!\sqcup n_1} \sqcup\!\sqcup\, l_2^{\sqcup\!\sqcup n_2} \sqcup\!\sqcup \ldots$, $n_i \in \mathbb{N}_0$ into

$$(n_1 + n_2 + \ldots)! \underbrace{l_1 \ldots l_1}_{n_1 \text{ times}} \underbrace{l_2 \ldots l_2}_{n_2 \text{ times}} l_3 \ldots + \text{ words with length } < (n_1 + n_2 + \ldots) \tag{B.1}$$

by replacing $l_i l_j$ for $i > j$ by $[l_i, l_j] + l_j l_i$ a finite number of times in each word. Thereby, the number $(n_1 + n_2 + \ldots)!$ is the number of permutations of $n_1 + n_2 + \ldots$ letters in a word without repetition. On the other hand, consider a set $(n_1, n_2, n_3 \ldots)$, $n_i \in \mathbb{N}_0$ and the non-filtrated word $w_{n_1+2n_2+3n_3+\ldots}$ (Eq. (3.37)). Denote the number of words in $w_{n_1+2n_2+3n_3+\ldots}$ consisting of n_1 letters l_1, n_2 letters l_2 etc. in any order by $F_{n_1,n_2,n_3,\ldots}$. Then, it is possible to transform all these $F_{n_1,n_2,n_3,\ldots}$ words into

$$F_{n_1,n_2,n_3,\ldots} \underbrace{l_1 \ldots l_1}_{n_1 \text{ times}} \underbrace{l_2 \ldots l_2}_{n_2 \text{ times}} l_3 \ldots + \text{ words with length } < (n_1 + n_2 + \ldots) \tag{B.2}$$

by replacing $l_i l_j$ for $i > j$ by $[l_i, l_j] + l_j l_i$ a finite number of times in each word. Comparison of Eqs. (B.1, B.2) yields a relation between the multiplicity of the full shuffle term $l_1^{\sqcup\!\sqcup n_1} \sqcup\!\sqcup\, l_2^{\sqcup\!\sqcup n_2} \sqcup\!\sqcup \ldots$ in the resulting filtrated word $w_{n_1+2n_2+3n_3+\ldots}$ and the number of words consisting of n_1 letters l_1, n_2 letters l_2 etc. in any order in the non-filtrated word $w_{n_1+2n_2+3n_3+\ldots}$, i.e.

$$f_{n_1,n_2,n_3,\ldots} = \frac{F_{n_1,n_2,n_3,\ldots}}{(n_1 + n_2 + n_3 + \ldots)!}. \tag{B.3}$$

Now, let $N(u)$ be the length of a word $u \in \mathcal{H}_W$. Then, for any words $u_1, u_2, \ldots, u_n \in \mathcal{H}_W$, the number of different words consisting of the letters of $u_1, u_2, \ldots, u_n$ in the shuffle product $u_1 \sqcup\!\sqcup\, u_2 \sqcup\!\sqcup \ldots \sqcup\!\sqcup\, u_n$ is $\frac{(N(u_1)+N(u_2)+\ldots+N(u_n))!}{N(u_1)!N(u_2)!\ldots N(u_n)!}$. This is the number of permutations

with repetition and is needed in the following recursive formula for $F_{n_1,n_2,n_3\ldots}$, which can be directly obtained from the definition of w_n in Eq. (3.37), i.e.

$$\begin{aligned} F_{n_1,n_2,n_3,\ldots} &= \sum_{m\geq 1}^{m\leq \sum_j jn_j} (1-\delta_{n_m 0}) \sum_{k_i\geq 0}^{k_1+\ldots k_{m+1}=\sum_j jn_j - m} \sum_{\sum_j jt_i^j = k_i}^{\sum_{i\geq 1}^{i\leq m+1} t_i^j = n_j - \delta_{mj}} \\ &\left(\frac{\left(\sum_{i\geq 1}^{i\leq m+1} \sum_j t_i^j\right)!}{\left(\sum_j t_1^j\right)! \left(\sum_j t_2^j\right)! \ldots \left(\sum_j t_{m+1}^j\right)!} F_{t_1^1,t_1^2,t_1^3,\ldots} F_{t_2^1,t_2^2,t_2^3,\ldots} \ldots F_{t_{m+1}^1,t_{m+1}^2,t_{m+1}^3,\ldots} \right). \end{aligned} \tag{B.4}$$

