Graph Grammars

Matilde Marcolli
MAT1509HS: Mathematical and Computational Linguistics

University of Toronto, Winter 2019, T 4-6 and W 4, BA6180

Main Reference:

- Matilde Marcolli, Alexander Port, *Graph grammars, insertion Lie algebras, and quantum field theory*, Math. Comput. Sci. 9 (2015), no. 4, 391–408.

Graph Grammars and Quantum Field Theory

• Example of a different setting where formal languages can be applied, with a different class of formal grammars (graph grammars)

Graph Grammars

Formal languages adapted to parallelism in computation

- instead of linear languages: strings in an alphabet obtained by production rules of a grammar
- grammars that produce a language consisting of a family of graphs
- production rules that substitute parts of a graph with other parts (gluing)
- an initial graph as starting point
- edge and vertex labels by terminal and non-terminal symbols

Graphs

Two main ways of thinking about graphs:

First description:

- $V(G)$ = set of vertices; $E(G)$ = set of edges; $\partial : E(G) \to V(G) \times V(G)$

- if G is oriented (directed) then source and target $s, t : E(G) \to V(G)$

- Σ_V, Σ_E sets of vertex and edge labels; $L_{V,G} : V(G) \to \Sigma_V$, $L_{E,G} : E(G) \to \Sigma_E$ assignment of labels

Second description:

- $C(G)$ = set of corollas with assigned valences
(a vertex with n half-edges)

- $\mathcal{F}(G)$ = set of all half-edges

- involution: $\mathcal{I} : \mathcal{F}(G) \to \mathcal{F}(G)$

- edges: pairs (f, f') with $f \neq f'$ in $\mathcal{F}(G)$ with $\mathcal{I}(f) = f'$
(an edge is a gluing of two half edges)

- external edges: $f \in \mathcal{F}(G)$ fixed by the involution $\mathcal{I}$
(half-edges not matched to anything else)

- assignment of labels $L_{\mathcal{F},G} : \mathcal{F}(G) \to \Sigma_{\mathcal{F}}$ and $L_{V,G} : C(G) \to \Sigma_V$

$$L_{\mathcal{F},G} \circ \mathcal{I} = L_{\mathcal{F},G}$$

(the involution must match labels)

Graph Grammar

$$(N_E, N_V, T_E, T_V, P, G_S)$$

- edge labels: $\Sigma_E = N_E \cup T_E$ non-terminal and terminal
- vertex labels: $\Sigma_V = N_V \cup T_V$ non-terminal and terminal
- $G_S =$ start graph
- $P =$ production rules: a finite set

Production rules of a Graph Grammar

$$P = (G_L, G_R, H)$$

- G_L = labelled graph (l.h.s. of production)
- G_R = labelled graph (r.h.s. of production)
- H = labelled graph with label preserving isomorphisms

$$\phi_L : H \stackrel{\simeq}{\to} \phi_L(H) \subset G_L, \quad \phi_R : H \stackrel{\simeq}{\to} \phi_R(H) \subset G_R$$

(isomorphic subgraphs in G_L and G_R)

Meaning: the production rule searches for a copy of G_L inside a given graph G and glues in a copy of G_R by identifying them along the common subgraph H

Context-free Graph Grammars

• when all production rules $P = (G_L, G_R, H)$ have G_L (hence H) a single vertex

• Chomsky hierarchy for Graph Grammar (different from the one for linear languages) was identified in

- M. Nagl, *Graph-Grammatiken: Theorie, Implementirung, Anwendung*, Vieweg, 1979

References on Graph Grammars

- H. Ehrig, K. Ehrig, U. Prange, G. Taentzer, *Fundamentals of algebraic graph transformation*. New York: Springer, 2010.
- H. Ehrig, H.J. Kreowski, G. Rozenberg, *Graph-grammars and their application to computer science*, Lecture Notes in Computer Science, Vol. 532, Springer, 1990.
- G. Rozenberg, *Handbook of Graph Grammars and Computing by Graph Transformation. Volume 1: Foundations*, World Scientific, 1997.

Perturbative Quantum Field Theory

- perturbative (massless, scalar) field theory: classical action

$$S(\phi) = \int \mathcal{L}(\phi) d^D x = S_0(\phi) + S_{int}(\phi)$$

in D dimensions

- Lagrangian density

$$\mathcal{L}(\phi) = \frac{1}{2}(\partial\phi)^2 - \frac{m^2}{2}\phi^2 - \mathcal{L}_{int}(\phi)$$

- Quantum effects: perturbative expansion in Feynman diagrams

$$S_{eff}(\phi) = S_0(\phi) + \sum_G \frac{G(\phi)}{\#\mathrm{Aut}(G)} \quad \text{(1PI graphs)}$$

contribution of each Feynman graph is a finite dimensional integral in momentum variables flowing through the graph

Renormalization problem in QFT

• most of the integrals $G(\phi)$ in the expansion are divergent

• need a regularization procedure (pole subtraction, cutoff, ...)

