

The Use of Graph Isomorphisms in Chemical Graph Theory

by

Megan Hanks

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
BACHELOR OF SCIENCE (HONS.)

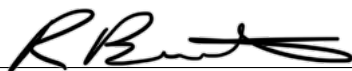
in the

Department of Mathematics & Statistics


THOMPSON RIVERS  UNIVERSITY

©Megan Hanks 2025

We accept this thesis as conforming to the required standards:



Dr. Richard Brewster
Dept. of Mathematics & Statistics
Thesis Supervisor



Dr. Lucas Mol
Dept. of Mathematics & Statistics



Dr. Jessica Allingham
Dept. of Chemistry

Dated July 2 2025, Kamloops, British Columbia, Canada

THOMPSON RIVERS UNIVERSITY
DEPARTMENT OF MATHEMATICS & STATISTICS

Permission is herewith granted to Thompson Rivers University to circulate and to have copied for non-commercial purposes, at its discretion, the above title upon request of individuals or institutions.



Signature of Author

THE AUTHOR RESERVES OTHER PUBLICATION RIGHTS, AND NEITHER THE THESIS NOR EXTENSIVE EXTRACTS FROM IT MAY BE PRINTED OR OTHERWISE REPRODUCED WITHOUT THE AUTHOR'S WRITTEN PERMISSION.

THE AUTHOR ATTESTS THAT PERMISSION HAS BEEN OBTAINED FOR THE USE OF ANY COPY-RIGHTED MATERIAL APPEARING IN THIS THESIS (OTHER THAN BRIEF EXCERPTS REQUIRING ONLY PROPER ACKNOWLEDGEMENT IN SCHOLARLY RIGHTING) AND THAT ALL SUCH USE IS CLEARLY ACKNOWLEDGED.

Abstract

This thesis examines the use of graphs as models for chemical molecules and processes. We begin by defining chemical graphs and their properties.

We then define the subgraph isomorphism problem and examine two algorithms which can be used to solve it: Ullmann's algorithm and the SubGemini algorithm. We further discuss the application of these algorithms to chemical graph theory and provide examples for each.

In addition to structural models, we describe an algorithm for the identification of intermediates and products of a reaction modeled as a network. The algorithm is demonstrated by means of an example.

Acknowledgements

I would like to extend my sincere gratitude to my supervisor, Dr. Richard Brewster. This project would not have been possible without his guidance, expertise and patience.

I would also like to thank my committee members, Dr. Lucas Mol and Dr. Jessica Allingham for their time and feedback on this project.

Finally I would like to thank the Department of Mathematics and Statistics at Thompson Rivers University, and all the faculty members from whom I have gained a deeper appreciation for mathematics and research.

Contents

Abstract	iii
Acknowledgements	iv
1 Introduction	1
2 Background in Graph Theory	2
2.1 Basic Definitions	2
2.2 Other Types of Graphs	5
2.2.1 Digraphs	5
2.2.2 Chemical Graphs	6
2.2.3 Subgraphs and Induced Subgraphs	8
2.2.4 Bipartite Graphs	10
2.3 Matrices	10
2.3.1 Adjacency Matrices	10
2.3.2 Other Types of Matrices	12
3 Background in Chemistry	14
3.1 Common Molecules and Functional Groups	15
3.2 R-Group Notation	15
3.3 Drawing Molecular Structure	16
3.3.1 Corey–Pauling–Koltun (CPK) Colouring	17
4 Graph Isomorphism	19
4.1 Graph Isomorphism Problem	19
4.1.1 Similar and Permutation Similar Matrices	20
4.1.2 Matrix Representation of an Isomorphism	22
4.2 Ullman’s Algorithm	24
4.2.1 The Algorithm	25
4.2.2 Labeling Condition	27
4.2.3 Neighbourhood Condition	27
4.2.4 Example	27

4.3	The Gemini and SubGemini Algorithms	30
4.3.1	Gemini and GeminiII	30
4.3.2	SubGemini	36
5	Reaction Networks	39
5.1	Orlova's Algorithm	39
5.1.1	Algorithm Components	40
5.2	Reaction Network for Ethyl Acetoacetate	40
5.2.1	Input: Initial Species	40
5.2.2	Input: Patterns	41
5.2.3	Input: Reaction Rules	42
5.2.4	Network	45
5.2.5	Output: List of Chemical Species	47
6	Conclusion	48
	References	48

List of Figures

2.1	A graph on 5 vertices.	3
2.2	The graph on the left is labeled from the set $\{C, O, N\}$. The graph on the right uses colours and shapes to distinguish vertices.	4
2.3	The same underlying graph with two different colourings. The colouring on the left is a proper colouring.	4
2.4	A directed graph showing a simple chemical reaction involving two reactants and a single product.	5
2.5	A graph on 3 vertices, and a labeling of that same graph to model a molecule of water (H_2O).	6
2.6	Graphs representing carbon dioxide (CO_2) and ozone (O_3).	7
2.7	The structure of D-Galacturonic Acid.	7
2.8	D-Galacturonic Acid with some hydrogen atoms removed.	8
2.9	The graph G from Figure 2.1 and two smaller graphs, G' in the center and G'' on the right.	9
2.10	Two induced subgraphs of G	9
2.11	Two drawings of a bipartite graph with with partite sets $A = \{a_1, a_2, a_3\}$ and $B = \{b_1, b_2, b_3\}$	10
2.12	The graph G and its corresponding adjacency matrix A	11
2.13	Another graph, G' with corresponding adjacency matrix B	11
2.14	The structure of P and P^T	12
3.1	The structure of (from left to right) a ketone, a carboxyl and an ester.	15
3.2	The General Form of a Carboxylic Acid and an Alcohol.	15
3.3	Methanol and Ethanol with the R groups shown in the box.	16
3.4	The structure of ethyl acetoacetate which include an ethyl group, shown in blue (dashed), an ester shown in red (solid), and a ketone group shown in green (dotted).	16
3.5	The Kekulé and line-bond formulas for ethanol.	17
3.6	Line-bond formulas for benzene and ethyl acetoacetate.	17
4.1	A graph G with $V(G) = \{a, b, c, d\}$ and a second graph G' with $V(G') = \{w, x, y, z\}$	19

4.2	Isomorphic bipartite graphs G and G' with partite sets $A = \{a_1, a_2, a_3\}$ and $B = \{b_1, b_2, b_3\}$	20
4.3	Graph G on the left, and graph H on the right.	23
4.4	Graphs G and H	25
4.5	Graph G corresponding to a molecule of D-Galaturonic acid.	27
4.6	The graph H corresponding to a hydroxyl functional group, and its adjacency matrix B	28
4.7	Graph G corresponding to a molecule of D-Galaturonic acid with labels assigned based on type of atom.	29
4.8	The graph H corresponding to a hydroxyl functional group, and its adjacency matrix B	29
4.9	The structure of cis-2-butene on the left, and a benzene ring on the right. . . .	31
4.10	The partitioning of hydrogen atoms in a molecule of cis-2-butene.	32
4.11	Determining the structure of cis-2-butene from one of the labels.	33
4.12	An H-NMR of 2-cis-butene showing two distinct peaks [4].	34
4.13	A graph representation of propen-2-ol using CPK colouring.	35
4.14	A graph representation of 2-propen-1-ol shown using CPK colouring.	35
4.15	4-Nitrobenzyl Acetoacetate and the carbonyl pattern. The dashed lines and vertices labeled X are corrupted vertices. The subscripts are shown for ease of identification.	37
5.1	The same simple chemical system modeled as a CRN and as an RRN.	39
5.2	The structure of ethyl acetoacetate and methanol.	41
5.3	The reaction network generated by Orlova's Algorithm generated from ethyl acetoacetate and methanol.	46

List of Tables

3.1	Common Functional Groups	15
3.2	CPK colourings for some common atoms.	18
5.1	Summary of Reactions.	43
5.2	The reaction intermediates from the reaction network for ethyl acetoacetate and methanol.	47
5.3	Final products of the reaction network for ethyl acetoacetate and methanol. . .	47

Chapter 1

Introduction

Graph theory is an area of mathematics that studies structures composed of points and the connections between them. These structures, which are referred to as graphs, have many applications and are used for modeling the relationships between objects in a wide array of different fields. Some applications of graph theory include modeling social networks, route optimization, search and sort type algorithms and even language structure. One of the first mathematicians to employ this type of modeling technique was Arthur Cayley [3]. In 1874, he published a paper detailing a method for enumerating isomers using diagrams involving linked points. From these types of linked-point models came the field of chemical graph theory which applies graph theory to the modeling of molecules, chemical properties, and reactions.

In this paper, we define chemical graphs and the ways in which graph theory can be applied to the study of chemical molecules and reactions. This includes the use of labeling and hydrogen-reduction to most efficiently model the structure of individual molecules, as well as the parts of a molecule which undergo reactions.

We also discuss the subgraph isomorphism problem and how it relates to the identification of functional groups in a molecule. Two algorithms for subgraph isomorphism, Ullmann's Algorithm [13] and the SubGemini Algorithm [10], are described along with conditions for their application to chemical graphs. Each algorithm is also applied to chemical examples.

Finally, we describe reaction networks, and an algorithm for finding all the chemical species and transformations that occur within that network [11]. This algorithm is demonstrated using an example with two initial species: methanol and ethyl acetoacetate.

Chapter 2

Background in Graph Theory

We will define key terms from graph theory and chemistry used in this thesis following Bondy and Murty for graph theory [2], and IUPAC (the International Union of Pure and Applied Chemistry) gold book for chemistry [9]. We refer the reader to these texts for terminology not defined in this thesis.

2.1 Basic Definitions

Definition 2.1. A *graph* $G = (V, E, \psi_G)$ is a non-empty set of vertices V together with a set of edges, E . The *incidence function* ψ_G associates each edge in E with an unordered pair of vertices in V . If $\psi_G(e) = \{u, v\}$ then we say that e *joins* u and v . For the purposes of this paper, a pair of vertices $\{u, v\}$ will be denoted as uv .

All graphs in this work are finite, which means that they contain a finite number of vertices and edges. Although a graph is an ordered triple, in this paper a graph will be denoted by $G = (V, E)$ where the incidence function ψ_G is not explicitly stated.

Example 2.2. The following is an example of a graph G with $V(G) = \{a, b, c, d, e\}$, $E(G) = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7\}$ and ψ_G is indicated by the diagram. For example $\psi_G(e_1) = ab$, $\psi_G(e_2) = ac$ etc.

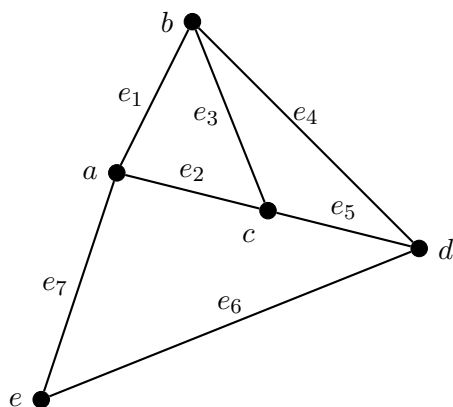


Figure 2.1: A graph on 5 vertices.

If $\psi(e) = uv$, the edge e is said to be *incident* to u and v . The vertices u and v are the *end points* of e . In Figure 2.1 e_2, e_3 and e_5 are incident to c . Two edges that share an endpoint are said to be *adjacent*, as are two vertices connected by an edge. In Figure 2.1 a, c and d are adjacent to b .

Definition 2.3. The number of edges incident to a particular vertex v is called the *degree* of v , which is denoted as $\deg(v)$. The *maximum degree* of G , denoted $\Delta(G)$, is the highest degree of any vertex in G . Conversely, the *minimum degree* of G , $\delta(G)$, is the lowest degree of any vertex in G .

