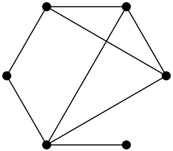
Classifying Graph Lie Algebras

Michael Gintz
Mentor: Dr. Tanya Khovanova

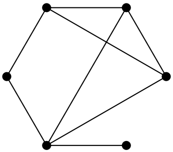
PRIMES Conference, May 20 2017

Defining a Graph



A *graph* is a set of points, some of which are connected by lines.

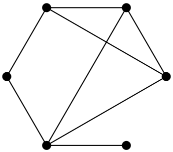
Defining a Graph



A *graph* is a set of points, some of which are connected by lines.

A *simple graph* has at most one connection between any two points, and no connection between a point and itself.

Defining a Graph



A *graph* is a set of points, some of which are connected by lines.
A *simple graph* has at most one connection between any two points, and no connection between a point and itself.
There are many well-known graph theory problems:

- The Konigsberg Bridges Problem
- The Traveling Salesman
- The Four-Color Theorem

Defining a Graph Algebra

An *algebra* is a vector space equipped with a multiplication operator.

Defining a Graph Algebra

An *algebra* is a vector space equipped with a multiplication operator. Consider a labeled graph G with n vertices $v_1, ..., v_n$. We create a *graph algebra* $\mathcal{A}(G)$ over $\mathbb{C}$ with n generators $e_1, ..., e_n$.

Defining a Graph Algebra

An *algebra* is a vector space equipped with a multiplication operator. Consider a labeled graph G with n vertices $v_1, \ldots, v_n$. We create a *graph algebra* $\mathcal{A}(G)$ over $\mathbb{C}$ with n generators $e_1, \ldots, e_n$.

- Our algebra includes 1.

Defining a Graph Algebra

An *algebra* is a vector space equipped with a multiplication operator. Consider a labeled graph G with n vertices $v_1, \ldots, v_n$. We create a *graph algebra* $\mathcal{A}(G)$ over $\mathbb{C}$ with n generators $e_1, \ldots, e_n$.

- Our algebra includes 1.
- Our set of monomials is *linearly independent*: no monomial is a sum of multiples of other monomials.

Defining a Graph Algebra

An *algebra* is a vector space equipped with a multiplication operator. Consider a labeled graph G with n vertices $v_1, \ldots, v_n$. We create a *graph algebra* $\mathcal{A}(G)$ over $\mathbb{C}$ with n generators $e_1, \ldots, e_n$.

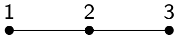
- Our algebra includes 1.
- Our set of monomials is *linearly independent*: no monomial is a sum of multiples of other monomials.
- $e_i^2 = -1$.

Defining a Graph Algebra

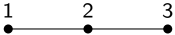
An *algebra* is a vector space equipped with a multiplication operator. Consider a labeled graph G with n vertices $v_1, \ldots, v_n$. We create a *graph algebra* $\mathcal{A}(G)$ over $\mathbb{C}$ with n generators $e_1, \ldots, e_n$.

- Our algebra includes 1.
- Our set of monomials is *linearly independent*: no monomial is a sum of multiples of other monomials.
- $e_i^2 = -1$.
- e_i and e_j anticommute ($e_i e_j = -e_j e_i$) when vertices i and j are connected: otherwise they commute ($e_i e_j = e_j e_i$).

Manipulating Monomials



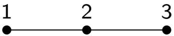
Manipulating Monomials



Definition

For a set $\alpha = \{e_{i_1}, ..., e_{i_k}\}$, the monomial e_α is equal to $e_{i_1}...e_{i_k}$.