• and renormalization is implementation consistently over subgraphs (nested subdivergences)

• Connes-Kreimer ($\sim$ 2000): the renormalization procedure is described algebraically by a Hopf algebra of Feynman graphs

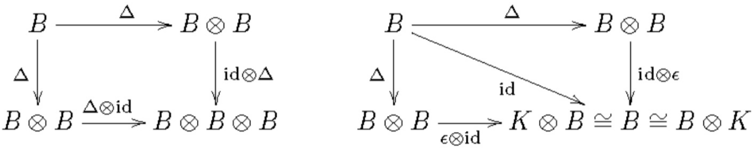
Hopf algebras

- **Bialgebra**
  - $\mathcal{B}$ algebra over a field $\mathbb{K}$ with $\mathbb{K}$-linear multiplication operation $\nabla : \mathcal{B} \otimes \mathcal{B} \to \mathcal{B}$ associative (not necessarily commutative) and unit $\eta : \mathbb{K} \to \mathcal{B}$
  - $\mathbb{K}$-linear comultiplication $\Delta : \mathcal{B} \to \mathcal{B} \otimes \mathcal{B}$ coassociative

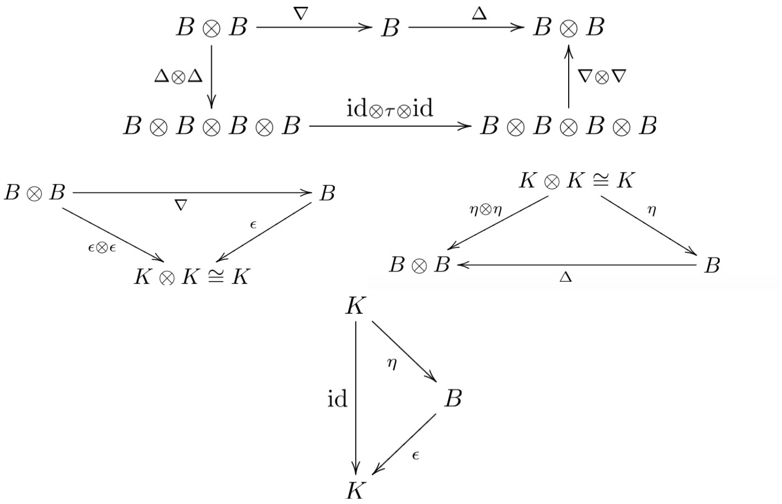
$$(id_{\mathcal{B}} \otimes \Delta) \circ \Delta = (\Delta \circ id_{\mathcal{B}}) \circ \Delta$$

(not necessarily cocommutative) and counit $\epsilon : \mathcal{H} \to \mathbb{K}$

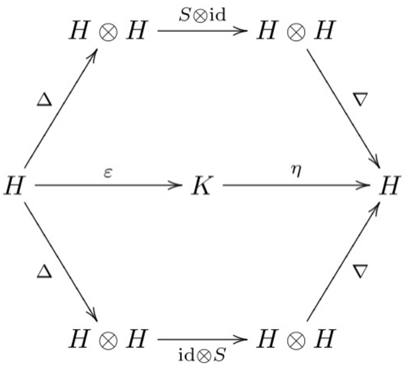
$$(id_{\mathcal{B}} \otimes \epsilon) \circ \Delta = id_{\mathcal{B}} = (\epsilon \otimes id_{\mathcal{B}}) \circ \Delta$$



- compatibility of ∇, η, Δ, ϵ for bialgebras



- Hopf algebra $\mathcal{H}$
  - $\mathcal{H}$ is a bialgebra over a field $\mathbb{K}$
  - $\mathbb{K}$-linear antipode map $S : \mathcal{H} \to \mathcal{H}$ with compatibility



Hopf algebras and Affine Group Schemes

• a *commutative* Hopf algebra $\mathcal{H}$ is equivalent to an affine group scheme G