In the graph G drawn in Figure 2.1, $\deg(a) = 3$ and $\deg(e) = 2$. Since the maximum degree of any vertex in G is 3, we have $\Delta(G) = 3$. It is possible for a vertex to have a degree of zero, as would be the case for the lone vertex in the graph containing only one vertex and no edges.

It is interesting to note that in early graph theory works, the degree was called the *valence* of a vertex. In chemistry, the valence of an atom is the number of bonds that the atom is able to form with other atoms. It is directly related to the valence electrons which are the outermost electrons of the atom.

Definition 2.4. The number of vertices in a graph G is called the *order* of G . The number of edges in G is called the *size* of G .

Unlike $\Delta(G)$, the order of a graph must always be greater than or equal to 1, since the set of vertices V is non-empty. Many papers on chemical graph theory use the word *size* to refer to the number of vertices in a graph, however this paper will use the definitions as given above. Generally, the vertices of a graph are distinguished only by their connections to other vertices. However, it is possible to assign *labels* to further distinguish between vertices.

Definition 2.5. Given a graph $G = (V, E)$ and a set of labels L , a *vertex labeled graph* or simply a *labeled graph* is a triple $G_\ell = (\ell, V, E)$ where $\ell : V \rightarrow L$ assigns each vertex v a label

$\ell(v)$. An *edge labeled graph* is similar, except that labels are assigned to edges. As an abuse of notation, we may write G to denote either the graph or the labeled graph.

Note that the vertices in the graph G from Figure 2.1 have unique labels to distinguish them. This is not always the case, and more than one vertex in the same graph can have identical labels as in Figure 2.2.

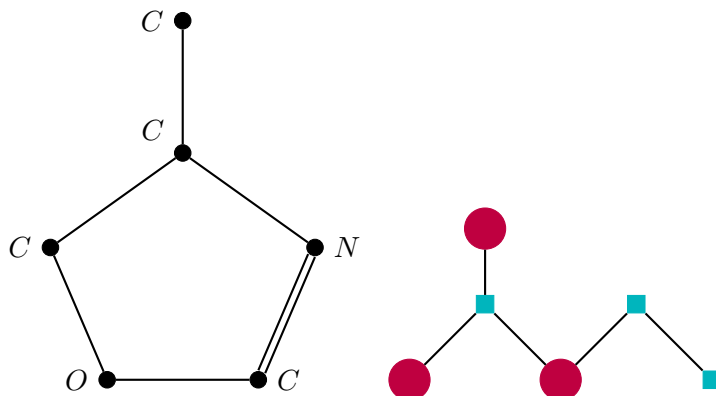


Figure 2.2: The graph on the left is labeled from the set $\{C, O, N\}$. The graph on the right uses colours and shapes to distinguish vertices.

A labeled graph $G_\ell = (\ell, V, E)$ has an *underlying graph* $G = (V, E)$. Two labeled graphs may have the same underlying graph and differ only in their labeling.

A special case of graph labeling is *graph colouring*. Colours can be used simply to differentiate between vertices in the same way that labels do, however a *proper graph colouring* refers to a graph where colours are assigned to labels so that no two adjacent vertices have the same label. Consider the underlying graph G shown below.

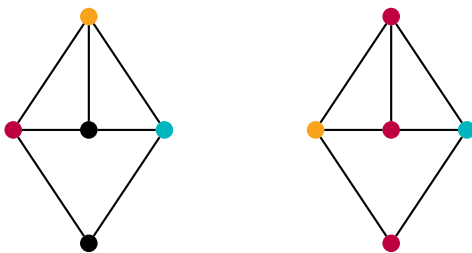


Figure 2.3: The same underlying graph with two different colourings. The colouring on the left is a proper colouring.

Labeled graphs are particularly useful for modeling chemical molecules, as the labels

allow for differentiation between types of atoms in a structure. In reaction networks, nodes can be labeled as chemical species and reaction types.

2.2 Other Types of Graphs

2.2.1 Digraphs

Generally, the edges of a graph are considered to be bidirectional, meaning it goes in both directions; however, there are times when it is useful to specify the direction of an edge. This can be thought of as the flow from one vertex to another. In all the graphs discussed so far in this paper, the edges are bidirectional. Such graphs are called undirected when needed, to distinguish from directed graphs.

There are applications for this type of model, such as the flow of traffic along roads, or the flow of messages in a social network. Flow is important for chemical graph theory when it comes to modeling the flow of reactions within a chemical system. This direction of flow can be represented by an arrow, as in Figure 2.4.

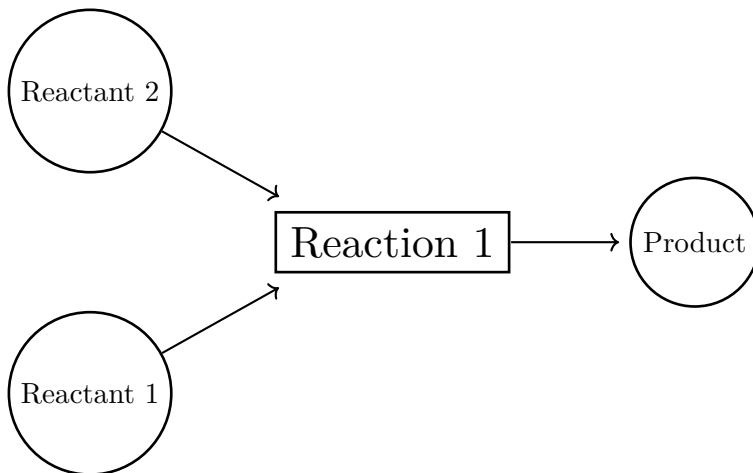


Figure 2.4: A directed graph showing a simple chemical reaction involving two reactants and a single product.

Formally a *directed graph* or *digraph* is a type of graph G which differs from an undirected graph in that edges between vertices are ordered pairs and are typically referred to as *arcs*. Each arc has a specified orientation. This is to say that if $u, v \in V(G)$ and uv is an arc in G , then u is the *tail* of uv and v is the *head*. Alternatively, if $\psi(e) = (u, v)$ then e is said to *join* u to v , but does not join v to u .

A *network* is a digraph where vertices are considered to be *sources*, *sinks* or *intermediate vertices*. A *source* is a vertex that initiates a process and all edges incident to that vertex are directed away from it. A *sink* is a vertex that only has in directed arcs, while *intermediate*

vertices have arcs directed toward and away from them.

In Example 2.4, the sources are Reactant 1 and Reactant 2, while the sink is the Product vertex. The only intermediate vertex is the Reaction 1 vertex. In the case of reaction networks, the sources will be any initial reactants in a system, while the sinks will be any final non-reactive products. Intermediate vertices will include reaction steps and intermediate chemical species. Figure 2.4 is an example of a very simple *reaction route network* or *RRN*. These will be discussed in greater detail in Chapter 5.

2.2.2 Chemical Graphs

In addition to chemical reactions, chemical molecules can also be modeled as graphs. In some of his later works, Arthur Cayley referred to these diagrams representing chemical structures as *plerograms*. In *chemical graphs*, nodes represent atoms, and edges represent the bonds between atoms. Labeling is used to differentiate between different atoms and groupings of atoms. The following example shows the importance of labeling when modeling a chemical molecule as a graph.

Example 2.6. The figure on the left is an example of a graph on three vertices with $V(G) = \{u, v, w\}$ and $E(G) = \{uv, vw\}$. By labeling the vertices as $\ell(u) = H$, $\ell(v) = O$ and $\ell(w) = H$, the graph can be used to model a molecule of water with chemical formula H_2O . In this figure colors have been added which correspond to the label of the vertices.

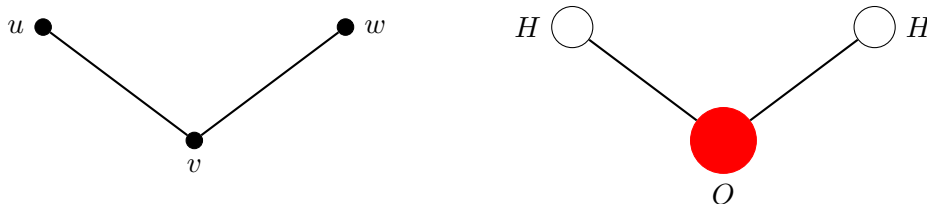


Figure 2.5: A graph on 3 vertices, and a labeling of that same graph to model a molecule of water (H_2O).

We can see that the labels allow for the distinction of different atoms in the molecule. The following are two more examples of graphs with the same set of vertices, but a different labeling. The molecules represented are carbon dioxide (CO_2) and ozone (O_3). Note the additional edges that have been added to reflect the bonds present in these two molecules. As a result, the underlying graph will have different edges with the same end points. Such edges are called *parallel*.

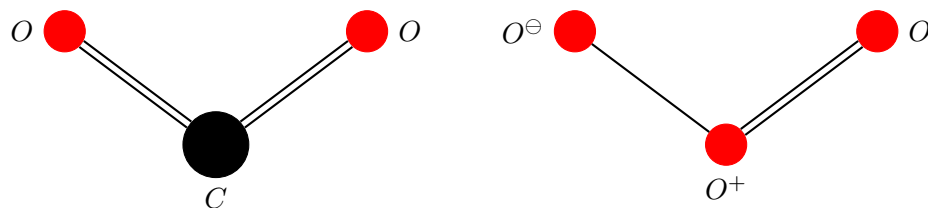


Figure 2.6: Graphs representing carbon dioxide (CO_2) and ozone (O_3).

For larger and more complex molecules, there are times when it is appropriate to omit certain atoms, such as hydrogen atoms, or to use a labeled vertex to represent multiple atoms rather than a single atom. In fact, in chemical graph theory, it is more common to omit hydrogen atoms than it is to use true plerograms which show every atom as an individual vertex [8]. Consider the molecular structure of D-Galacturonic Acid shown in Figure 2.7.

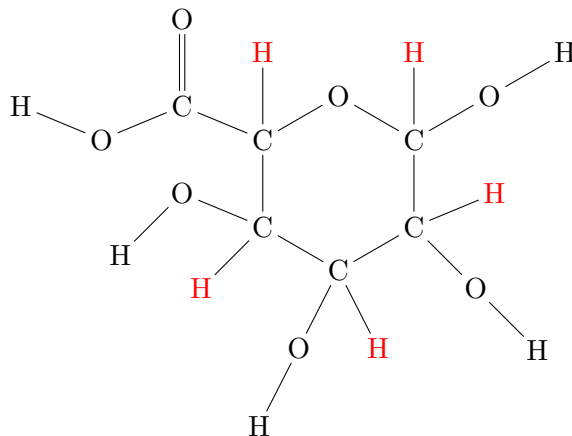


Figure 2.7: The structure of D-Galacturonic Acid.

We can see that in the full molecule of Galacturonic acid we have 23 nodes, and therefore the corresponding adjacency matrix, which is defined in Section 2.3.1, is a 23×23 matrix. By removing the hydrogen atoms attached to the carbon atoms (shown in red), the reduced graph has 18 nodes and thus the adjacency matrix is an 18×18 matrix.

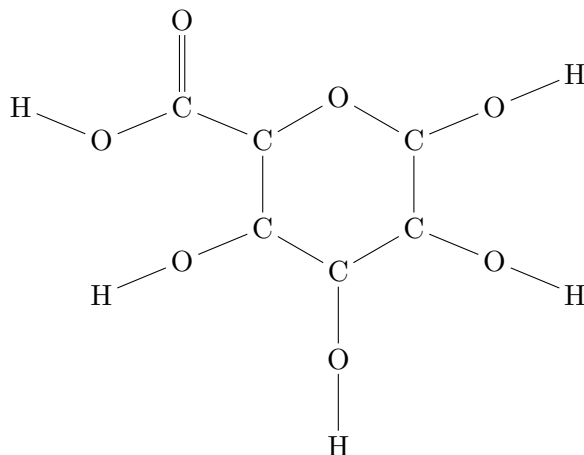


Figure 2.8: D-Galacturonic Acid with some hydrogen atoms removed.

