

Graph operations and Lie algebras

J. Cáceres¹, M. Ceballos², J. Núñez², M.L. Puertas¹, Á.F.Tenorio³

¹Dpto. de Estadística y Matemática Aplicada.

Universidad de Almería.

Ctra. Sacramento s/n, La Cañada de San Urbano. 04120-Almería (Spain).

²Departamento de Geometría y Topología.

Facultad de Matemáticas. Universidad de Sevilla.

Aptdo. 1160. 41080-Seville (Spain).

³Dpto. de Economía, Métodos Cuantitativos e Historia Económica.

Escuela Politécnica Superior. Universidad Pablo de Olavide.

Ctra. Utrera km. 1. 41013-Seville (Spain).

jcaceres@ual.es mceballos@us.es jnvaldes@us.es mpuertas@ual.es aftenorio@upo.es

Abstract

This paper deals with several operations on graphs and combinatorial structures linking them with their associated Lie algebras. More concretely, our main goal is to obtain some criteria to determine when there exists a Lie algebra associated with a combinatorial structure arising from those operations. Additionally, we show an algorithmic method for one of those operations.

Keywords: Digraph, Combinatorial structure, Lie algebra, Combinatorial operations, Algorithm.

2010 Mathematics Subject Classification: 17B60, 05C25, 05C20, 05C90, 68W30, 68R10, 05C85.

1 Introduction

At present, one of the most stimulating research in Mathematics consists in discovering new links and relations between different fields. From this perspective, researchers can achieve alternative techniques to solve open problems, improve known theories and reveal new ones. This paper works out with the relation between Graph and Lie theories. Regarding this working strategy, our research gives continuity to the line started in [1], where a mapping between Lie algebras and combinatorial structures was introduced in order to translate properties of Lie algebras into the language of Graph Theory and vice versa.

Research on Lie Theory is very extensive because of its own theoretical importance as well as of its application to many different fields, like Engineering, Physics and Applied Mathematics. Nevertheless, many general questions about Lie algebras remain unanswered when using traditional techniques, such as the classification of Lie algebras which is still unsolved. Since Levi's and Malcev's theorems (see [7] and [8], respectively) state that every finite-dimensional Lie algebra is decomposable into the semidirect sum of a semisimple Lie

algebra and a solvable ideal, the classification of Lie algebras can be reduced to classify the semisimple and the solvable families of Lie algebras. Whereas semisimple Lie algebras were completely classified by Killing and Cartan's classification of simple Lie algebras in 1890, solvable Lie algebras have only been classified for low dimensions. As current techniques do not allow researchers successfully to face up to this classification problem, studying new and different properties of Lie algebras is compulsory and arises naturally. To achieve this purpose, mathematicians have dealt during the last decades with different links between solvable Lie algebras and other fields.

Analogously, Graph Theory is also running in high-level research nowadays due to its use as a helpful tool to be applied to almost any type of topics and fields. Concerning Lie Theory, graphs have been essential for studying semisimple Lie algebras, because trees perform an important role to determine the Dynkin diagrams associated to such algebras [10]. Nowadays, Graph Theory is also applied to study the representation of finite-dimensional algebras [9].

Our main goal consists in making progress with the link between Lie algebras and combinatorial structures (including graphs and simplicial complexes), by proceeding in the line of [1, 2, 3, 4, 6]. Indeed, these papers are based on the definition of a mapping between Lie algebras and combinatorial structures. Furthermore, the graph-based approach in this research line is completely different from that of papers using Dynkin diagrams, which are strictly graphs and do not include simplicial complex of higher dimension. This time, we study the translation of several operations on graphs and combinatorial structures into the language of Lie algebras.

The structure of this paper is the following: after reviewing some well-known results on Lie and Graph theories in Section 2, Section 3 recalls the mapping introduced in [1] to associate combinatorial structures with Lie algebras, as well as remarks some of its properties proved in [1]. Next, Section 4 shows the behavior of several operations on graphs and combinatorial structures in relation with this mapping. In this sense, we state some criteria under which structures coming from these operations preserve the property of "being associated with Lie algebras". Finally, Section 5 presents an algorithmic method which computes the amalgamation of two digraphs and checks if that resulting graph is associated or not with a Lie algebra.