Manipulating Monomials

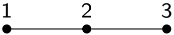


Definition

For a set $\alpha = \{e_{i_1}, \ldots, e_{i_k}\}$, the monomial e_α is equal to $e_{i_1} \ldots e_{i_k}$.

$$e_1 e_3 \cdot e_2 e_3$$

Manipulating Monomials

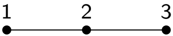


Definition

For a set $\alpha = \{e_{i_1}, ..., e_{i_k}\}$, the monomial e_α is equal to $e_{i_1} ... e_{i_k}$.

$$e_1 e_3 \cdot e_2 e_3 = -e_1 e_2 e_3 e_3$$

Manipulating Monomials

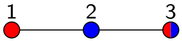


Definition

For a set $\alpha = \{e_{i_1}, \ldots, e_{i_k}\}$, the monomial e_α is equal to $e_{i_1} \ldots e_{i_k}$.

$$\begin{aligned} e_1 e_3 \cdot e_2 e_3 &= -e_1 e_2 e_3 e_3 \\ &= e_1 e_2. \end{aligned}$$

Manipulating Monomials



Definition

For a set $\alpha = \{e_{i_1}, ..., e_{i_k}\}$, the monomial e_α is equal to $e_{i_1} ... e_{i_k}$.

$$
\begin{aligned}
e_1 e_3 \cdot e_2 e_3 &= -e_1 e_2 e_3 e_3 \\
&= e_1 e_2.
\end{aligned}
$$

Theorem

Monomials e_α and e_β anticommute if there exist an odd number of pairs of connected vertices with one in α and one in β.

Structure of a Graph Algebra

Definition

The *symmetric difference* of two sets α, β is defined as follows:

$$\alpha \triangle \beta = (\alpha \cup \beta) \backslash (\alpha \cap \beta).$$

Structure of a Graph Algebra

Definition

The *symmetric difference* of two sets α, β is defined as follows:

$$\alpha \triangle \beta = (\alpha \cup \beta) \backslash (\alpha \cap \beta).$$

Multiplication is easily calculated up to a sign:

Lemma

For two sets α, β, the product $e_\alpha e_\beta$ is equal to $\pm e_{\alpha \triangle \beta}$.

Structure of a Graph Algebra

Definition

The *symmetric difference* of two sets α, β is defined as follows:

$$\alpha \triangle \beta = (\alpha \cup \beta) \backslash (\alpha \cap \beta).$$

Multiplication is easily calculated up to a sign:

Lemma

For two sets α, β, the product $e_\alpha e_\beta$ is equal to $\pm e_{\alpha \triangle \beta}$.

Mutiplying $e_1 e_3 \cdot e_2 e_3$ always yields $\pm e_1 e_2$:

$$\begin{array}{ll} e_1 & \cancel{e_3} \\ \quad e_2 & \cancel{e_3} = \pm e_1 e_2. \end{array}$$

Defining a Lie Algebra

A Lie algebra is a vector space equipped with a Lie bracket.

Defining a Lie Algebra

A Lie algebra is a vector space equipped with a Lie bracket.
Any Lie bracket must have certain properties.

Defining a Lie Algebra

A Lie algebra is a vector space equipped with a Lie bracket.
Any Lie bracket must have certain properties.

- It must be bilinear: $[ax + by, z] = a[x, z] + b[y, z]$ for all a, b in our field (in our case $\mathbb{C}$).

Defining a Lie Algebra

A Lie algebra is a vector space equipped with a Lie bracket.
Any Lie bracket must have certain properties.

- It must be bilinear: $[ax + by, z] = a[x, z] + b[y, z]$ for all a, b in our field (in our case $\mathbb{C}$).
- It must be alternative: $[x, x] = 0$.

Defining a Lie Algebra

A Lie algebra is a vector space equipped with a Lie bracket.
Any Lie bracket must have certain properties.

- It must be bilinear: $[ax + by, z] = a[x, z] + b[y, z]$ for all a, b in our field (in our case $\mathbb{C}$).
- It must be alternative: $[x, x] = 0$.
- It must satisfy the Jacobi Identity:

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

Graph Lie Algebras

We wish to build a Lie algebra structure from the algebra we already created.