Although there are cases where this will not be possible, it is possible to remove hydrogen atoms from most models while preserving the relevant aspects of the molecules structure. In fact, Arthur Caley coined the term *kenogram* to refer to “hydrogen suppressed” graphs [1] and it has been noted that these are used far more often than plerograms [8]. If required, it is also possible to represent small groupings of atoms as a single vertex, for example using distinct labels for carbons based on the number of hydrogen atoms they are bonded to. Using this method, $(-CH_3)$ would become a single vertex as opposed to 4 vertices, and different labels would be used for $(-CH_2-)$, $(-CH-)$ and $(-C-)$ groups. In the example of D-galacturonic acid, one could further reduce the number of vertices by using labels to differentiate between oxygen bonded to a hydrogen and the oxygen molecules bonded to two non-hydrogen atoms. When reducing the number of vertices in this manner, it is important to carefully consider which patterns and substructures will be significant.

2.2.3 Subgraphs and Induced Subgraphs

Definition 2.7. A graph G' is a *subgraph* of the graph G if $V(G') \subseteq V(G)$ and $E(G') \subseteq E(G)$ where for each edge $uv \in E(G')$, we have $u, v \in V(G')$.

Consider the graph from Figure 2.1 and two graphs constructed on a subset of $V(G)$.

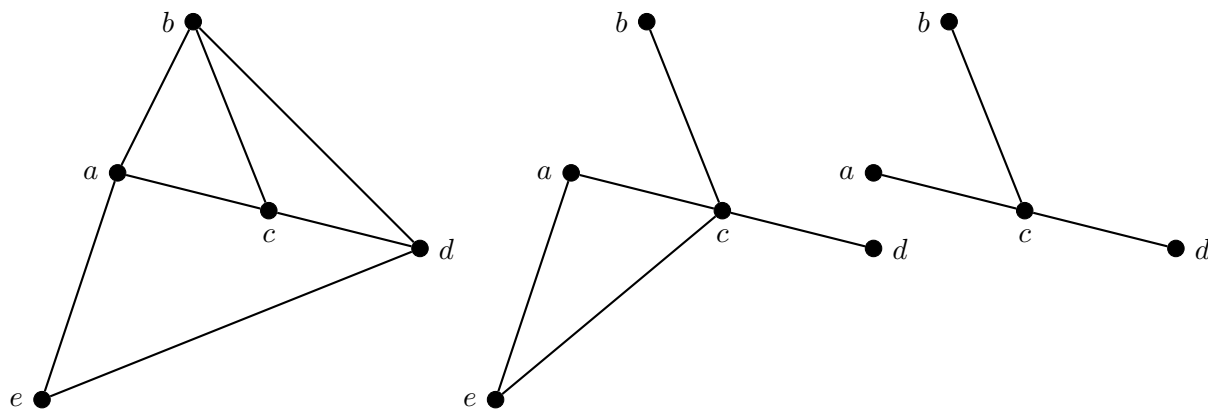


Figure 2.9: The graph G from Figure 2.1 and two smaller graphs, G' in the center and G'' on the right.

The graph G' is not a subgraph of G . This can be proved simply by observing that vertex c in G' has degree 4, while $\Delta(G) = 3$. Therefore, G' cannot be a subgraph of G . The graph G'' , however, is a subgraph of G .

Definition 2.8. A subgraph G' is an *induced* subgraph of G if whenever $u, v \in V(G')$ and $uv \in E(G)$ we have $uv \in E(G')$. The subgraph G' containing vertices $S = V(G')$ is called the *subgraph of G induced by S* and is denoted by $G[S]$.

Of the previous examples, neither is an induced subgraph. G' not a subgraph, and therefore cannot be an induced subgraph, while G'' is missing the edges ab and bd . The subgraphs of G induced by $\{b, c, d, e\}$, and $\{a, c, d, e\}$ are shown below.

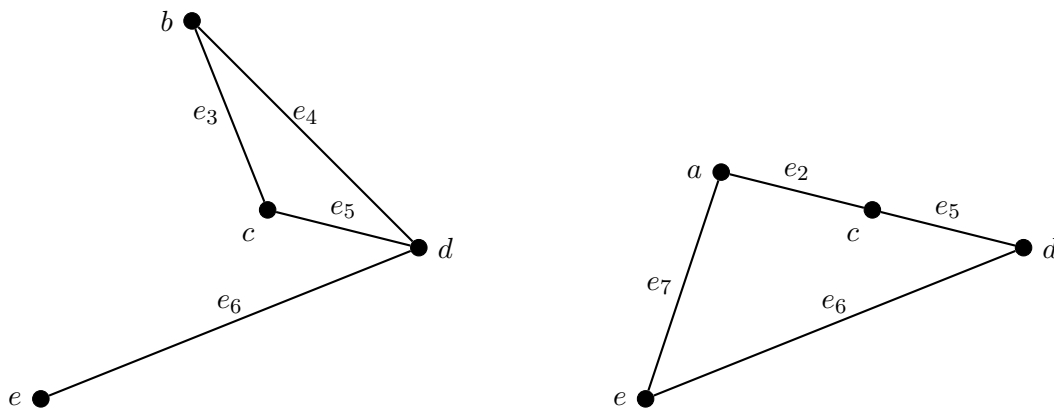


Figure 2.10: Two induced subgraphs of G .

2.2.4 Bipartite Graphs

Definition 2.9. A graph or digraph, $G = (V, E)$ is *bipartite* if the vertices of G can be partitioned into two sets, A and B , called partite sets, such that every edge in the graph has one end in A and the other in B .

Bipartite graphs can be drawn in such a way as to make the two partite sets visually distinct using different shapes or colours for the vertices.

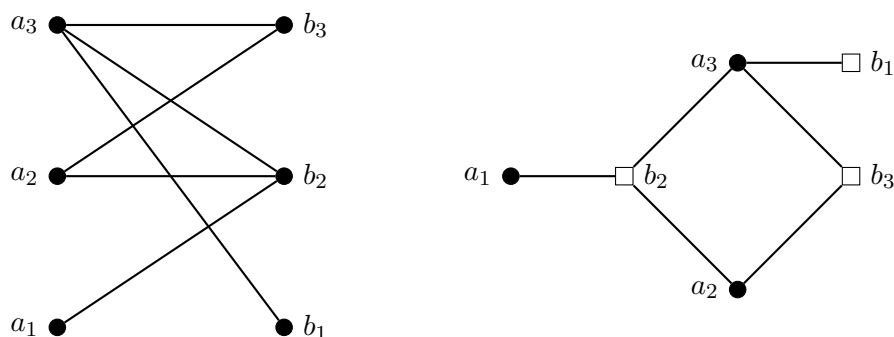


Figure 2.11: Two drawings of a bipartite graph with partite sets $A = \{a_1, a_2, a_3\}$ and $B = \{b_1, b_2, b_3\}$.

These types of graphs can be very useful in a number of applications, such as modeling the association of individuals with groups or affiliations within a social network, or the association between specific genes and types of cancer. In chemical graph theory, bipartite graphs are used to model reaction networks, where nodes are classified as either a chemical species node or a reaction node. Chemical species nodes have directed edges towards those reactions that they are able to participate in. Reaction nodes then have edges towards the products of those reactions. A very simple example is shown in Figure 2.4. Reaction networks are discussed in greater detail in Chapter 5.

2.3 Matrices

2.3.1 Adjacency Matrices

A common way to represent a graph G , is as an *adjacency matrix*. If G has n vertices, then its adjacency matrix, A is an $n \times n$ matrix with one row and one column for each vertex. The entry a_{ij} is the number of edges that exist between vertex i and vertex j . For example, we have $a_{ij} = 1$ if there is a single edge between vertex i and vertex j , and if there are no edges between i and j , then $a_{ij} = 0$.

The adjacency matrix A for a graph G used to represent a chemical molecule will have the following properties.

1. $a_{ij} = a_{ji}$ (as bonds are undirected).
2. $A = A^T$ (hence A is symmetric).
3. The diagonal entries of A are all zero. These are entries a_{ij} where $i = j$.

Example 2.10. Consider the graph G with order 4 shown in Figure 2.12. The set of vertices is $V(G) = \{a, b, c, d\}$ and the set of edges is $E(G) = \{ab, bc, bd\}$. The adjacency matrix for this graph is also shown in Figure 2.12. We can see that the edges in E correspond to the non-zero entries in A .

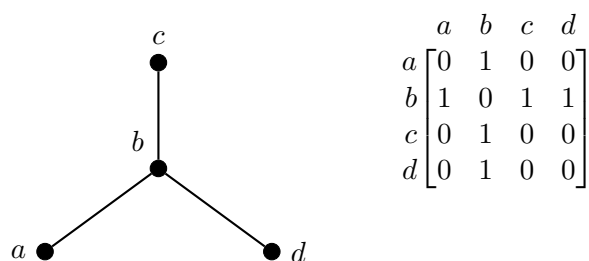


Figure 2.12: The graph G and its corresponding adjacency matrix A .

Consider the graph G' with $V(G') = \{w, x, y, z\}$ shown in Figure 2.13.

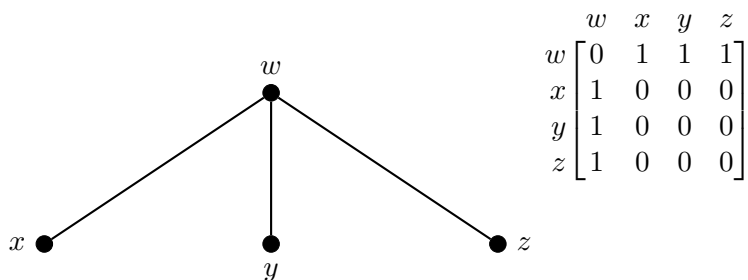


Figure 2.13: Another graph, G' with corresponding adjacency matrix B .

We can see that both matrices A and B are symmetric and zero-diagonal, and that both contain the same number of non-zero entries. We also see a natural correspondence between the graphs: $a \rightarrow x$, $b \rightarrow w$, $c \rightarrow y$ and $d \rightarrow z$. Despite the matrices looking different, we can obtain B from A by reordering the rows and columns as b, a, c, d . This is the heart of the “graph isomorphism problem” discussed in great detail in Chapter 4.

2.3.2 Other Types of Matrices

Definition 2.11. The *identity matrix* is a square matrix with 1's as its diagonal entries and all other entries equal to 0. This is equivalent to saying that for an identity matrix $I = [a_{ij}]$, $a_{ij} = 1$ if $i = j$; otherwise, $a_{ij} = 0$.

By the above definition, each row and column of the identity matrix has exactly one 1. By re-arranging the rows and columns of the identity matrix, we obtain a matrix that retains this property, however, the resulting matrix no longer has the condition that $a_{ij} = 1$ if $i = j$. We call the process of re-arranging rows and columns in a matrix *permuting* the rows and columns.

Definition 2.12. A *permutation matrix* is an $n \times n$ matrix with exactly one 1 in each row and each column.

Permutation matrices have the property, that multiplying a matrix A on the left by a permutation matrix P the rows of A are permuted. When multiplying A by P on the right, the columns of A are permuted.