In our opinion, the tools and results dealt with in this article are very useful and helpful to advance in understanding the relation between Lie algebras and simplicial complexes, since the classification of combinatorial structures may involve an easier method to solve the classification problem of Lie algebras by means of the classification of their associated combinatorial structures.

2 Preliminaries

We show some preliminary concepts on Lie algebras, bearing in mind that the reader can consult [11] for a general overview. From here on, we only consider finite-dimensional Lie algebras over the complex number field \mathbb{C} .

Definition 1. A Lie algebra \mathfrak{g} is a vector space with a second bilinear inner composition law $([\cdot, \cdot])$ called the bracket product or Lie bracket, which satisfies

$$[X, X] = 0, \quad \forall X \in \mathfrak{g} \quad \text{and} \quad J(X, Y, Z) = 0, \quad \forall X, Y, Z \in \mathfrak{g},$$

where J is the Jacobiator defined as $J(X, Y, Z) = [[X, Y], Z] + [[Y, Z], X] + [[Z, X], Y]$. The last expression is called the Jacobi identity.

Given a basis $\{e_i\}_{i=1}^n$ of \mathfrak{g} , its structure (or Maurer-Cartan) constants are defined by $[e_i, e_j] = \sum c_{i,j}^h e_h$, for $1 \leq i < j \leq n$.

Definition 2. Given a Lie algebra \mathfrak{g} , its center is defined as $Z(\mathfrak{g}) = \{X \in \mathfrak{g} \mid [X, Y] = 0, \forall Y \in \mathfrak{g}\}$.

Although the reader can consult [5] as an introductory reference to Graph Theory, some notions are recalled next in this section.

Definition 3. A graph consists in an ordered pair $G = (V, E)$, where V is a non-empty set called the vertex set and E is a set of unordered pairs (edges) of two vertices, called the edge set. If the edges are ordered pairs of vertices, then the graph is named digraph.

Definition 4. Let $G = (V, E)$ be a graph. For a vertex $v \in V$, the (open) neighbourhood of v in G is the vertex subset $N(v) = \{w \in V \mid (v, w) \in E\}$.

Definition 5. Given a graph $G = (V, E)$, two vertices $u, v \in V$ are twin if they have the same neighbourhoods; i.e. $N(u) = N(v)$.

Definition 6. Given a digraph $G = (V, E)$, a vertex $v \in V$ is a sink (resp. a source) if all the edges incident with v are oriented towards v (resp. oriented from v). This definition is illustrated in Figure 1.



Figure 1: Example of sinks and sources, respectively.

Definition 7. Given $n \in \mathbb{N}$, P_n is a weighted digraph of n vertices alternating sources with sinks.

3 Associating combinatorial structures with Lie algebras

Let \mathfrak{g} be an n -dimensional Lie algebra with basis $\mathcal{B} = \{e_i\}_{i=1}^n$. The structure constants are given by $[e_i, e_j] = \sum_{k=1}^n c_{i,j}^k e_k$ and, hence, the pair $(\mathfrak{g}, \mathcal{B})$ is associated with a combinatorial structure built according to the following steps in the method introduced in [1]

- a) Draw vertex i for each $e_i \in \mathcal{B}$.
- b) Given three vertices $i < j < k$, draw the full triangle ijk if and only if $(c_{i,j}^k, c_{j,k}^i, c_{i,k}^j) \neq (0, 0, 0)$. Then, the edges ij , jk and ik have weights $c_{i,j}^k$, $c_{j,k}^i$ and $c_{i,k}^j$, respectively.
 - b1) Use a discontinuous line (named *ghost edge*) for edges with weight zero.
 - b2) If two triangles ijk and ijl with $1 \leq i < j < k < l \leq n$ satisfy $c_{i,j}^k = c_{i,j}^l$, draw only one edge between vertices i and j shared by both triangles (see Figure 2).

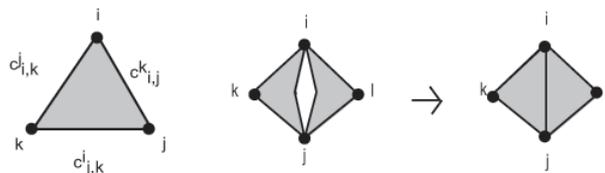


Figure 2: Full triangle and two triangles sharing an edge.