Graph Lie Algebras

We wish to build a Lie algebra structure from the algebra we already created.
Since we have defined a multiplication function on the vector space of monomials, we can define our Lie bracket as the commutator of its operands:

$$[x, y] = xy - yx.$$

Graph Lie Algebras

We wish to build a Lie algebra structure from the algebra we already created.
Since we have defined a multiplication function on the vector space of monomials, we can define our Lie bracket as the commutator of its operands:

$$[x, y] = xy - yx.$$

We can define a Lie algebra $\mathfrak{L}(G)$ of an n-vertex graph G.

Graph Lie Algebras

We wish to build a Lie algebra structure from the algebra we already created.
Since we have defined a multiplication function on the vector space of monomials, we can define our Lie bracket as the commutator of its operands:

$$[x, y] = xy - yx.$$

We can define a Lie algebra $\mathfrak{L}(G)$ of an n-vertex graph G.

- Every generator $e_1, ..., e_n$ is in $\mathfrak{L}(G)$.

Graph Lie Algebras

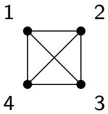
We wish to build a Lie algebra structure from the algebra we already created.
Since we have defined a multiplication function on the vector space of monomials, we can define our Lie bracket as the commutator of its operands:

$$[x, y] = xy - yx.$$

We can define a Lie algebra $\mathfrak{L}(G)$ of an n-vertex graph G.

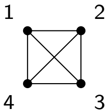
- Every generator $e_1, ..., e_n$ is in $\mathfrak{L}(G)$.
- No Lie subalgebra of $\mathfrak{L}(G)$ contains every generator.

Forming a Lie Algebra



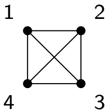
We can form the Lie algebra of the complete graph K_4 by hand.

Forming a Lie Algebra



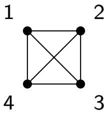
We can form the Lie algebra of the complete graph K_4 by hand.
We must have monomials e_1, e_2, e_3, e_4.

Forming a Lie Algebra



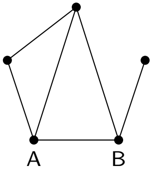
We can form the Lie algebra of the complete graph K_4 by hand.
We must have monomials e_1, e_2, e_3, e_4.
We can combine each pair of generators using the Lie bracket:
$e_1 e_2, e_1 e_3, e_1 e_4, e_2 e_3, e_2 e_4, e_3 e_4$.

Forming a Lie Algebra



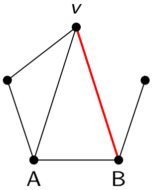
We can form the Lie algebra of the complete graph K_4 by hand.
We must have monomials e_1, e_2, e_3, e_4.
We can combine each pair of generators using the Lie bracket:
$e_1 e_2, e_1 e_3, e_1 e_4, e_2 e_3, e_2 e_4, e_3 e_4$.
We can not have any more monomials, as every pair of monomials other than pairs of generators contain commuting monomials. Therefore our Lie Algebra has 10 dimensions.

Swapping our Graph



We will create a series of alterations on our graph called a *swap* about vertex A with respect to B. This is denoted as ${}_AG_B$. We consider all vertices $v \neq A, B$ connected to A.

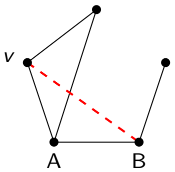
Swapping our Graph



We will create a series of alterations on our graph called a *swap* about vertex A with respect to B. This is denoted as ${}_AG_B$. We consider all vertices $v \neq A, B$ connected to A.

- If v is connected to B, we remove the edge connecting v and B.

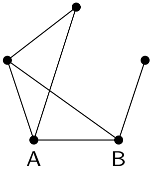
Swapping our Graph



We will create a series of alterations on our graph called a *swap* about vertex A with respect to B. This is denoted as ${}_AG_B$. We consider all vertices $v \neq A, B$ connected to A.