More formally, let P be a permutation matrix, and suppose that the unique 1 in row i is found in column j , which is to say $p_{ij} = 1$ and $p_{ik} = 0, k \neq j$. Let A be an $n \times n$ matrix and $B = PA$. Then $b_{ik} = \sum_{t=1}^n p_{it} \cdot a_{tk} = p_{ij} \cdot a_{jk} = a_{jk}$, since p_{ij} is the only non-zero entry in row i of P . Thus the i th row of the matrix PA will be the j th row of A . By similar reasoning, P^T has a unique 1 in column i at row j . Hence the i th column of AP^T will be the j th column of A . We record this as the following observation.

$$P = i \begin{bmatrix} & & & j & & \\ & & & \vdots & & \\ 0 & \cdots & 0 & 1 & 0 & \cdots \\ & & & \vdots & & \end{bmatrix} \quad P^T = j \begin{bmatrix} & & i & & \\ & & 0 & & \\ & & \vdots & & \\ 0 & \cdots & 0 & 1 & 0 & \cdots \\ & & & \vdots & & \\ & & & 0 & & \end{bmatrix}$$

Figure 2.14: The structure of P and P^T .

Observation 1. Let P be a permutation matrix, and A another matrix. If $p_{ij} = 1$, then row j of PA is the same as row i of A . Further, if $p_{ij} = 1$, then $p_{ji}^T = 1$ and column i of $AP^T =$ column j of A .

Furthermore, we can use these together to get the following observation.

Observation 2. Suppose for a permutation matrix P , $p_{ij} = 1$ and $p_{kl} = 1$. Then the entry a_{jl} in the matrix A is equal to the entry b_{ik} in the matrix $B = PAP^T$.

The following example shows this principle for a 3×3 matrix A . For this particular P , we have $P = P^T$; however, this is not always the case. First we see the effect of multiplying A

by P on the left and by P^T on the right.

$$PA = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} = \begin{bmatrix} a & b & c \\ g & h & i \\ d & e & f \end{bmatrix}$$

$$AP^T = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} a & c & b \\ d & f & e \\ g & i & h \end{bmatrix}$$

Now, if both are applied simultaneously, we see a permutation of both rows and columns.

$$PAP^T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} a & c & b \\ g & i & h \\ d & f & e \end{bmatrix}$$

Returning to the adjacency matrices in Figure 2.12 and Figure 2.13, we see that the matrix B can be obtained by multiplying A by a permutation matrix on the left and by its transpose on the right.

$$B = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} AP^T$$

When $B = PAP^T$ for a permutation matrix P , we say A and B are *permutation similar*. In the case of adjacency matrices, this corresponds to relabeling or reordering the vertices, but it does not change the structure of the graph. *Similar* and *permutation similar* matrices will be discussed in greater detail in Section 4.1.1.

It is also important to note that the transpose of a permutation matrix is also its inverse. Consider a permutation matrix $P = [p_{ij}]$ and inverse P^{-1} . By definition of the inverse, $PP^{-1} = P^{-1}P = I$ where I is the identity matrix and, by definition of the transpose, $P^T = [p_{ji}]$. Suppose p_{ij} is the unique 1 in row i of P . If we take the product of PP^T , then by Observation 1 the i th column of PP^T is the j th column of P . Since the j th column of P has its unique 1 in the i th row, the unique 1 in column i of PP^T will occur in row i . By this logic, all the 1's in PP^T will occur as diagonal entries, meaning $PP^T = I$. The same can be shown for P^TP .

Chapter 3

Background in Chemistry

In order to understand the application of graph theory to chemical molecules and reactions, some basic terminology is required.

Definition 3.1. An *atom* is the smallest unit of a chemical element.

Some common atoms, particularly in organic compounds are carbon, oxygen, hydrogen, nitrogen and metals like sodium or magnesium. Atoms are often referenced by their one or two letter abbreviation from the periodic table. The abbreviations for atoms referenced in this paper include C for carbon, O for oxygen, and H for hydrogen. Other atoms may have less obvious abbreviations such as Fe for iron or Na for sodium.

Definition 3.2. Atoms bond with other atoms to form *molecules*. Molecules can contain atoms of the same element, or atoms of different elements. *Chemical compounds* are molecules which contain more than one type of atom.

The terms molecule and compound are often used interchangeably, as most molecules are compounds; however, there are a number of common molecules that are not compounds such as, H_2 , N_2 and O_3 . The term *chemical species* includes molecules, compounds as well as individual atoms or other chemical complexes that may be involved in a chemical reaction.

Definition 3.3. A *functional group* is a grouping of atoms within a larger molecule that exhibits a specific set of chemical properties. Functional groups play an important role in the reactivity of a molecule. Some common functional groups are listed in the next section.

For the purposes of modeling chemical molecules, it will be useful to think of functional groups as subgraphs of the larger molecular graph. When thought of in this way, functional groups may be referred to as *patterns*. In the context of chemical graph theory, a pattern refers to the portion of the graph that will undergo a transformation during a particular reaction. A molecule may contain more than one pattern, and the pattern of interest may depend on the specific reaction being studied. In order to determine how a molecule will react, it is important to determine which patterns it contains.

3.1 Common Molecules and Functional Groups

There are several common functional groups with distinctive properties that are frequently seen in organic chemistry. Several of these groups are listed in Table 3.1.

Molecular Formula	IUPAC name	Chemical characteristic
$-CH_3$	methyl group	small, non-polar, non-reactive
$-CH_2CH_3$	ethyl group	non-polar, non-reactive
$-OH$	hydroxyl group	polar, reactive
$-COH$	aldehyde	polar, reactive
6 membered carbon ring	phenyl, benzene	bulky, non-reactive

Table 3.1: Common Functional Groups

Other important functional groups include ketones, carboxyls and esters. These are shown below, with dotted lines showing where these groups would connect to the rest of the molecule.

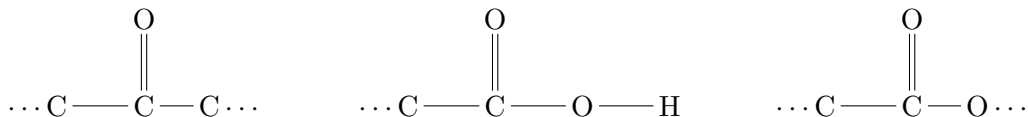


Figure 3.1: The structure of (from left to right) a ketone, a carboxyl and an ester.

3.2 R-Group Notation

When referring to chemical structures, it is common to use the letter R to denote any grouping of molecules where a carbon or hydrogen is directly bonded to the atom shown. Different R groups are differentiated through using dashes or apostrophes (R, R', R''). Consider the following example.

Example 3.4. The molecules shown below including an R group are the general forms of a carboxylic acid and an alcohol.



Figure 3.2: The General Form of a Carboxylic Acid and an Alcohol.

If an alcohol has an R' group that is a methyl group or an ethyl group, then we have the specific alcohol methanol or ethanol respectively. The structure of these two alcohols is shown in Figure 3.3.

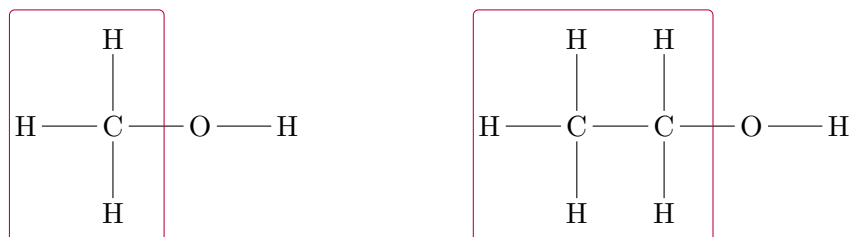


Figure 3.3: Methanol and Ethanol with the R groups shown in the box.

Molecules can also contain multiple function groups, as with ethyl acetoacetate shown in Figure 3.4 which contains an ethyl group, an ester and a ketone.

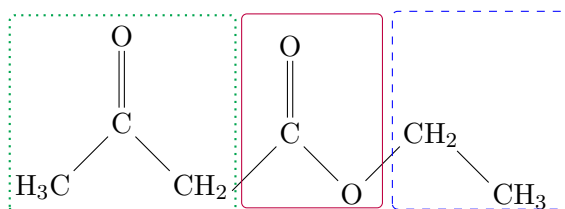


Figure 3.4: The structure of ethyl acetoacetate which include an ethyl group, shown in blue (dashed), an ester shown in red (solid), and a ketone group shown in green (dotted).

3.3 Drawing Molecular Structure

There are several ways in which chemical molecules are commonly shown, three of which are described in this section.

Figure 3.2 and 3.3 show *Kekulé* structures, where each atom is shown and bonds between atoms are drawn as lines between the atoms. This type of diagram is very useful for molecules that contain only a small number of atoms, and make it easy to visualize the arrangement of atoms within the molecules structure.

A similar type of diagram called a *line-bond formula* or the *skeletal formula* also uses lines to represent bonds, however it generally does not explicitly include carbon or hydrogen atoms. Instead, it is assumed the each bond is connected to a carbon atom unless another atom is specified. Carbon atoms are also assumed to have a total of 4 bonds, with any bonds not explicitly shown being to hydrogen atoms. Hydrogen atoms are typically not shown, unless they are part of a hydroxyl group or they are required for a particular reaction step. Like Kekulé structures, line-bond formulas clearly show the connectivity and arrangement of atoms within a molecule. As an example, the Kekulé structure and line-bond formulas for ethanol are shown in Figure 3.5. The line-bond formula for benzene is shown in Figure 3.6.

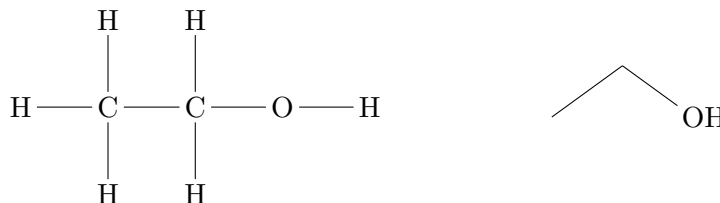


Figure 3.5: The Kekulé and line-bond formulas for ethanol.

When drawing line-bond formulas, benzene rings or phenol groups are typically shown with a circle in the center to show the delocalization of the pi electrons. Figure 3.6 shows the line-bond formula of a benzene ring and an ester. Although these models do not have explicit vertices, they are useful for visualizing large molecules and for visualizing the structure without $C-H$ bonds.

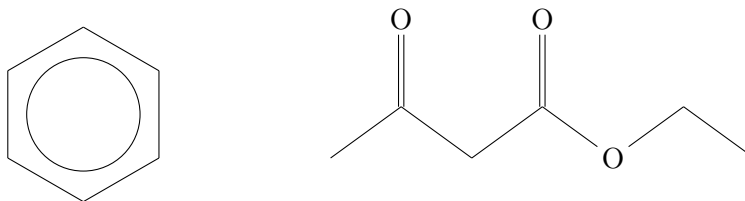


Figure 3.6: Line-bond formulas for benzene and ethyl acetoacetate.

Unlike the previous two methods, *condensed formulas* simply list the type and quantity of atoms. In this type of model, most or all of the bonds in a molecule are omitted and the order in which the atoms are written implies the structure. Parentheses are used to differentiate substituents from the base structure when needed. Ethyl acetoacetate shown in Figure 3.6 has the condensed formula: $CH_3CHOCH_2CHOOCH_2CH_3$. The functional groups in Table 3.1 are shown as condensed formulas.

3.3.1 Corey–Pauling–Koltun (CPK) Colouring

A common way to differentiate atoms in a chemical structure is to use a standard colouring system for specific atoms, namely the Corey–Pauling–Koltun (CPK) colouring system which originated from a paper published by Corey and Pauling in 1952 [5]. The colours for some common atoms are shown in Table 3.2.

Atom	Colour
Carbon	Black
Oxygen	Red
Hydrogen	White
Nitrogen	Blue
Halogens	Green
Metals	Silver

Table 3.2: CPK colourings for some common atoms.

Chapter 4

Graph Isomorphism

This chapter provides an introduction to the graph and subgraph isomorphism problems, as well as several algorithms which can be used to solve these types of problems.