- c) Given two vertices i and j with $1 \leq i < j \leq n$ and such that $c_{i,j}^i \neq 0$ (resp. $c_{i,j}^j \neq 0$), draw a directed edge from j to i (resp. from i to j), as can be seen in Figure 3.



Figure 3: Directed edges.

Consequently, every Lie algebra with a given basis is associated with a combinatorial structure of this type, which turns out to be simplicial complexes of dimension less than 3.

Throughout the paper, we will refer to the following three results from [1], indicating both forbidden and allowed configurations in graphs associated with Lie algebras.

Lemma 1. [1, Lemma 3.1] *Let \mathfrak{g} be a Lie algebra associated with a digraph G . Then, the configurations shown in Fig. 4 are forbidden in G , for any three different vertices i, j, k (independently of the weights of the edges).*

Hence, there exist only four types of digraphs of 3 vertices associated with 3-dimensional Lie algebras (see Figure 5). Isomorphism classes of their associated Lie algebras are obtained thanks to the following two results.

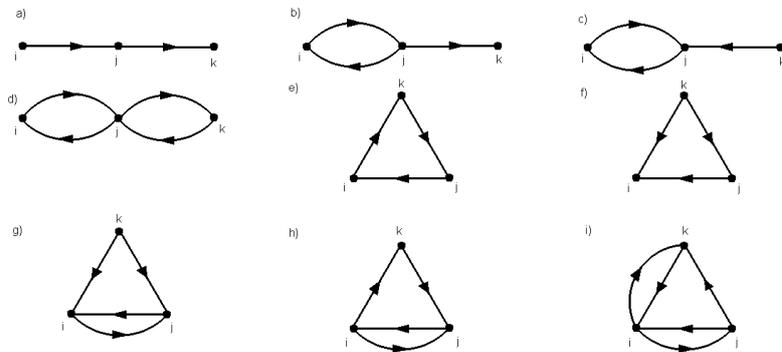


Figure 4: Forbidden configurations from [1].

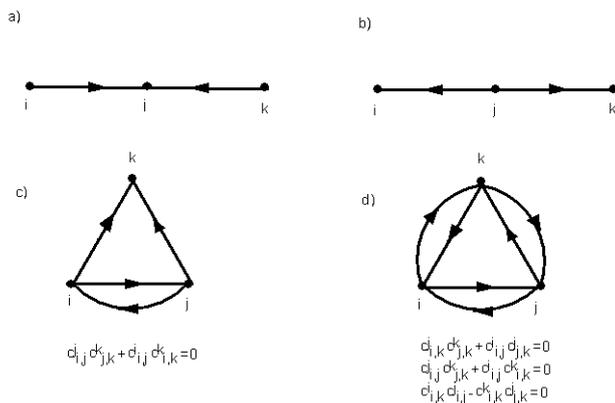


Figure 5: Digraphs of 3 vertices associated with Lie algebras.

Theorem 1. [1, Theorem 3.2] Let G be a digraph without (oriented) 3-cycles, associated with a Lie algebra. Then, G is

- (i) a unique double edge, or
- (ii) a well oriented digraph without double edges.

Conversely, any digraph satisfying (i) or (ii) is associated with a Lie algebra.

Theorem 2. [1, Theorem 3.6] Let G be a digraph containing (oriented) 3-cycles and associated with a Lie algebra. Then, G satisfies the following conditions

- (i) The double edges of G lie on the 3-cycles and there are no 3-cycles without double edges.
- (ii) The adjacent vertices with the end vertices of the double edges are not mutually adjacent. Moreover, they appear in one of the configurations of Fig. 6.
- (iii) The subdigraph obtained from G by removing its double edges satisfies condition (ii) of Theorem 1.

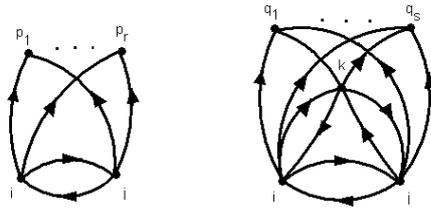


Figure 6: Configurations containing double edges from [1].