The nominator on the R.H.S. of Eq. (B.4) is equal to $\left(\sum_j n_j\right) - 1$. Thus, using Eq. (B.3) yields

$$\begin{aligned} f_{n_1,n_2,n_3,\ldots} \sum_j n_j &= \sum_{m\geq 1}^{m\leq \sum_j jn_j} (1-\delta_{n_m 0}) \sum_{k_i\geq 0}^{k_1+\ldots k_{m+1}=\sum_j jn_j - m} \sum_{\sum_j jt_i^j = k_i}^{\sum_{i\geq 1}^{i\leq m+1} t_i^j = n_j - \delta_{mj}} \\ &\left(f_{t_1^1,t_1^2,t_1^3,\ldots} f_{t_2^1,t_2^2,t_2^3,\ldots} \ldots f_{t_{m+1}^1,t_{m+1}^2,t_{m+1}^3,\ldots} \right). \end{aligned} \tag{B.5}$$

Now, it is convenient to multiply Eq. (B.5) by $z^{\left(\sum_j n_j\right)-1}$, where $z \in \mathbb{R}$, $0 \leq z < 1$, which gives

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}z}\left(f_{n_1,n_2,n_3,\ldots} z^{\sum_j n_j}\right) &= \sum_{m\geq 1}^{m\leq \sum_j jn_j} (1-\delta_{n_m 0}) \sum_{k_i\geq 0}^{k_1+\ldots k_{m+1}=\sum_j jn_j - m} \sum_{\sum_j jt_i^j = k_i}^{\sum_{i\geq 1}^{i\leq m+1} t_i^j = n_j - \delta_{mj}} \\ &\left(f_{t_1^1,t_1^2,t_1^3,\ldots} z^{\sum_j t_1^j} f_{t_2^1,t_2^2,t_2^3,\ldots} z^{\sum_j t_2^j} \ldots f_{t_{m+1}^1,t_{m+1}^2,t_{m+1}^3,\ldots} z^{\sum_j t_{m+1}^j} \right). \end{aligned} \tag{B.6}$$

As can be easily seen, in the summation on the R.H.S. of Eq. (B.6), the condition $\sum_{i\geq 1}^{i\leq m+1} t_i^j = n_j - \delta_{mj}$ implies the conditions $k_i \geq 0$, $\sum_j jt_i^j = k_i$ and $k_1 + \ldots k_{m+1} = \sum_j jn_j - m$. Thus, Eq. (B.6) becomes

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}z}\left(f_{n_1,n_2,n_3,\ldots} z^{\sum_j n_j}\right) &= \sum_{m\geq 1}^{m\leq \sum_j jn_j} (1-\delta_{n_m 0}) \sum_{\sum_{i\geq 1}^{i\leq m+1} t_i^j = n_j - \delta_{mj}} \\ &\left(f_{t_1^1,t_1^2,t_1^3,\ldots} z^{\sum_j t_1^j} f_{t_2^1,t_2^2,t_2^3,\ldots} z^{\sum_j t_2^j} \ldots f_{t_{m+1}^1,t_{m+1}^2,t_{m+1}^3,\ldots} z^{\sum_j t_{m+1}^j} \right). \end{aligned} \tag{B.7}$$

The final step is to introduce the generating function of $f_{n_1,n_2,n_3,\ldots}$, i.e.