4.1 Graph Isomorphism Problem

Let us consider the graphs G and G' shown in Figure 4.1.

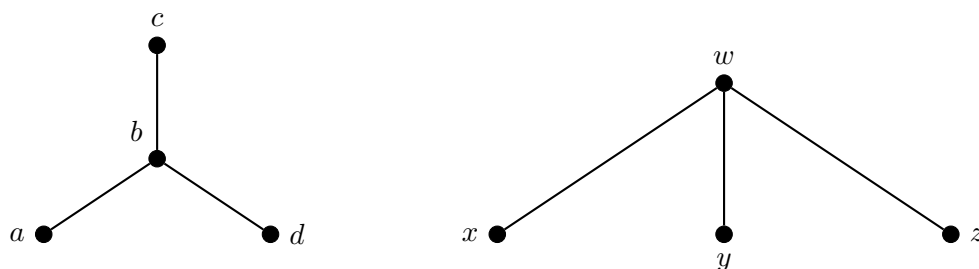


Figure 4.1: A graph G with $V(G) = \{a, b, c, d\}$ and a second graph G' with $V(G') = \{w, x, y, z\}$.

In Section 2.3.1, we observed that we could permute the rows and columns of the adjacency matrix of G to get the adjacency matrix of G' . This was done through matrix multiplication and permutation matrices. Just from looking at the structure of the graphs, it can be observed that there is a mapping ϕ from $V(G)$ to $V(G')$ that preserves adjacency and non-adjacency. That is, ij is an edge in $E(G)$ if and only if $\phi(i)\phi(j)$ is an edge in $E(G')$ where,

$$\phi(a) \rightarrow x$$

$$\phi(b) \rightarrow w$$

$$\phi(c) \rightarrow y$$

$$\phi(d) \rightarrow z.$$

This mapping is an example of an isomorphism between the two graphs. As another example, consider the bipartite graphs from Figure 2.11 (shown below). Visually these seem very different, but when we examine the connectivity between nodes, it can be seen that a_x, b_y are adjacent in G , if and only if they are also adjacent in G' . Therefore, there exists an isomorphism between these two graphs.

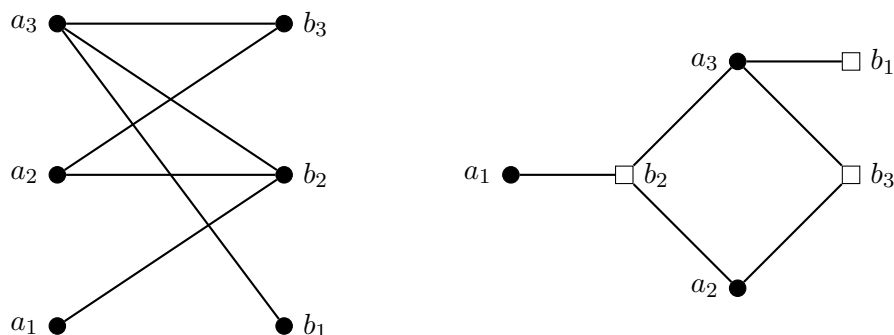


Figure 4.2: Isomorphic bipartite graphs G and G' with partite sets $A = \{a_1, a_2, a_3\}$ and $B = \{b_1, b_2, b_3\}$.

Definition 4.1. Let G and H be graphs. A mapping $\phi : V(G) \rightarrow V(H)$ is an *isomorphism* between G and H if it is a bijection, and preserves the edges and non-edges. That is, for all $u, v \in V(G)$, $uv \in E(G)$ if and only if $\phi(u)\phi(v) \in E(H)$. We say G and H are *isomorphic* if such a mapping exists. For labeled graphs $G_\ell = (\ell, V, E)$, $H_\ell = (\ell', V', E')$ an isomorphism must also preserve labels. That is $\phi(\ell(u)) = \ell'(\phi(u))$.

In the context of a chemical molecule, an isomorphism that preserves labels indicates that the bond connectivity is the same, and so the molecules are identical or stereoisomers of one another.

4.1.1 Similar and Permutation Similar Matrices

Definition 4.2. Let A and B be $n \times n$ matrices. Then A is *similar* to B if there exists an invertible matrix P such that $A = PBP^{-1}$.

Recall from Section 2.3.2 that a permutation matrix is an $n \times n$ matrix with exactly one 1 in each row and each column.

Definition 4.3. If A and B are matrices where there exists a *permutation matrix* P such that $A = PBP^{-1}$ then A is said to be *permutation similar* to B .

Recall that for every permutation matrix P we have $P^{-1} = P^T$. In particular, the inverse of a permutation matrix is a permutation matrix. Hence, if A and B are permutation similar, then we can write $A = PBP^T$ where P is a permutation matrix. An example of this operation is as follows.

Let

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}.$$

The permutation matrix $P = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$ can be applied to show that A and B are permutation similar.

$$PBP^T = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} = A$$

Thus A and B are permutation similar.

Proposition 4.4. *Permutation similarity is an equivalence relation on $n \times n$ matrices.*

Proof. We must show that permutation similarity is reflexive, symmetric and transitive. Since the identity matrix I contains exactly one 1 in each row and each column, it is a permutation matrix. Let A be an $n \times n$ matrix A , and I be the $n \times n$ identity matrix. Since $IAI^{-1} = A$, A is permutation similar to itself, and we conclude that permutation similarity is reflexive.

Let A and B be $n \times n$ matrices, where A is permutation similar to B . Then there exists a permutation matrix P such that

$$PBP^{-1} = A \iff PBP^{-1}P = AP \iff P^{-1}PB = P^{-1}AP \iff B = (P^{-1})A(P^{-1})^{-1}.$$

Thus, if A is permutation similar to B , then B is permutation similar to A , so permutation similarity is symmetric.

Suppose there are $n \times n$ matrices A, B, C where for some permutation matrices P, Q where

$$PAP^{-1} = B \quad \text{and} \quad QBQ^{-1} = C.$$

Then

$$\begin{aligned} C &= QBQ^{-1} \\ &= Q(PAP^{-1})Q^{-1} \\ &= QPAP^{-1}Q^{-1} \\ &= (QP)A(QP)^{-1} \end{aligned}$$

Note that QP is a permutation matrix, as permutation matrices are closed under taking products. Thus A and C are permutation similar, and permutation similarity is transitive. Therefore, we conclude that the relation is an equivalence relation. \square

Observe that the above proof works for general matrices P and Q , thus similarity is also an equivalence relation.

4.1.2 Matrix Representation of an Isomorphism

Let G be a graph on n vertices with adjacency matrix A and G' be a graph on n vertices with adjacency matrix B . Both A and B are $n \times n$ matrices. Suppose both A and B have vertices $\{1, 2, \dots, n\}$. If $\phi : G \rightarrow G'$ is an isomorphism, then the permutation matrix P with $p_{i\phi(i)} = 1$ for each i shows $B = PAP^T$, by Observation 2. Conversely, if A and B are permutation similar, then G and G' are isomorphic. This, along with Observation 2 gives the following additional observation.

Observation 3. *Two graphs G and G' are isomorphic if and only if their adjacency matrices are permutation similar.*

Suppose B and C are symmetric matrices and suppose B is permutation similar to C . Then there is a matrix P such that

$$B = PCP^T = P(PC^T)^T = P(PC)^T.$$

This reformation is used by Ullmann's algorithm for the subgraph isomorphism in Section 4.2. Let G be a graph on n vertices with adjacency matrix A , and G' be a graph on m vertices with adjacency matrix B where $m \leq n$. Let $V(G') = \{j_1, j_2, \dots, j_m\} \subseteq \{1, 2, \dots, n\} = V(G)$. Define S to be the $m \times n$ matrix with row i containing exactly one 1 in column j_i . Then row i of SA is row j_i of A . These are the rows of A corresponding to the subgraph G' . When A is multiplied on the left by the $m \times n$ matrix S , the result will be an $m \times n$ matrix, and $(SA)^T$ will therefore be an $n \times m$ matrix. Multiplying this again by S on the left, the result will be an $m \times m$ matrix, which is the adjacency matrix C of G' . To show a graph G'' with adjacency matrix B is isomorphic to G' we find S such that

$$\begin{aligned} B = PCP^{-1} &= P(S(SA)^T)P^{-1} \\ &= P(SA^T S^T)P^{-1} \\ &= (PS)A^T(S^T P^T) \\ &= (PS)A^T(PS)^T \\ &= (PS)((PS)A)^T. \end{aligned}$$

To recap, if S is a matrix which contains exactly one 1 in each row, and at most one 1 in each column, the result of the operation $S(SA)^T$ is the adjacency matrix of a subgraph of

G . Thus, to identify a subgraph of A which is isomorphic to B , requires an $m \times m$ permutation matrix P and an $m \times n$ matrix S which meets the criteria listed above. That is,

$$C = PS((PS)A)^T$$

or

$$C = M(MA)^T$$

where $M = PS$. Observe M will have exactly one 1 in each row and at most one 1 in each column, as this is a property of S and is preserved by P .

Consider again the graph from Figure 2.1, and a smaller graph H .

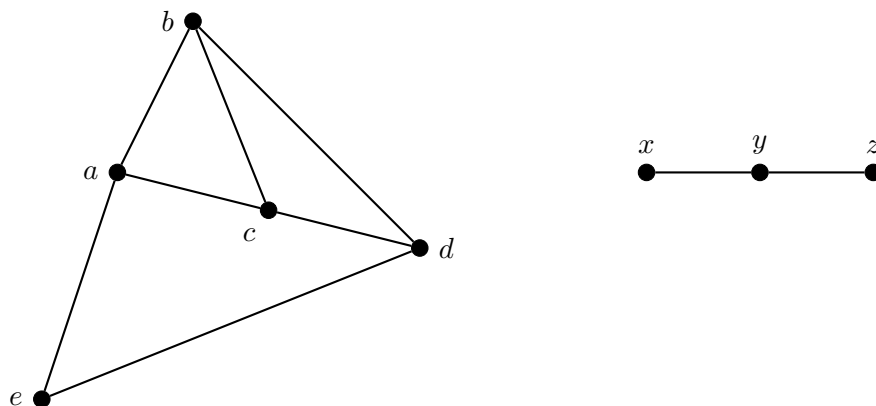


Figure 4.3: Graph G on the left, and graph H on the right.

Then we have

$$A = \begin{matrix} & \begin{matrix} a & b & c & d & e \end{matrix} \\ \begin{matrix} a \\ b \\ c \\ d \\ e \end{matrix} & \begin{bmatrix} 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{bmatrix} \end{matrix}$$

the adjacency matrix of G , and

$$B = \begin{matrix} & \begin{matrix} x & y & z \end{matrix} \\ \begin{matrix} x \\ y \\ z \end{matrix} & \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \end{matrix} \text{ the adjacency matrix of } H.$$

To show that there is an isomorphism between $G[\{a, d, e\}]$ and H , we first multiply A by the 3×5 matrix S , with 1's in the columns corresponding to the vertices a, d and e .

$$SAS^T = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} = C$$

We can see that $C \neq B$, however they are permutation similar as there is a permutation matrix P such that

$$PCP^T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} = B.$$

Thus there is an isomorphism between H and the induced subgraph $G[\{a, d, e\}]$ of G .

If we relax the condition that $B = PCP^T$ to simply whenever $b_{ij} = 1$, then the ij entry of PCP^T is also 1, then we may conclude that H is isomorphic to a subgraph of G , but it may not be induced. Consider

$$S = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}.$$

Then

$$C = SAS^T = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix},$$

while

$$B = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}.$$

Definition 4.5. Let B and C be $m \times m$ $(0,1)$ -matrices. If $b_{ij} \leq c_{ij}$ for all i, j , then we say that C dominates B .