Remark 1. Note that $\{p_1, \dots, p_r\}$ and $\{q_1, \dots, q_s\}$ in Fig. 6 are both sets of twin vertices.

4 Operations on combinatorial structures associated with Lie algebras

This section studies the translation of some usual combinatorial operations on graphs and simplicial complexes into the languages of Lie algebras; namely, vertex amalgamation and edge addition, deletion and contraction.

4.1 Vertex amalgamation

The amalgamation of two combinatorial structures consists in pasting both structures by identifying a vertex in both configurations. We determine under which conditions the structure arising from this operation preserves the association with a Lie algebra.

4.1.1 Digraphs associated with Lie algebras and amalgamation

First, we start studying digraphs associated with Lie algebras, obtaining the following results.

Proposition 1. *Let G be a digraph not associated with Lie algebras. Then, every digraph obtained from G by using vertex amalgamation is neither associated with Lie algebras.*

Proof. Let G' be a digraph obtained from G by applying vertex amalgamation. Assume that the number of vertices of G and G' is respectively n and m , with $n < m$. The vector space L with basis $\{e_i\}_{i=1}^n$ associated with the graph G is not a Lie algebra. Therefore, there exist $1 \leq i < j < k \leq n$ such that $J(e_i, e_j, e_k) \neq 0$. Since $L \subset L'$, this Jacobi identity is neither satisfied for the vector space L' of the digraph G' . \square

Proposition 2. *Let G and G' be two digraphs associated with the Lie algebras L and L' respectively. We consider the amalgamation of G and G' by an isolated vertex of G' . Then, there exists a unique Lie algebra associated with the amalgamation given by the directed sum $L \oplus \bar{L}$, \bar{L} is the Lie algebra associated with the subgraph $G' - \{v\}$ of G' .*

Proof. If the vertex v of G' is isolated, then $e_v \in L \cap L'$ and $e_v \in Z(L')$. Therefore, the non-zero brackets of L' are the ones corresponding to the subalgebra \bar{L} of L' associated with $G' - \{v\}$. The Jacobi identity $J(e_v, e_i, e_k) = 0$ holds for all $e_i \in L$ and $e_k \in \bar{L}$, since the brackets between elements of L and L' are zero. \square

Proposition 3. *Let G and G' be two digraphs associated with Lie algebras. We consider the amalgamation by a non-isolated vertex. Then, the following statements hold*

- 1) *If G is an oriented 2-cycle, then no Lie algebra is associated with this amalgamation.*
- 2) *If G contains 3-cycles (structures from Theorem 2), then the amalgamation is associated with a Lie algebra if and only if the amalgamation vertex is a sink in both G and G' . Moreover, either G and G' are digraphs of the same type or G' is a digraph P_n .*
- 3) *If G and G' do not contain 3- or 2-cycles, the amalgamation is associated with a Lie algebra if and only if the amalgamation vertex is of the same type in both G and G' .*

Proof. In Case 1), G is associated with a 2-dimensional Lie algebra. If we take the amalgamation by such a vertex, we get configurations b), c) or d) in Figure 4.

For Case 2), according to the previous one, we cannot consider the amalgamation by vertices of an oriented 2-cycle. Assume that we build the amalgamation by a twin sink of G . It is easy to prove that the amalgamation is associated with a Lie algebra if the vertex is also a sink in G' .

In Case 3), the typology of the digraphs G and G' is based on Theorem 1. If we build the amalgamation by a vertex of the same type in both digraphs, then the resulting digraph

must be of type P_n and, hence, is associated with a Lie algebra. Otherwise, if we consider different types of vertex in each digraph, we obtain the forbidden configuration a) from Lemma 1. \square

4.1.2 Full triangles associated with Lie algebras and amalgamation

Next, we study the amalgamation of full triangles associated with Lie algebras. Using a reasoning analogous to Proposition 2, we can prove

Lemma 2. *Let G and T be respectively a digraph and a triangular structure, both associated with Lie algebras. Then, the amalgamation of G and T by an isolated vertex v of G is associated with the Lie algebra $L \oplus \bar{L}$, where L and \bar{L} are the Lie algebras associated with T and $G - \{v\}$, respectively.* \square

Proposition 4. *The amalgamation of a full triangle and a digraph by a non-isolated vertex k is associated with a Lie algebra if and only if k is a source and the opposite edge to k in the full triangle is ghost.*