• affine scheme: functor $X : \mathrm{Alg}_{\mathbb{K}} \to \mathrm{Sets}$ from category of commutative algebras over $\mathbb{K}$ to sets

$$X(A) = \mathrm{Hom}_{\mathrm{Alg}_{\mathbb{K}}}(R, A)$$

some commutative algebra R

• affine group scheme: functor $G : \mathrm{Alg}_{\mathbb{K}} \to \mathrm{Grps}$ from category of commutative algebras over $\mathbb{K}$ to category of groups

$$G(A) = \mathrm{Hom}_{\mathrm{Alg}_{\mathbb{K}}}(\mathcal{H}, A)$$

- product from coproduct $\phi_1 \star \phi_2(x) = \phi_1 \otimes \phi_2(\Delta(x))$
- inverse from antipode $\phi^{-1}(x) = \phi(S(x))$
- unit from counit $\epsilon(x)$

Lie algebras and pre-Lie structures

• Lie algebra: vector space V with bilinear bracket $[\cdot,\cdot]$ operation with $[x,y]=-[y,x]$ and Jacobi identity

$$[x,[y,z]]+[z,[x,y]]+[y,[z,x]]=0.$$

• tangent space at the identity of a Lie group is a Lie algebra

• pre-Lie structure: a bilinear map $\star : V\otimes V\to V$ on a vector space V

$$(x\star y)\star z - x\star(y\star z) = (x\star z)\star y - x\star(z\star y)$$

identity of associators under $y\leftrightarrow z$

• Given a pre-Lie structure

$$[x,y]=x\star y - y\star x$$

is a Lie bracket (pre-Lie identity $\Rightarrow$ Jacobi identity)

Lie Algebra of an Affine Group Scheme

• functor $\mathfrak{g} : \mathrm{Alg}_{\mathbb{K}} \to \mathrm{Lie}$ from category of commutative algebras over $\mathbb{K}$ to category of Lie algebras

- $\mathfrak{g}(A)$ linear maps $L : \mathcal{H} \to A$ such that

$$L(xy) = L(x)\epsilon(y) + \epsilon(x)L(y), \quad \forall x, y \in \mathcal{H}$$

- Lie bracket

$$[L_1, L_2](x) = (L_1 \otimes L_2 - L_2 \otimes L_1)(\Delta(x))$$

• Milnor–Moore theorem: for a commutative graded connected ($\mathcal{H}_0 = \mathbb{K}$) Hopf algebra the affine group scheme G dual to $\mathcal{H}$ is completely determined by its Lie algebra $\mathfrak{g}$

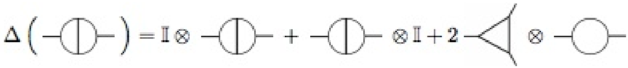
Hopf algebra of Feynman graphs

• commutative algebra generated by all the 1PI graphs G of the QFT (polynomial algebra in the G)

• comultiplication $\Delta : \mathcal{H} \to \mathcal{H} \otimes \mathcal{H}$ (coassociative, non-cocommutative)

$$\Delta(G) = G \otimes 1 + 1 \otimes G + \sum_{\gamma \subset G} \gamma \otimes G/\gamma$$

• Example:



• antipode (related algebra and coalgebra structure) constructed inductively on number of edges (or loops)

Hopf algebra and Lie algebra

• $\mathcal{H} = \oplus_{n\geq 0}\mathcal{H}_n$ with $\mathcal{H}_0 = \mathbb{C}$ connected commutative graded Hopf algebra

• $A =$ commutative algebra, $\mathrm{Hom}(\mathcal{H}, A) = \mathcal{G}(A)$ is a group

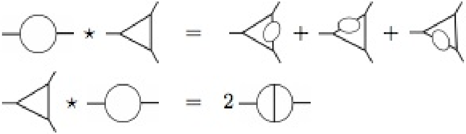
• the Hopf algebra $\mathcal{H}$ is determined by the Lie algebra $\mathcal{L}$ of $\mathcal{G}(\mathbb{C})$

• insertion of graphs is a pre-Lie operator $\Rightarrow$ Lie algebra

• insertion Lie algebra of Feynman graphs

• given two graphs G_1, G_2: count in how many ways can insert one into the other at a vertex (so that external edges glued to corolla of edges at the vertex)

- Examples of graph insertions:



gives pre-Lie structure

Lie algebra of Feynman graphs

- Lie bracket

$$[G, G'] = \sum_{v \in V(G)} G \circ_v G' - \sum_{v' \in V(G')} G' \circ_{v'} G,$$

sum over vertices and counting all possible ways of inserting the other graph at that vertex matching external edges