$$\mathcal{F}_{n_2,n_3,\ldots}(z) = \sum_{n_1\geq 0} f_{n_1,n_2,n_3,\ldots} z^{\sum_j n_j}. \tag{B.8}$$

It fulfills the boundary condition

$$\mathcal{F}_{n_2,n_3,\ldots}(0) = \delta_{n_2 0}\delta_{n_3 0}\ldots \tag{B.9}$$

and the multiplicities of the full shuffle terms are given by

$$f_{n_1,n_2,n_3,\ldots} = \frac{1}{(n_1+n_2+\ldots)!}\frac{\mathrm{d}^{n_1+n_2+\ldots}}{\mathrm{d}z^{n_1+n_2+\ldots}}\mathcal{F}_{n_2,n_3,\ldots}(z)\Big|_{z=0}. \tag{B.10}$$

Insertion of Eq. (B.8) into the summation of Eq. (B.7) over n_1 finally yields

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}z}\mathcal{F}_{n_2,n_3,\ldots}(z) &= \sum\nolimits_{m\geq 1}\sum\nolimits_{t_1^j+\ldots+t_{m+1}^j=n_j-\delta_{mj}}^{t_i^j\geq 0}\mathcal{F}_{t_1^2,t_1^3,\ldots}(z)\ldots\mathcal{F}_{t_{m+1}^2,t_{m+1}^3,\ldots}(z)\times \\ &\quad\times\begin{cases}1 & m=1\\ 1-\delta_{n_m0} & m\neq 1\end{cases}.\end{aligned} \tag{B.11}$$

This is an inhomogeneous ordinary first order differential equation, which can be exactly solved. Consider for example the case $n_2 = n_3 = \ldots = 0$. Then, Eq. (B.11) reads

$$\frac{\mathrm{d}}{\mathrm{d}z}\mathcal{F}(z) = \mathcal{F}(z)^2. \tag{B.12}$$

Together with the boundary condition $\mathcal{F}(0) = 1$ (Eq. (B.9)) the solution of Eq. (B.12) is

$$\mathcal{F}(z) = \frac{1}{1-z}. \tag{B.13}$$

Now, in the R.H.S. of Eq. (B.11), there are two terms containing the function $\mathcal{F}_{n_2,n_3,\ldots}$ itself, i.e. the $\left(m=1, t_1^j=0, t_2^j=n_j\right)$- and $\left(m=1, t_1^j=n_j, t_2^j=0\right)$-terms. Writing these two terms explicitly and using Eq. (B.13) yields

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}z}\mathcal{F}_{n_2,n_3,\ldots}(z) &= \frac{2}{1-z}\mathcal{F}_{n_2,n_3,\ldots} + \sum\nolimits_{m\geq 1}\sum\nolimits_{t_1^j+\ldots+t_{m+1}^j=n_j-\delta_{mj}}^{t_i^j\geq 0,\sum_j t_i^j<\sum_j n_j} \\ &\quad\left(\mathcal{F}_{t_1^2,t_1^3,\ldots}(z)\ldots\mathcal{F}_{t_{m+1}^2,t_{m+1}^3,\ldots}(z)\right)\begin{cases}1 & m=1\\ 1-\delta_{n_m0} & m\neq 1\end{cases}.\end{aligned} \tag{B.14}$$

Using variation of constants, the inhomogeneous ordinary first order differential equation in Eq. (B.14) is solved by the Ansatz

$$\mathcal{F}_{n_2,n_3,\ldots}(z) = \frac{\mathcal{C}_{n_2,n_3,\ldots}(z)}{(1-z)^2}. \tag{B.15}$$