Proof. The two possible cases are shown in Figures 7 and 8, although edges may be ghost or full a priori. First, the Lie brackets of the structure in Figure 7 are the following: $[e_i, e_j] = c_{i,j}^k e_k$, $[e_i, e_k] = c_{i,k}^j e_j$, $[e_j, e_k] = c_{j,k}^i e_i$ and $[e_k, e_l] = c_{k,l}^k e_k$, where $c_{k,l}^k \neq 0$. We compute the following Jacobi identities

$$J(e_i, e_j, e_l) = c_{i,j}^k c_{k,l}^k e_k = 0; \quad J(e_i, e_k, e_l) = -c_{k,l}^k c_{i,k}^j e_j = 0; \quad J(e_j, e_k, e_l) = -c_{k,l}^k c_{j,k}^i e_i = 0.$$

Since $c_{k,l}^k \neq 0$, it is verified that $c_{i,j}^k = c_{j,k}^i = c_{i,k}^j = 0$. This comes into contradiction with the fact that the triangle is full. Now, we prove that the edge ij in Figure 8 must be ghost. Effectively, the non-zero brackets are as follows: $[e_i, e_j] = c_{i,j}^k e_k$, $[e_i, e_k] = c_{i,k}^j e_j$, $[e_j, e_k] = c_{j,k}^i e_i$ and $[e_k, e_l] = c_{k,l}^l e_l$, where $c_{k,l}^l \neq 0$. Now, we consider this Jacobi identity $J(e_i, e_j, e_l) = c_{i,j}^k c_{k,l}^l e_l = 0$. Since $c_{k,l}^l \neq 0$, we deduce that $c_{i,j}^k = 0$. Therefore, the opposite edge to the vertex of the amalgamation is ghost. \square

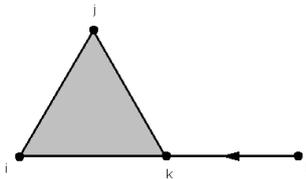


Figure 7: Amalgamation by a sink.

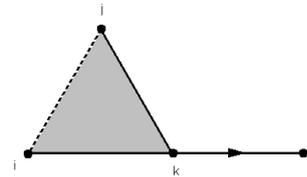


Figure 8: Amalgamation by a source.

Proposition 5. *The amalgamation of two full triangles by a vertex is associated with a Lie algebra if and only if one of the following conditions holds*

- The amalgamation vertex is only incident with ghost edges and its opposite edges are full.
- The edges not being incident with the amalgamation vertex are ghost.

Proof. We consider the amalgamation of two full triangles ijk and klm . The non-zero brackets for this structure are the following: $[e_i, e_j] = c_{i,j}^k e_k$, $[e_i, e_k] = c_{i,k}^j e_j$, $[e_j, e_k] = c_{j,k}^i e_i$, $[e_k, e_l] = c_{k,l}^m e_m$, $[e_k, e_m] = c_{k,m}^l e_l$ and $[e_l, e_m] = c_{l,m}^k e_k$. We compute the following Jacobi identities

$$J(e_i, e_j, e_l) = c_{i,j}^k c_{k,l}^m e_m = 0, \quad J(e_i, e_l, e_m) = -c_{l,m}^k c_{i,k}^j e_j = 0,$$

$$J(e_i, e_j, e_m) = -c_{i,j}^k c_{k,m}^l e_l = 0, \quad J(e_j, e_l, e_m) = -c_{l,m}^k c_{j,k}^i e_i = 0,$$

and solve the resulting system of equations, being obtained two families of solutions:

$$\left\{ c_{i,j}^k = c_{l,m}^k = 0, (c_{i,k}^j, c_{j,k}^i) \neq (0, 0), (c_{k,l}^m, c_{k,m}^l) \neq (0, 0) \right\} \quad \text{and}$$

$$\left\{ c_{k,m}^l = c_{k,l}^m = c_{i,k}^j = c_{j,k}^i = 0, c_{l,m}^k \neq 0, c_{i,j}^k \neq 0 \right\}. \quad \square$$

In Figure 9, we show all the possible amalgamations between two triangles.