References on Hopf and Lie algebras in QFT

- A. Connes, D. Kreimer, *Renormalization in quantum field theory and the Riemann-Hilbert problem. I.*, Communications in Mathematical Physics 210 (2000), no. 1, 249–273.
- A. Connes, D. Kreimer, *Insertion and elimination: the doubly infinite Lie algebra of Feynman graphs*. Ann. Henri Poincaré 3 (2002), no. 3, 411–433.
- K. Ebrahimi-Fard, J.M. Gracia-Bondia, F. Patras. *A Lie theoretic approach to renormalization.* Commun. Math. Phys 276, no. 2 (2007): 519-549.
- M. Bachmann, H. Kleinert, A. Pelster, *Recursive graphical construction for Feynman diagrams of quantum electrodynamics*, Phys.Rev. D61 (2000) 085017.
- H. Kleinert, A. Pelster, B. Kastening, M. Bachmann, *Recursive graphical construction of Feynman diagrams and their multiplicities in ϕ^4 and in $\phi^2 A$ theory*, Phys.Rev. E62 (2000) 1537–1559.

Graph Grammars

This story can be generalized using Graph Grammars in two ways

1. Any context free graph grammar determines an insertion Lie algebra and a commutative Hopf algebra
2. Feynman graphs of a QFT are a graph language

From context free Graph Grammars to Insertion Lie Algebras

- Insertion Graph Grammar consists of data

$$(N_E, N_V, T_E, T_V, P, G_S)$$

edge labels $\Sigma_E = N_E \cup T_E$, nonterminal and tarminal, vertex labels is given $\Sigma_V = N_V \cup T_V$, start graph is G_S and production rules $P = (G_L, H, G_R)$, with G_L and G_R labelled graphs and H a labelled graph with isomorphisms

$$\phi_L : H \stackrel{\simeq}{\to} \phi_L(H) \subset G_L, \quad \phi_R : H \stackrel{\simeq}{\to} \phi_R(H) \subset G_R.$$

ϕ_L label preserving

- production $P = (G_L, H, G_R)$ searches for a copy of G_L inside a graph G and glues in a copy of G_R identifying them along common subgraph H, new labels matching those of $\phi_R(H)$
- context free if $G_L = \{v\}$ (hence $H = \{v\}$ also)

• formulation in terms of graphs as corollas and matched half-edges

• production rules $P = (G_L, H, G_R)$ as before with additional requirement that $\phi_L(E_{ext}(H, G_R)) \subset E_{ext}(G_L, G)$ and $\phi_R(E_{ext}(H, G_L)) \subset E_{ext}(G_R)$, for any G the production rule is applied to, $G_L \subset G$

• here gluing two graphs $G_L \cup_H G_L$ along common subgraph H by corollas

$$C_{G_L \cup_H G_L} = C_{G_L} \cup_{C_H} C_{G_R},$$

identifying corollas around each vertex of H in G_L and G_R and matching half-edges by involution

• here context-free: $G_L = H = C(v)$ corolla of a vertex v, and all vertices of graphs in the graph language have same valence

Insertion Operator and Lie Algebra

- given a context-free insertion graph grammar $\mathcal{G}$
- $\mathcal{V}$ vector space spanned by set $\mathcal{W}_\mathcal{G}$ of all the graphs obtained by repeated application of production rules starting with G_S (not same as graph language $\mathcal{L}_\mathcal{G}$ because also nonterminal labels)
- insertion operator $\triangleleft : \mathcal{V} \otimes \mathcal{V} \to \mathcal{V}$

$$G_1 \triangleleft G_2 = \sum_{v \in V(G_1)} P(v, v_2, G_2)(G_1) = \sum_{v \in V(G_1)} G_1 \triangleleft_v G_2$$

defines a pre-Lie structure on $\mathcal{V}$

- Lie algebra $\mathrm{Lie}_\mathcal{G}$ vector space $\mathcal{V}$ spanned graphs of $\mathcal{W}_\mathcal{G}$ with Lie bracket $[G_1, G_2] = G_1 \triangleleft G_2 - G_2 \triangleleft G_1$
- there is also a version using corollas, and there are versions for context-sensitive cases

The Graph Language of a Quantum Field Theory

• Note: this is not the same construction as the Lie algebra of the Connes–Kreimer Hopf algebra (because that would require an infinite number of production rules: generated by all primitive elements of the Hopf algebra

• This method gives a different insertion Lie algebra with a similar structure based on just finitely many production rules that realize all Feynman graphs of a given QFT as the graph language of a graph grammar