4.2 Ullman's Algorithm

In this section we describe an algorithm for finding all the isomorphisms between a given graph H and subgraphs of another graph G . The algorithm uses the adjacency matrices of H and G to identify possible isomorphisms, and then verify them. Checking all possible mappings from $V(H)$ into $V(G)$ is computationally expensive, so a neighborhood condition is added to reduce the number of mappings and improve the efficiency of the algorithm.

For the remainder of this section, we consider a given graph G with adjacency matrix A and a second graph H with adjacency matrix B where $|V(H)| \leq |V(G)|$. Let $|V(G)| = m$ and $|V(H)| = n$, so A is a $m \times m$ matrix and B is an $n \times n$ matrix.

4.2.1 The Algorithm

The algorithm first computes an $n \times m$ matrix M where the entries of M are determined by comparing the degrees of vertices in G with degrees of vertices in H .

$$m_{ij} = \begin{cases} 1, & \text{if the sum of entries in column } j \text{ in matrix } A \text{ is greater than or equal to the sum} \\ & \text{of entries in column } i \text{ of matrix } B; \\ 0, & \text{otherwise.} \end{cases}$$

Observe that the sum of entries in a row or column of an adjacency matrix is equal to the degree of the corresponding vertex. Therefore $m_{ij} = 1$ means vertex i in H has degree less than or equal to vertex j in G . This is clearly a necessary condition for vertex i to map to vertex j under a subgraph isomorphism.

Symbolically,

$$m_{ij} = \begin{cases} 1 & \text{if } \deg(j) \geq \deg(i) \text{ for } j \in V(G), i \in V(H) \\ 0 & \text{otherwise} \end{cases}$$

The matrix M may have several 1's in each row. Taking the matrix M the algorithm considers every matrix $M_k, k = 1, 2, 3, \dots$ generated from M by choosing exactly one 1 in each row so that there is at most one 1 in each column. The matrix

$$C = M_k(M_k A)^T$$

is the adjacency matrix of a subgraph G' of G . If C dominates B , then B is isomorphic to a subgraph of G' and the matrix M_k describes this isomorphism.

As an example, let G and H be the following graphs.

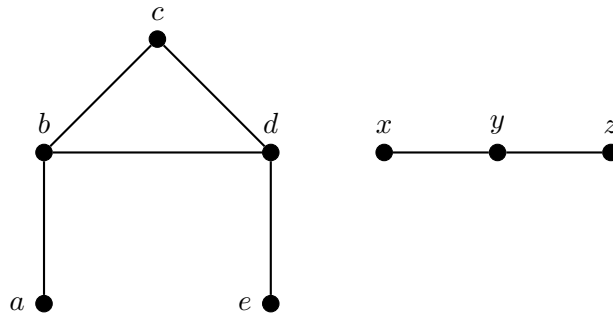


Figure 4.4: Graphs G and H .

The adjacency matrices of G and H are

$$A = \begin{matrix} & a & b & c & d & e \\ \begin{matrix} a \\ b \\ c \\ d \\ e \end{matrix} & \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \end{matrix} \quad \text{and} \quad B = \begin{matrix} & x & y & z \\ \begin{matrix} x \\ y \\ z \end{matrix} & \begin{bmatrix} 0 & 1 & z \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \end{matrix}.$$

The corresponding matrix M is

$$M = \begin{matrix} & a & b & c & d & e \\ \begin{matrix} x \\ y \\ z \end{matrix} & \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \end{matrix}.$$

We see that $\deg(y) = 2$ and that $\deg(a) = \deg(e) = 1$ therefore the corresponding entries in M are 0. There are $3 \times 4 \times 3 = 36$ choices for M_k , two of which are shown below.

$$M_1 = \begin{matrix} & a & b & c & d & e \\ \begin{matrix} x \\ y \\ z \end{matrix} & \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix} \end{matrix}$$

Giving

$$C = M_1(M_1A)^T = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Which maps $z \rightarrow a$, $y \rightarrow d$ and $x \rightarrow c$, however, since a and d are not adjacent, but y and z are, $\{a, c, d\}$ is not a subgraph isomorphic to H . This can be seen as B dominates C , specifically because $c_{32} = 0$ but $b_{32} = 1$.

$$M_2 = \begin{matrix} & a & b & c & d & e \\ \begin{matrix} x \\ y \\ z \end{matrix} & \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix} \end{matrix}$$

Working through this example in more detail, we get

$$D = M_2(M_2A)^T = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix} \left(\begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \right)^T = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

The matrix M_2 is like M_1 except that it maps $z \rightarrow b$. Here we observe that if $b_{ij}=1$, then $d_{ij} = 1$. In this case, adjacency is preserved and H is isomorphic to a subgraph of G on $\{b, c, d\}$. Note H is not isomorphic to $G[\{b, c, d\}]$ as $b_{13} = 0$ but $d_{13} = 1$.

4.2.2 Labeling Condition

A labeling condition can be added to reduce the number of 1's in the matrix M and therefore reduce the number of matrices M_k to check for the subgraph isomorphism described above. The application of a labeling condition is natural for the case of chemical graphs which are comprised of different types of atoms. When searching for copies of a particular functional group it is important to take into consideration the type of molecule.

4.2.3 Neighbourhood Condition

Another method of reducing the number of 1's in the matrix M is to add a neighborhood condition. The idea behind this condition is to impose the restriction that a vertex in H can only be matched to a vertex in G if their neighbours are the same. This can be done both in the context of labeling, and in the context of comparing vertex degrees.

Let β be a vertex in $V(H)$ where $\beta_1, \beta_2, \dots, \beta_s$ are the vertices adjacent to β , and let α be a vertex in $V(G)$ where $\alpha_1, \alpha_2, \dots, \alpha_t$ are the vertices adjacent to α in $V(G)$. If there exists an isomorphism where β maps to α , then for each $x = 1, 2, \dots, s$ there must be a vertex α_y that is adjacent to α such that β_x maps to α_y .

4.2.4 Example

Consider the following graph G representing a simplified sugar molecule, and the smaller graph H on 3 vertices. The adjacency matrix B which corresponds to H is shown. The adjacency matrix A which corresponds to G is an 18×18 matrix, and is not shown. We show how using a labelling conditions speeds up the search for the functional group H in G .

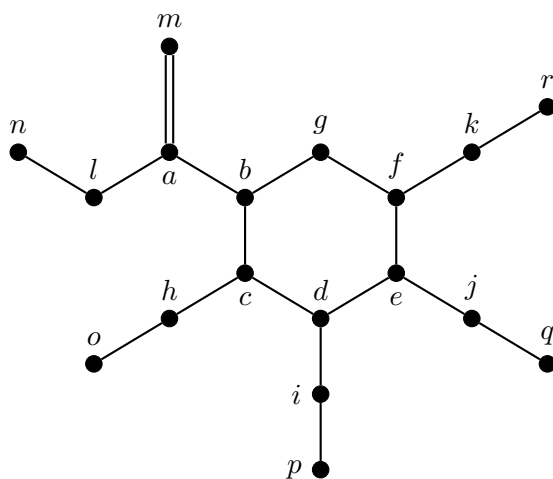


Figure 4.5: Graph G corresponding to a molecule of D-Galaturonic acid.

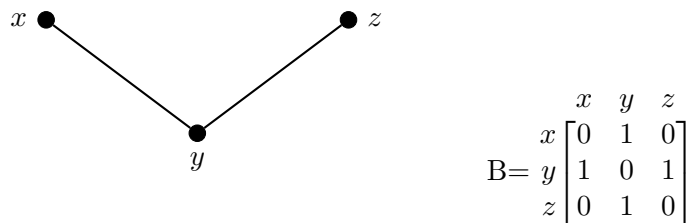


Figure 4.6: The graph H corresponding to a hydroxyl functional group, and its adjacency matrix B .

The Matrix M

We begin by defining the matrix M using the conditions set by Ullman's algorithm. Since $\deg(y) = 2$, the zero entries in the second row of the matrix M correspond to those vertices in G with degree less than 2. From this matrix we can see that there are $13 \times 17 \times 16 = 3536$ ways to choose exactly one 1 from each row, each from a distinct column, and therefore there would be many possible isomorphic mappings to be checked.

$$M = \begin{matrix} & \begin{matrix} a & b & c & d & e & f & g & h & i & j & k & l & m & n & o & p & q & r \end{matrix} \\ \begin{matrix} x \\ y \\ z \end{matrix} & \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \end{matrix}$$

Applying the Labeling Condition

Since G represents a sugar molecule, and H represents a hydroxyl group, the vertices can be labeled according to the type of atom they represent. Both G and H use the set of labels $\{C, O, H\}$. For this example, the subscripts are only shown for ease of reference and identification. Vertices labeled with the same letter are considered to have the same label regardless of subscript.

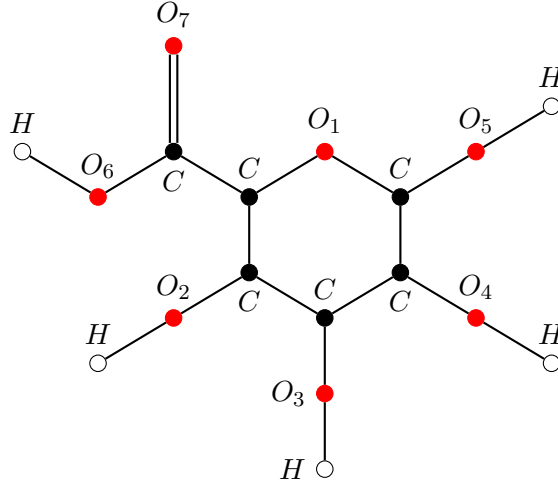


Figure 4.7: Graph G corresponding to a molecule of D-Galaturonic acid with labels assigned based on type of atom.

When H is labeled to represent a hydroxy group attached to a carbon atom, we get the following graph and adjacency matrix B_ℓ

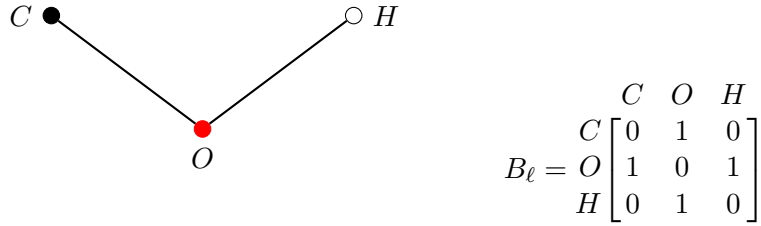


Figure 4.8: The graph H corresponding to a hydroxyl functional group, and its adjacency matrix B .

The labeling condition requires that $m_{ij} = 1$ only if $\ell(i) = \ell(j)$. The resulting matrix M_ℓ by adding the labeling condition to the original matrix M is shown below.

$$\begin{matrix} & \begin{matrix} C_1 & C_2 & C_3 & C_4 & C_5 & C_6 & O_1 & O_2 & O_3 & O_4 & O_5 & O_6 & O_7 & H_1 & H_2 & H_3 & H_4 & H_5 \end{matrix} \\ \begin{matrix} C \\ O \\ H \end{matrix} & \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \end{matrix}$$

We can see that in M_ℓ , the only non-zero entries occur when the column and row labels indicate the same type of atom. Adding this condition allows us to reduce the number ways to choose exactly one 1 in each row, and therefore reduce the number of possible mappings to $6 \times 7 \times 5 = 210$.