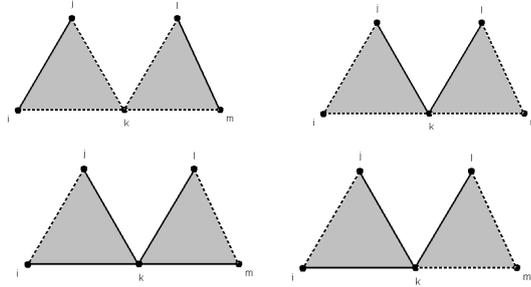


Figure 9: Amalgamation between two triangles.

4.2 Edge addition, deletion and contraction in relation with Lie algebras

Next, we study the edge addition on a digraph. Consider a digraph G and two non-adjacent vertices i and j of G . Then, a new edge can be added to G connecting both vertices. After inserting this additional edge, a new digraph arises from G and we can naturally ask whether this new digraph would also be associated with a Lie algebra. When answering this question, we obtain the following results

Proposition 6. *If G is a digraph from a configuration in Theorem 1, then every digraph arising from G by edge addition is not associated with Lie algebras.*

Proof. First, note that it is not possible to add edges in an oriented 2-cycle. If we have the digraph P_n and add a new edge, then the forbidden configurations $b)$ and $c)$ in Lemma 1 appear. \square

Proposition 7. *Let G be a digraph corresponding to the first configuration in Theorem 2 with r twin vertices. Then we must add $r + 1$ edges to obtain a digraph associated with a Lie algebra. Moreover, the resulting digraph corresponds to the second configuration in Theorem 2.*

Proof. If we add a new edge between two twin vertices, we obtain the forbidden configurations $a)$ or $i)$ from Lemma 1. Analogously, if we add k edges, with $2 < k < r + 1$, then the forbidden configurations $b)$, $f)$, $g)$ or $i)$ in Lemma 1 appear. In case of adding $r + 1$ edges, we consider the procedure indicated in Figure 10. \square

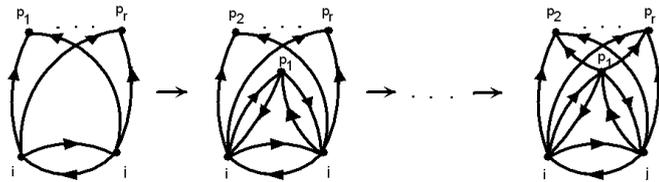


Figure 10: Addition of edges.

Proposition 8. *Let G be a digraph corresponding to the second configuration in Theorem 2 with s twin vertices. Then we must add, at least, $s(s+2)$ edges to obtain a digraph associated with a Lie algebra.*

Proof. Obviously, we cannot add a unique edge to G since the forbidden configurations $f)$ and $i)$ would appear. Therefore, we must add the edges between a twin vertex and a second vertex belonging to an oriented 2-cycle. Moreover, we need to add a new edge connecting each twin vertex to each vertex from oriented 2-cycles to avoid configurations $b)$ and $i)$. \square

The following result is related to the operation called edge deletion for a particular type of digraphs.

Proposition 9. *Let G be a digraph corresponding to the second configuration in Theorem 2 with s twin vertices. Then a digraph corresponding to the first configuration in Theorem 2 can be obtained by deletion of $s + 2$ edges.*

Proof. We only must consider the method indicated in Figure 11. \square

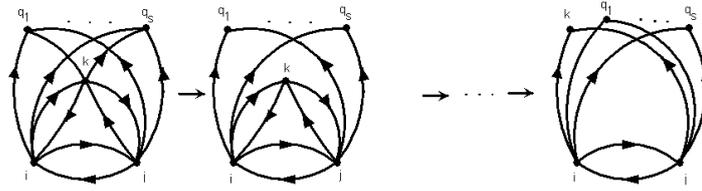


Figure 11: Edge deletion.

Finally, we conclude this subsection with a result about edge contraction for the digraph P_n to generate subgraphs associated with Lie algebras.