- If v is connected to B, we remove the edge connecting v and B.
- If v is not connected to B, we add an edge connecting v and B.

Swapping our Graph



We will create a series of alterations on our graph called a *swap* about vertex A with respect to B. This is denoted as ${}_AG_B$. We consider all vertices $v \neq A, B$ connected to A.

- If v is connected to B, we remove the edge connecting v and B.
- If v is not connected to B, we add an edge connecting v and B.

Theorems Concerning Swaps

Swapping our graph always preserves its algebra:

Theorem

For all graphs G, algebras $\mathcal{A}(G)$ and $\mathcal{A}({}_AG_B)$ are isomorphic.

Theorems Concerning Swaps

Swapping our graph always preserves its algebra:

Theorem

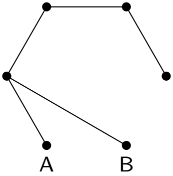
For all graphs G, algebras $\mathcal{A}(G)$ and $\mathcal{A}({}_AG_B)$ are isomorphic.

Sometimes, swapping our graph can preserve the Lie algebra as well:

Theorem

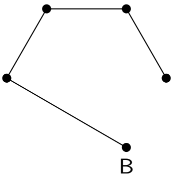
For all graphs G, Lie algebras $\mathfrak{L}(G)$ and $\mathfrak{L}({}_AG_B)$ are isomorphic when A and B are connected.

Removing Leaves from our Graph



Say we have a graph G with 2 leaves A, B connected to the same vertex.

Removing Leaves from our Graph

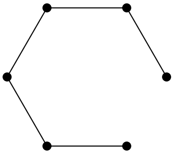


Say we have a graph G with 2 leaves A, B connected to the same vertex.

Theorem

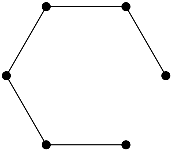
$$\mathfrak{L}(G) = \mathfrak{L}(G \backslash A) \oplus \mathfrak{L}(G \backslash A).$$

Completely Classified Graphs

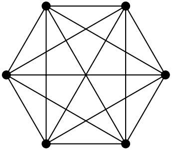


A path graph with n vertices has a Lie algebra isomorphic to a skew symmetric matrix Lie algebra with size $n + 1$.

Completely Classified Graphs

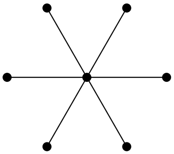


A path graph with n vertices has a Lie algebra isomorphic to a skew symmetric matrix Lie algebra with size $n + 1$.



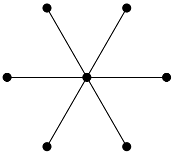
The Lie algebra of a complete graph with n vertices is isomorphic to that of a path graph with n vertices.

More Completely Classified Graphs

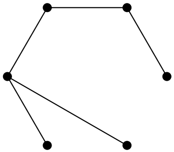


The Lie algebra of a star graph with n vertices is the direct sum of 2^{n-2} copies of a connected 2-vertex graph Lie algebra.

More Completely Classified Graphs



The Lie algebra of a star graph with n vertices is the direct sum of 2^{n-2} copies of a connected 2-vertex graph Lie algebra.



The Lie algebra of a $n-2$ vertex graph with 2 leaves attached to the same end (Dynkin diagram D_n) is a direct sum of two copies of the Lie algebra of an $n-1$ vertex path graph.

Future

In the future, we wish to classify all possible graph Lie algebras.

Future

In the future, we wish to classify all possible graph Lie algebras.

- Generalize our decomposition move.

Future

In the future, we wish to classify all possible graph Lie algebras.

- Generalize our decomposition move.
- Create similar alterations to swaps.

Future

In the future, we wish to classify all possible graph Lie algebras.

- Generalize our decomposition move.
- Create similar alterations to swaps.
- Relate graph Lie algebras to matrix algebras.

Acknowledgements

- Dr. Tanya Khovanova
- The PRIMES Program
- My parents

Questions?