Applying the Neighbourhood Condition

The neighborhood condition allows us to take the labelling condition one step further. If we take the matrix M_ℓ from the previous step, we can see from examining Figure 4.7 that the columns corresponding to hydrogen atoms and carbons will not be affected by this new condition. This is because, the only condition for both hydrogen atoms and carbon atoms is that they each be adjacent to an oxygen atom. However, if we examine the vertices labeled as oxygens, we can see that not all oxygen atoms in G are adjacent to a carbon and a hydrogen as in H . It is therefore possible to change the second row entries for O_1 and O_7 from 1 to 0. There are now $6 \times 5 \times 5 = 150$ mappings to check.

$$\begin{array}{c} C \\ O \\ H \end{array} \begin{array}{ccccccccccccccccccccc} C_1 & C_2 & C_3 & C_4 & C_5 & C_6 & O_1 & O_2 & O_3 & O_4 & O_5 & O_6 & O_7 & H_1 & H_2 & H_3 & H_4 & H_5 \\ \left[\begin{array}{cccccccccccccccccccc} 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{array} \right] \end{array}$$

4.3 The Gemini and SubGemini Algorithms

The SubGemini Algorithm [10] was created to determine whether a larger circuit contained copies of a smaller circuit. For this algorithm, circuits were represented as graphs, where the vertices represented devices and nets, which were connected by edges. Devices and nets were labeled based on type as well as degree. The algorithm is based on the Gemini and GeminiII algorithms [7] which were both designed to find isomorphisms between graphs of the same order. This was done through partitioning the vertices into sets, following the principle outline by [6]. The labeling associated with chemical graphs make them very well suited to this particular algorithm.

4.3.1 Gemini and GeminiII

The Gemini and GeminiII algorithms can simply begin by partitioning the vertices based on the original label of each vertex. Recall that *partitioning* a set involves grouping elements of the set into non-empty subsets where each element of the set appears in exactly one of the subsets.

The Gemini algorithm, which was developed for use on electric circuits represented by bipartite graphs, starts by partitioning vertices into partite sets. Chemical graphs are rarely bipartite, however, the natural first step is to partition vertices based on the type of atom they represent.

Next, each vertex is re-labeled to include information about its neighbours. For a current label L_o , we obtain a new label L^+ , where $+$ denotes concatenation, by setting

$$L^+ = L_o + \sum c_i L_i \tag{4.1}$$

where L_i is the original label of the neighbour node, and c_i is the number of bonds, or edges, between the nodes. For example, the oxygen atom in water (Figure 2.5) would have the label $O1H1H$, while a carbon double bonded to another carbon and attached to two hydrogen atoms would have the label $C2C1H1H$. The labeling conventions could follow the IUPAC standard for priority of atoms and functional groups.

This relabeling procedure can then be applied to neighbours of neighbours until each partition contains only a single element. From there, the algorithm would look for an isomorphic mapping between the partitions of one graph and the partitions of the other. If such a mapping exists, then the two graphs are isomorphic.

There are some exceptional cases where two vertices may remain identical through this relabeling process. Some examples are shown below. In these cases, molecules possess at least one plane of symmetry.

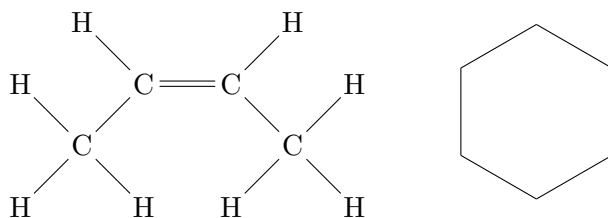


Figure 4.9: The structure of cis-2-butene on the left, and a benzene ring on the right.

Example 4.6. Benzene and cis-2-butene (Figure 4.9) are examples of molecules that contain planes of symmetry. Applying the partitioning algorithm to cis-2-butene would not be able to differentiate between the two central carbons and the two terminal carbons. In the benzene ring, all the carbon atoms would receive the same label and thus be indistinguishable for the algorithm.

Definition 4.7. *H-NMR* or *proton nuclear magnetic resonance* is a form of spectroscopy which can be used to determine the structure of a molecule. The results of H NMR show peaks, whose position and number indicate the arrangement of hydrogen atoms. Hydrogen atoms in the same partition after applying the labeling procedure described above, show a peak at the same position.

If we take the structure of cis-2-butene and consider only the hydrogen atoms, we can see that each of the hydrogen atoms shown in blue would receive the same label, as would the two shown in red. For example, the red hydrogen atoms would receive the label $H1C2C1C...$ while the blue hydrogen atoms would receive the label $H1C1C1H1H...$

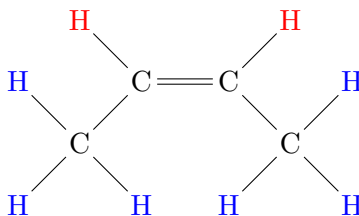


Figure 4.10: The partitioning of hydrogen atoms in a molecule of cis-2-butene.

It is also possible to determine the structure of a molecule from one of the labels. Choosing one of the red hydrogen atoms, and applying the labeling algorithm, the label would be created according to the following sequence.

$H, H1C, H1C2C1C, H1C2C1C1C1H1H1H1H, H1C2C1C1C1H1H1H1H1H1H1H$

Given a label, we can reconstruct a graph using information about the types of atoms. We begin with a hydrogen which can form only one bond, in this case to a carbon. The carbon atom forms 4 bonds, which in this case includes a double bond to another carbon. The full process for this reconstruction is shown below.

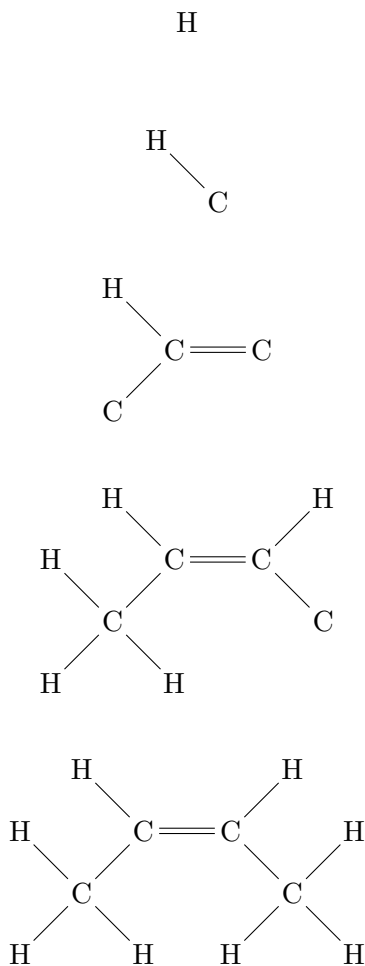


Figure 4.11: Determining the structure of cis-2-butene from one of the labels.

Vertices with identical labels are in the same orbit under the automorphism group of a graph G . More information about group actions on a set, and orbits can be found in the text by Bondy and Murty [2].

In the case of cis-2-butene, there are two distinct partite sets containing hydrogen atoms, and so we would expect to see two distinct peaks on an H-NMR spectrum for cis-2-butene, as is shown in Figure 4.12.

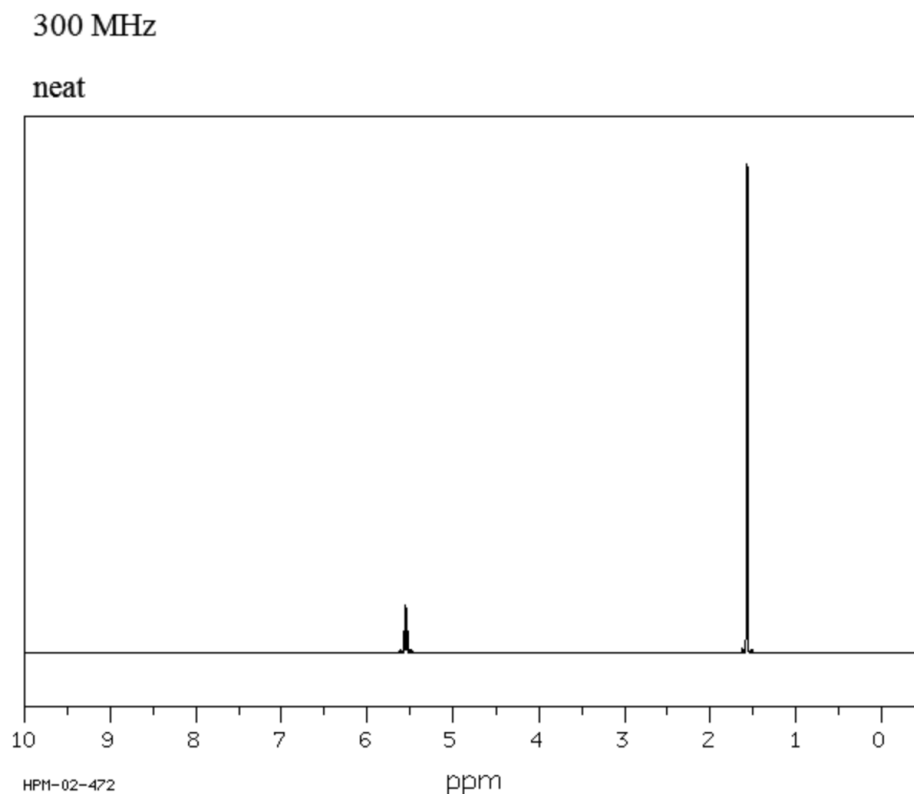


Figure 4.12: An H-NMR of 2-cis-butene showing two distinct peaks [4].

The first peak indicates the two red hydrogen atoms, and the taller peak indicates the six blue hydrogen atoms. The splitting of the peaks indicates the number of nearby, non-equivalent hydrogen atoms and gives further information about the structure. Most chemical molecules will have more than one equivalent hydrogen meaning they will always have the same labels. For this reason, the Gemini algorithm would require a condition that hydrogen atoms do not need to be in singleton partitions, or would require equivalent hydrogen atoms to be omitted altogether.

Example: Propene Alcohol Isomers

In chemistry, the term *isomer* is used to refer to two molecules with the same number of each type of atom, but with different bond connectivity. There are different types of isomers such as *stereoisomers* and *structural isomers* with different characteristics and properties. An example of a pair of isomers is propen-2-ol shown in Figure 4.13 and 2-propen-1-ol shown in Figure 4.14. Both of these species contain 1 oxygen, 3 carbon and 6 hydrogen atoms. We will use the Gemini algorithm to show that these two structural isomers are not isomorphic.

For this example, each labeling step will use the order O, C, H and in the case of two

atoms with the same initial label, the atoms connected by multiple bonds will be listed first.

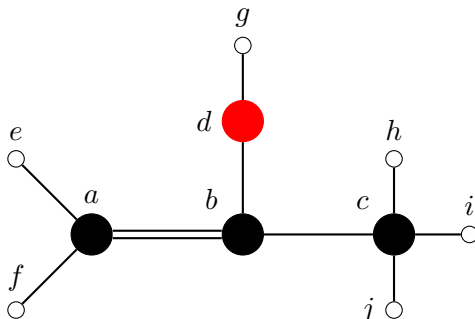


Figure 4.13: A graph representation of propen-2-ol using CPK colouring.

We begin by labeling each vertex from the set $\{C, O, H\}$ based on the atom type.

$$C = \{a, b, c\}$$

$$O = \{d\}$$

$$H = \{e, f, g, h, i, j\}$$

The partition of vertices with the label O contains only one vertex, so it does not require further labeling. Furthermore, we will omit the hydrogen atoms for the next labeling step, as both vertex a and vertex c clearly show equivalent hydrogen atoms. Next, we re-label the remaining carbons based on those atoms immediately adjacent.

$$C2C1H1H = \{a\}$$

$$C1O2C1C = \{b\}$$

$$C1C1H1H1H = \{c\}$$

Since each vertex has a unique label, we stop.