Proposition 10. *A subgraph of the digraph P_n is associated with a Lie algebra if and only if it is obtained by successive contractions of edges incident with end vertices.*

Proof. If we apply a contraction by an edge incident with an end vertex, we obtain the digraphs P_{n-1} . In case of contracting another different edge, we get the forbidden configuration *a*) in Lemma 1. \square

5 Algorithm for the amalgamation of two digraphs

In this section, we show an algorithmic procedure to compute the amalgamation of two digraphs associated with Lie algebras. Moreover, we also study if the digraph obtained in the amalgamation is associated with a Lie algebra. Such an algorithm consists of the following two steps:

- a) Compute the amalgamation of two digraphs associated with Lie algebras.
- b) Check if the digraph obtained in the previous step is associated with a Lie algebra.

In order to implement the algorithm, we have used the symbolic computation package MAPLE 12. The libraries `DifferentialGeometry`, `LieAlgebras` and `GraphTheory` must be loaded. In this way, we can use commands related to Lie algebras and Graph Theory.

The first step of this algorithm is executed by the routine `amalgamation`, which receives as input two digraphs G and H . Both digraphs must be defined with the order `Digraph(V, E)`, where V is a list with the vertices of G and E is a set whose elements are the edge (i.e. ordered pairs of vertices) with their corresponding weight. To implement this routine, several local variables are defined and a loop is programmed to compute the amalgamation.

```
> amalgamation:=proc(G,H)
> local U,V,A,B,W,C;
> U:=Vertices(G);
> V:=Vertices(H);
```

```

> A:=Edges(G,weights);
> B:=Edges(H,weights);
> W:=U;
> C:=A union B;
> for i from 1 to nops(V) do
> if member(V[i],W)=true then W:=W;
> else W:=[op(W),V[i]];
> end if;
> end do;
> Ga:=Digraph(W,C);
> return Ga;
> end proc:

```

Now, the representation of this digraph can be obtained with the following sentence

```

> DrawGraph(amalgamation(G,H));

```

Next, we show the implementation of the second step of our algorithm, where we check if the digraph obtained with the amalgamation is associated with a Lie algebra. To do so, we implement the routine `program`, which allows us to build a vector space associated with the digraph. This vector space is the candidate for the bracket product. The routine `program` receives the following two inputs: the list `V` with the vertices of the digraph and the set `E` with its directed, weighted edges. As outputs, we obtain the vector space with basis $\{e_i\}_{i=1}^n$, where e_i corresponds to vertex i in the list `V`, and the brackets associated with the edges in the set `E`. To implement this routine, two local variables `B` and `L` are defined, where `B` saves the basis $\{e_i\}_{i=1}^n$ and `L` is a list containing the indexes of the structure constants from the non-zero brackets.

```

> program:=proc(V,E)
> local B, L;
> B:=[]; L:=[];
> for x from 1 to nops(V) do
> B:=[op(B),e[x]];
> end do;
> for i from 1 to nops(E) do
> if E[i][1][1] < E[i][1][2] then
> L:=[op(L),[[E[i][1][1],E[i][1][2],E[i][1][2]],E[i][2]]];
> else L:=[op(L),[[E[i][1][2],E[i][1][1],E[i][1][2]],E[i][2]]];
> end if;
> end do;
> return _DG(["LieAlgebra",Alg1,[nops(V)],L]);
> end proc:

```

Once the vector space and the structure constants (i.e. the bracket product) are generated by the routine `program`, we must define the law corresponding to these, which is done by evaluating the sentence

```
> DGsetup(program(V,E));
```

After defining this vector space, saved as `Alg1`, we can operate over it. More concretely, running the next sentence, we can test if the Jacobi identities hold for `Alg1`:

```
Alg1 > Query(Alg1,"Jacobi");
```

The vector space `Alg1`, defined by the output of `program`, is a Lie algebra if and only if the answer `true` is obtained for this question. In the affirmative case, `Alg1` saves all the data of the Lie algebra, including its law.