Then, the function $\mathcal{C}_{n_2,n_3,\ldots}(z)$ must fulfill

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}z}\mathcal{C}_{n_2,n_3,\ldots}(z) &= \sum\nolimits_{m\geq 1}\frac{1}{(1-z)^{2m}}\sum\nolimits_{t_1^j+\ldots+t_{m+1}^j=n_j-\delta_{mj}}^{t_i^j\geq 0,\sum_j t_i^j<\sum_j n_j}\mathcal{C}_{t_1^2,t_1^3,\ldots}(z)\ldots\mathcal{C}_{t_{m+1}^2,t_{m+1}^3,\ldots}(z)\times \\ &\quad\times\begin{cases}1 & m=1\\ 1-\delta_{n_m0} & m\neq 1\end{cases}.\end{aligned} \tag{B.16}$$

The condition $\sum_j t_i^j < \sum_j n_j$ on the R.H.S. of Eq. (B.16) ensures that there is no occurring term $\mathcal{C}_{n_2,n_3,\ldots}(z)$ in the sum. Thus, all the functions $\mathcal{C}_{n_2,n_3,\ldots}(z)$ can be recursively obtained by an integration of the R.H.S. of Eq. (B.16) over z. The integration constant is uniquely defined by Eq. (B.9). Thus, the generating functions for the multiplicities $f_{n_1,n_2,n_3,\ldots}$ of full shuffle products in the words w_n (Eq. (3.37)) are given by Eq. (B.15). Consider two examples. First, let $i > 1$ be an integer. Then, in a given filtrated word w_n, there will be one full shuffle product $l_1 \sqcup\!\sqcup \ldots \sqcup\!\sqcup l_1 \sqcup\!\sqcup l_i$. Its corresponding multiplicity is $f_{n-i,0,\ldots,0,n_i=1}$ and its generating function is $\mathcal{F}_{0,\ldots,0,n_i=1}$. In order to calculate $\mathcal{F}_{0,\ldots,0,n_i=1}$, Eq. (B.16) needs to be integrated. The given values $n_j = \delta_{ij}$ yield in

$$\frac{\mathrm{d}}{\mathrm{d}z}\mathcal{C}_{0,\ldots,0,n_i=1}(z) = \frac{1}{(1-z)^{2i}}(\mathcal{C}(z))^{i+1}, \tag{B.17}$$

where $\mathcal{C}(z)$ is given by Eqs. (B.13, B.15) to be $(1-z)$. Thus,

$$\frac{\mathrm{d}}{\mathrm{d}z}\mathcal{C}_{0,\ldots,0,n_i=1}(z) = \frac{1}{(1-z)^{i-1}} \tag{B.18}$$

and the integration over z gives

$$\mathcal{C}_{0,\ldots,0,n_i=1}(z) = \begin{cases} -\ln(1-z) + C_1 & i = 2 \\ \frac{1}{(i-2)(1-z)^{i-2}} + C_2 & i > 2 \end{cases}, \tag{B.19}$$

where $C_{1,2}$ denote the integration constants, which are determined by Eq. (B.15) and the boundary condition in Eq. (B.9) to be $C_1 = 0$ and $C_2 = \frac{1}{2-i}$. Thus, using Eqs. (B.15, B.19) the generating functions for $f_{n-i,0,\ldots,0,n_i=1}$ are

$$\mathcal{F}_{0,\ldots,0,n_i=1}(z) = \begin{cases} -\frac{\ln(1-z)}{(1-z)^2}, & i = 2 \\ \frac{1}{i-2}\left(\frac{1}{(1-z)^i} - \frac{1}{(1-z)^2}\right), & i > 2 \end{cases}. \tag{B.20}$$

The second example relies on these results. Let $i > 2$ be an integer. Consider the multiplicity $f_{n-i-2,1,0,\ldots,0,n_i=1}$ of the full shuffle product $l_1 \sqcup\!\sqcup \ldots \sqcup\!\sqcup l_1 \sqcup\!\sqcup l_2 \sqcup\!\sqcup l_i$ in the filtrated word w_n. Its generating function $\mathcal{F}_{1,0,\ldots,0,n_i=1}$ is determined by Eq. (B.15) using Eq. (B.16), i.e.