Example: Feynman graph language of ϕ^4-theory

• quantum field theory with Lagrangian

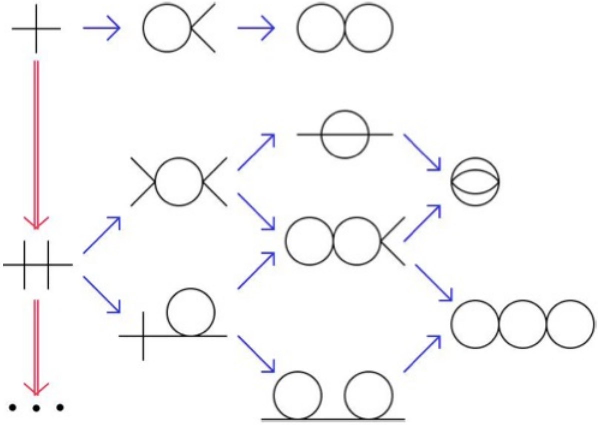
$$\mathcal{L}(\phi) = \frac{1}{2}(\partial\phi)^2 + \frac{1}{2}m^2\,\phi^2 + \frac{1}{4!}\lambda\,\phi^4$$

Feynman graphs have all vertices of valence four

• graph language $\mathcal{L}_{\mathcal{G}}$ generated by a graph grammar $\mathcal{G}$ with G_S a 4-valent corolla and two production rules:

1. $P(G_S, \{f, f'\} \subset \mathcal{F}_{G_S}, G_e)$ glues two external edges of G_S
2. $P(G_S, \{f\} \subset \mathcal{F}_{G_S}, G_S \cup_{f'} G_e)$ glues two copies of G_S along an edge

Example: Feynman graph language of ϕ^4-theory



Example: other scalar field theory examples ϕ^3 and ϕ^4

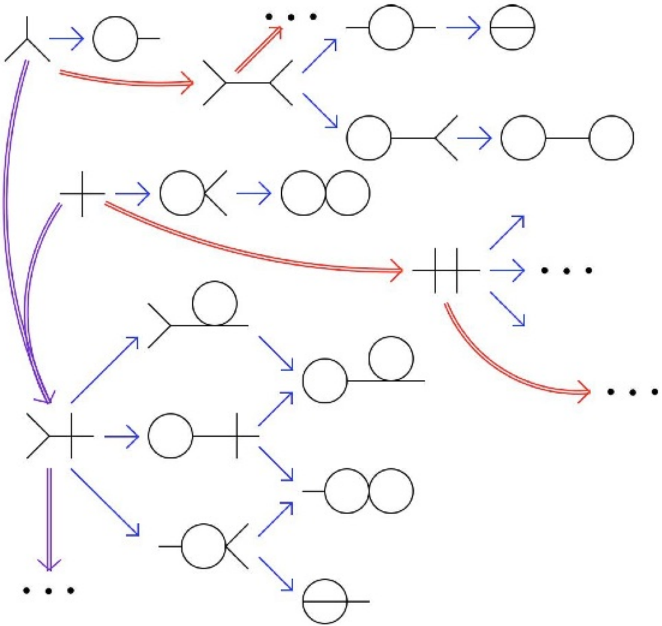
- scalar field theory with Lagrangian

$$\mathcal{L}(\phi) = \frac{1}{2}(\partial\phi)^2 + \frac{1}{2}m^2\,\phi^2 + \frac{1}{6}\lambda_3\,\phi^3 + \frac{1}{24}\lambda_4\,\phi^4$$

- start graph G_S given by a k-valent corolla, for smallest k in interaction Lagrangian (here $k=3$) and production rules

1. $P(G_S, \{f, f'\} \subset \mathcal{F}_{G_S}, G_e)$ gluing two external edges of G_S
2. $P(G_S, G_e, G_{S,f})$ gluing a copy of G_e (edge propagator) to start graph G_S one half-edge of G_e with one half-edge of G_S other half edge f as new external edge
3. $P(G_{S,f_1,\ldots,f_r}, \{f_i\} \subset \mathcal{F}_{G_S}, G_{S,f_1,\ldots,f_r} \cup_{f_i=f'_j} G_{S,f'_1,\ldots,f'_s})$ gluing along an edge two corollas $G_{S,f_1,\ldots,f_r}$ and $G_{S,f'_1,\ldots,f'_s}$

Example: ϕ^3 and ϕ^4 terms



Example: Feynman graph language of $\phi^2 A$-theory

• similar to a ϕ^3 theory but with two fields A and ϕ (similar to electrodynamics) with a cubic interaction term $\phi^2 A$

• all graphs have trivalent vertices: corolla with one A-labelled half-edge and two ϕ-labelled half-edges

• graph grammar with initial graph that is more complicated than a corolla: two vertices connected by one A-edge and each with two ϕ half-edges and two production rules (gluing two ϕ half-edges; gluing two copies of initial graph along a ϕ-edge)

Example: Feynman graph language of $\phi^2 A$-theory