Example 1. Consider digraphs *b)* and *c)* from Figure 5. Now, we computationally define both digraphs by means of the list of vertices and the set of edges.

```
> U:=[1,2,3];
> A:=[[1,2],-1],[2,1],1],[1,3],1],[2,3],1];
> G:=Digraph(U,A);
> V:=[3,4,5];
> B:=[[4,3],1],[4,5],1];
> H:=Digraph(V,B);
```

After defining both digraphs, we compute the amalgamation between them and draw it with the sentence

```
> amalgamation(G,H);
> DrawGraph(amalgamation(G,H));
```

Now, we use the routine `program` with the set of vertices and edges obtained in the amalgamation. This routine returns its associated vector space, which is endowed with a law given by the weights (i.e. structure constants). In this way, we execute the sentence

```
> program(Vertices(amalgamation(G,H)),Edges(amalgamation(G,H),weights));
```

Finally, we check the Jacobi identities to test if this vector space is or not a Lie algebra.

```
> DGsetup(program(Vertices(amalgamation(G,H)),Edges(amalgamation(G,H),weights)));
Alg1 > Query(Alg1,"Jacobi");
> true
```

Since the answer is `true`, the digraph in Figure 11 is associated with a 5-dimensional Lie algebra.

Next, we compute the complexity of the algorithm. To do so, we consider the number of operations carried out in the worst case. We use the big O notation to express the complexity. To recall the big O notation, the reader can consult [12]: given two functions $f, g : \mathbb{R} \rightarrow \mathbb{R}$, we could say that $f(x) = O(g(x))$ if and only if there exist $M \in \mathbb{R}^+$ and $x_0 \in \mathbb{R}$ such that $|f(x)| < M \cdot g(x)$, for all $x > x_0$.

We denote by $N_i(n)$ the number of operations when considering the step i . This function depends on the dimension n of the Lie algebra. Table 1 shows the number of computations and the complexity of each step, as well as indicating the name of the routine corresponding to each step.

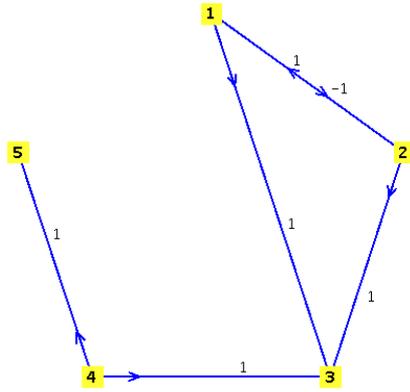


Figure 12: Digraph of Example 1.

Table 1: Complexity and number of operations.

Step	Routine	Complexity	Operations
1	amalgamation	$O(n)$	$N_1(n) = 14 + \sum_{i=1}^n 1$
2	program	$O(n)$	$N_2(n) = 5 + 2 \sum_{i=1}^n 1$

Acknowledgment

This work has been partially supported by MTM2010-19336 and FEDER.

References

- [1] [A. Carriazo, L.M. Fernández, J. Núñez, Combinatorial structures associated with Lie algebras of finite dimension, Linear Algebra Appl. 389 \(2004\), 43–61.](#)
- [2] [M. Ceballos, J. Núñez, A.F. Tenorio, Complete triangular structures and Lie algebras, Int. J. Computer Math. 88:9 \(2011\), 1839–1851.](#)
- [3] [M. Ceballos, J. Núñez, A.F. Tenorio, Study of Lie algebras by using combinatorial structures, Linear Algebra Appl. 436 \(2012\), 349–363.](#)
- [4] [M. Ceballos, J. Núñez, A.F. Tenorio, Combinatorial structures and lie algebras of upper-triangular matrices, Appl. Math. Lett. 25 \(2012\), 514–519.](#)
- [5] R. Diestel, Graph Theory, 4th. Edition, Springer-Verlag, Heidelberg, 2010.
- [6] [L.M. Fernández, L. Martín-Martínez, Lie algebras associated with triangular configurations. Linear Algebra Appl. 407 \(2005\), 43–63.](#)
- [7] [E.E. Levi, Sulla struttura dei gruppi finiti e continui, Atti. Accad. Sci. Torino 40 \(1905\), 551–565.](#)
- [8] A.I. Malcev, Solvable Lie algebras. Trans. Amer. Math. Soc. Transl. 9 (1962), 228–262.
- [9] [M. Primc, Basic representations for classical affine Lie algebras, J. of Algebra 228 \(2000\), 1–50.](#)
- [10] [J.P. Serre, Algèbres de Lie Semi-Simples Complexes, Benjamin Inc., New York, 1996.](#)
- [11] [V.S. Varadarajan, Lie Groups, Lie Algebras and Their Representations, Springer, New York, 1984.](#)
- [12] H.S. Wilf, Algorithms and Complexity, Prentice Hall, 1986.