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}z}\mathcal{C}_{1,0,\ldots,0,n_i=1}(z) &= \frac{2}{(1-z)^2}\mathcal{C}_{0,\ldots,0,n_i=1}(z)\mathcal{C}_1(z) + \frac{i+1}{(1-z)^{2i}}(\mathcal{C}(z))^i\mathcal{C}_1(z) + \\ &\quad + \frac{3}{(1-z)^4}(\mathcal{C}(z))^2\mathcal{C}_{0,\ldots,0,n_i=1}(z). \end{aligned} \tag{B.21}$$

Inserting $\mathcal{C}(z) = 1 - z$ and Eq. (B.19) yields

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}z}\mathcal{C}_{1,0,\ldots,0,n_i=1}(z) &= -\frac{2\ln(1-z)}{(i-2)(1-z)^i} + \frac{2\ln(1-z)}{(i-2)(1-z)^2} - \frac{i+1}{(1-z)^i}\ln(1-z) + \\ &\quad + \frac{3}{(i-2)(1-z)^i} - \frac{3}{(i-2)(1-z)^2}. \end{aligned} \tag{B.22}$$

In order to integrate the R.H.S. of Eq. (B.22), the primitive integral

$$\int \frac{\ln(1-z)}{(1-z)^j}\mathrm{d}z = \frac{\ln(1-z)}{(j-1)(1-z)^{j-1}} + \frac{1}{(j-1)^2(1-z)^{j-1}} \tag{B.23}$$

is needed for any integer $j > 1$. Eq. (B.23) was obtained by partial integration. Thus, integration of Eq. (B.22) over z yields

$$\begin{aligned}
\mathcal{C}_{1,0,\ldots,0,n_i=1}(z) &= \frac{2}{i-2}\left(\frac{-\ln(1-z)}{(i-1)(1-z)^{i-1}} - \frac{1}{(i-1)^2(1-z)^{i-1}} + \frac{\ln(1-z)+1}{(1-z)}\right) - \\
&\quad -(i+1)\left(\frac{\ln(1-z)}{(i-1)(1-z)^{i-1}} + \frac{1}{(i-1)^2(1-z)^{i-1}}\right) + \\
&\quad + \frac{3}{(i-2)(i-1)(1-z)^{i-1}} - \frac{3}{(i-2)(1-z)} + C \\
&= \left(-\frac{i}{i-2}\frac{1}{(1-z)^{i-1}} + \frac{2}{i-2}\frac{1}{(1-z)}\right)\ln(1-z) - \\
&\quad - \frac{i-3}{(i-2)(i-1)}\frac{1}{(1-z)^{i-1}} - \frac{1}{i-2}\frac{1}{(1-z)} + C,
\end{aligned} \tag{B.24}$$

where C denotes the integration constant. Using the boundary condition in Eq. (B.9) together with Eq. (B.15) determines $C = \frac{2}{i-1}$. Thus, the generating functions for $f_{n-i-2,1,0,\ldots,0,n_i=1}$ are finally obtained by Eqs. (B.15, B.24) to be

$$\begin{aligned}
\mathcal{F}_{1,0,\ldots,0,n_i=1}(z) &= \left(-\frac{i}{i-2}\frac{1}{(1-z)^{i+1}} + \frac{2}{i-2}\frac{1}{(1-z)^3}\right)\ln(1-z) - \\
&\quad - \frac{i-3}{(i-2)(i-1)}\frac{1}{(1-z)^{i+1}} - \frac{1}{i-2}\frac{1}{(1-z)^3} + \frac{2}{i-1}\frac{1}{(1-z)^2}.
\end{aligned} \tag{B.25}$$

The point is that every generating function can be recursively obtained in the way described above. Of course, they become very complicated for more $n_j \neq 0$, but it is still possible to use a computer algebra program for this calculation, e.g. `FORM` [17].

Bibliography

[1] R. P. Feynman, "Space-time approach to quantum electrodynamics," *Physical Review* **76** (1949), no. 6, 769.

[2] D. Kreimer and K. Yeats, "An Etude in non-linear Dyson-Schwinger Equations," *Nucl.Phys.Proc.Suppl.* **160** (2006) 116–121, hep-th/0605096.

[3] L. Klaczynski, "Lecture notes to renormalization and renormalization group." The script can be found at http://www.mathematik.hu-berlin.de/~maphy/introRGE.html.

[4] A. Connes and D. Kreimer, "Renormalization in quantum field theory and the Riemann-Hilbert problem. 1. The Hopf algebra structure of graphs and the main theorem," *Commun.Math.Phys.* **210** (2000) 249–273, hep-th/9912092.

[5] L. Foissy, "General Dyson-Schwinger equations and systems," *ArXiv e-prints* (Dec., 2011) 1112.2606.

[6] M. Maggiore, *A Modern introduction to quantum field theory*. Oxford University Press, 2005.

[7] S. Weinberg, *The Quantum Theory of Fields, Volume 1: Foundations*. Cambridge University Press, 2005.

[8] E. Panzer, "Hopf-algebraic renormalization of Kreimer's toy model," Master's thesis, Humboldt Universität zu Berlin, 2012.

[9] L. Foissy, "General Dyson-Schwinger equations and systems," *ArXiv e-prints* (Dec., 2011) 1112.2606.

[10] G. van Baalen, D. Kreimer, D. Uminsky, and K. Yeats, "The QED beta-function from global solutions to Dyson-Schwinger equations," *Annals Phys.* **324** (2009) 205–219, 0805.0826.

[11] C. Bergbauer and D. Kreimer, "Hopf algebras in renormalization theory: Locality and Dyson-Schwinger equations from Hochschild cohomology," *IRMA Lect.Math.Theor.Phys.* **10** (2006) 133–164, hep-th/0506190.

[12] C. Reutenauer, *Free lie algebras*. London Mathematical Society monographs. Clarendon Press New York, Oxford, 1993.

[13] A. Connes and D. Kreimer, "Hopf algebras, renormalization and noncommutative geometry," *Commun.Math.Phys.* **199** (1998) 203–242, hep-th/9808042.

[14] M. Hall, Jr., "A basis for free Lie rings and higher commutators in free groups," *Proc. Amer. Math. Soc.* **1** (1950).

[15] J. W. Milnor and J. C. Moore, "On the structure of Hopf algebras," *Annals of Mathematics* **81** (1965), no. 2, pp. 211–264.

[16] D. Kreimer, "Quantum fields, periods and algebraic geometry," *Proceedings of the converence PM2012 - Periods and Motives (Madrid, July 2-6, 2012)* (Apr., 2013). to appear. http://www.mathematik.hu-berlin.de/~maphy/MadridAMS.pdf.

[17] J. Vermaseren, "FORM." Version 4.0 is freely available at http://www.nikhef.nl/~form/maindir/maindir.html.

Acknowledgments

First of all, I want to thank my adviser Professor Dirk Kreimer. He motivated my thesis and gave a lot of ideas. In particular, he always took the time to answer my questions in detail. Last but not least, I thank him for the lectures to 'renormalization and renormalization group' and 'Dyson-Schwinger equations'. I am glad that I had the chance to learn so much about Hopf algebras and the mathematical background to renormalization in physics. Second, I want to thank Eric Panzer for so much mathematical help. He took a lot of time to explain things in the very detail and I found a lot of answers in the discussions we had. I acknowledge useful discussions with Lutz Klaczynski, Michael Borinsky, Hagen Münkler, Bettina Grauel and Martin Lüders and I also thank the whole research group for the great working atmosphere. Finally, I thank my girlfriend for the correction of many English mistakes and my whole family for the mental and financial support, especially during the last five years of my studies.

Statutory declaration

Herewith I affirm that the presented thesis was made autonomous making use only of the specified literature and aids.

Berlin, 28.10.2013 Olaf Krüger