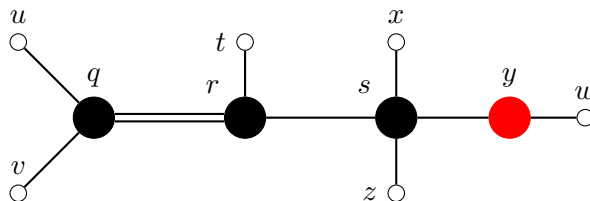


Figure 4.14: A graph representation of 2-propen-1-ol shown using CPK colouring.

As we did with the first molecule, we begin by partitioning based on the type of atom.

$$O = \{y\}$$

$$C = \{q, r, s\}$$

$$H = \{t, u, v, w, x, z\}$$

Once again the oxygen results in a singleton partition. Omitting the hydrogen, we proceed with the labeling process.

$$C2C1H1H = \{q\}$$

$$C2C1C1H = \{r\}$$

$$C1O1C2H = \{s\}$$

This result shows that $\{d\}$ could be mapped to $\{y\}$ and $\{a\}$ could be mapped to $\{q\}$, however the remaining vertices have different labels and so have different bond connectivity. Therefore, we conclude that the two isomers are not isomorphic.

4.3.2 SubGemini

Although the Gemini algorithm can identify isomorphisms between two graphs with the same number of vertices, the method of identifying an isomorphism between partite sets containing single vertices does not work for graphs of different order. The goal of the SubGemini algorithm is therefore to identify a *key vertex* in the smaller graph and use that to create a *candidate vector* of all possible vertices in the larger graph that it could map to. To visualize this, consider a large organic molecule containing mostly carbon and hydrogen atoms, with a small number of oxygen and other atoms as well. If the pattern of interest contained an oxygen molecule, the SubGemini algorithm would identify the oxygen as the key vertex, and then generate a vector of all oxygen molecules with the same adjacencies within the larger molecule. This approach would be more effective than checking every possible mapping that included a carbon or a hydrogen atom which are much more numerous.

The process of identifying the key vertex uses the same basic principle as the Gemini algorithm, but with the important consideration that there will be one or more vertices in the smaller graph that will be mapped to a vertex in the larger graph with more edges. More specifically, let G and H be connected graphs such that $|V(G)| \geq |V(H)|$ and suppose that an isomorphism exists between H and the subgraph G' of G . Then there is at least one vertex in G' that is connected to vertices in G that are not in G' . Subgemini accounts for this type of vertex by marking it as “corrupted” meaning that it does not undergo further labeling. These corrupted vertices can not be used as the key vertex, and if the next iteration of the labeling process for a particular vertex v would include a corrupted vertex, the labeling process terminates for v . At this point, the label for vertex v is complete.

The outline of the algorithm described by Ohrich, Ebeling, Ginting and Sather [10] is as follows.

- Use the labeling procedure to partition the vertices in H until all labels are complete and the parts contain only a single vertex.
- Label the vertices in G using the same number of iterations used to label the vertices in H .
- Choose the partition of G with the fewest elements to be the *candidate vector*. The vertex in H with the corresponding label is the *key vertex* K .
- If there is more than one partition of $V(G)$ with the smallest number of elements, K can be selected arbitrarily.

- Check for isomorphisms between H and subgraphs of G containing vertices in the candidate vector.

In some cases, identifying the key vertex in the smaller graph may be obvious just from the initial label; however, the labeling process is still valuable as it can further reduce the size of the candidate vector based on those vertices adjacent to it as is shown the following example.

Example - Ethyl Acetoacetate

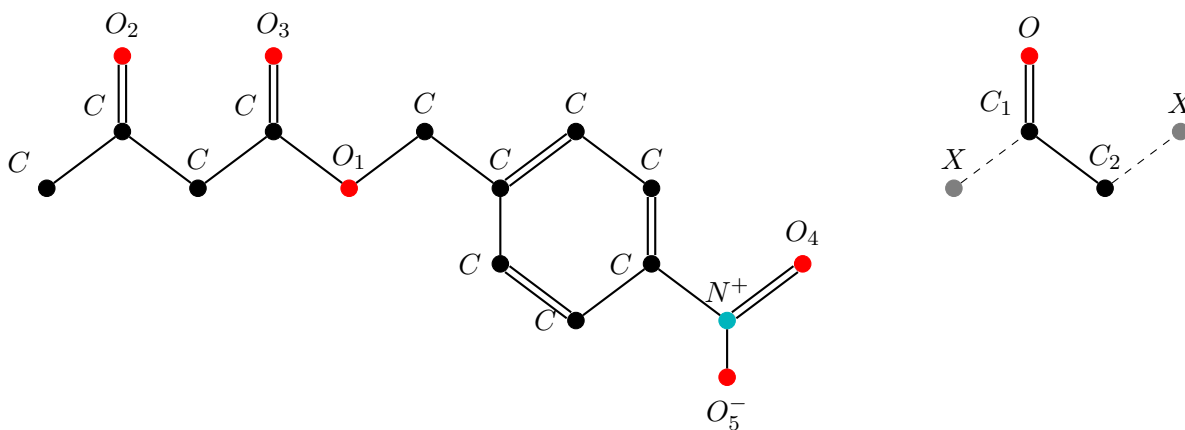


Figure 4.15: 4-Nitrobenzyl Acetoacetate and the carbonyl pattern. The dashed lines and vertices labeled X are corrupted vertices. The subscripts are shown for ease of identification.

Key Vertex

To begin, the labeling process is applied to the carbonyl pattern using initial labels only, giving $\{O\}$ and $\{C_1, C_2\}$. For the next step, the oxygen would receive the label $O2C$, however both carbons are corrupted at this step and so do not receive labels. Although it was possible to identify oxygen as the key vertex based solely on the initial labels, it is important to proceed with the labeling to reduce the number of vertices in the subsequent candidate vector.

Candidate Vector

Since the vertices in the candidate vector must have labels that begin with O , we will only consider the vertices in 4-nitrobenzyl acetoacetate which correspond to oxygen atoms. After the initial partitioning we have the partition.

$$\{O_1, O_2, O_3, O_4, O_5\}$$

Had we not fully labeled the key vertex, we would need to check for isomorphisms involving all of these oxygen atoms, however the additional labeling step allows us to eliminate some of

these vertices. The next labeling iteration give us the following partitions.

$$O2N = \{O_4\} \quad O1N = \{O_5\} \quad O1C1C = \{O_1\} \quad O2C = \{O_2, O_3\}$$

So our final candidate vector, $\{O_2, O_3\}$ contains only 2 vertices as opposed to 5. From this it can be determined that there are multiple subgraph isomorphisms between 4-nitrobenzyl acetoacetate and the carbonyl pattern.

Chapter 5

Reaction Networks

Reactions networks are typically modeled in one of two ways. In a chemical reaction network, or CRN, the vertices represent the chemical species, and the edges represent the reactions occurring between them [14]. In a reaction route network, or RRN, a system is modeled as a bipartite digraph, where chemical species and chemical reactions are modeled as partite sets of nodes. In this model, species nodes are only adjacent to reaction nodes and vice versa.

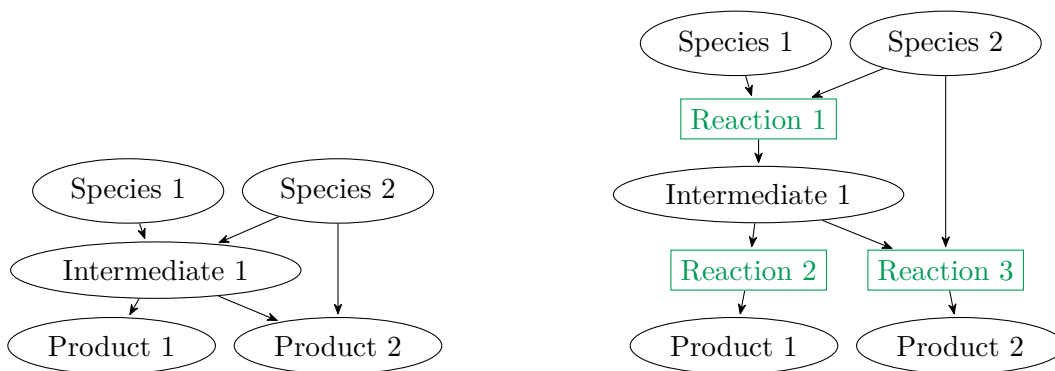


Figure 5.1: The same simple chemical system modeled as a CRN and as an RRN.

There are benefits to both models, depending on the specifics of the reaction and system being modeled.

5.1 Orlova's Algorithm

In this section we present the algorithm from Orlova's thesis [11] which also appears in the paper [12]. The algorithm overview is as follows.

Algorithm 1: Algorithm

input: list of initial chemical species and complete list of reaction rules/transformations

```
1 update = true;
2 while update=true do
3   update=false;
4   for  $1 \leq i \leq \# \text{ of reaction rules}$  do
5     Find all candidates to undergo reaction  $i$  from the current list of chemical
       species
6     Apply the transformation to create products
7   end
8   if A product is not in the list of chemical species then
9     Add the product or products to the list
10    update=true;
11  end
12 end
```

5.1.1 Algorithm Components

Line 5: To find candidates for a particular reaction, the algorithm searches the current list of chemical species to determine which species have the requisite pattern or patterns. This is done by employing Ullmann’s algorithm to identify isomorphisms between the required pattern and subgraphs of the molecular graphs on specific vertices.

Line 8: To verify if any of the species generated by the previous step have already been included in the list of chemical species the algorithm checks for graph isomorphism between the new and existing chemical molecules.

Line 6: Pre-determined reaction rules and transformations are applied to reaction candidates by replacing or altering portions of the reacting species. An example of such a transformation is shown below.

5.2 Reaction Network for Ethyl Acetoacetate

This section details an example of a reaction network constructed using the algorithm in Section 5.1. A complete list of the patterns and reaction rules is shown, as well as the reaction network that is generated.

5.2.1 Input: Initial Species

For this example, we start with only two chemical species, ethyl acetoacetate and methanol. The structure of these two molecules is shown in Figure 5.2.

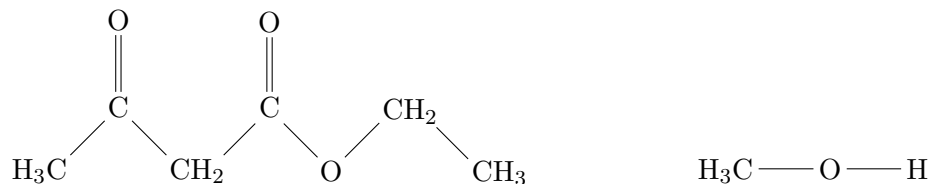
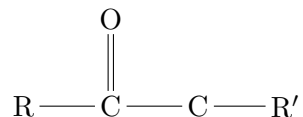


Figure 5.2: The structure of ethyl acetoacetate and methanol.

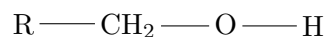
5.2.2 Input: Patterns

In order to create the list of reaction rules, we must identify the patterns capable of undergoing reactions in this system. Some molecules, like the initial species ethyl acetoacetate, may have more than one pattern and so be a candidate species for more than one reaction.

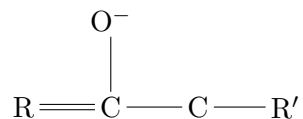
Pattern 1. Ketone - Keto Form



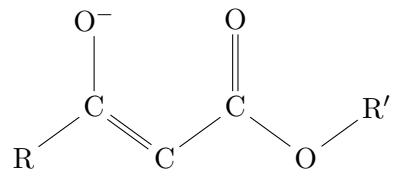
Pattern 2. Alcohol



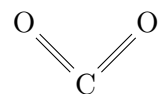
Pattern 3. Ketone - Enol Form



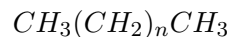
Pattern 4. Acetoacetate Enolate

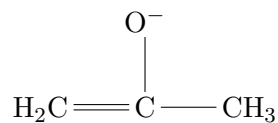
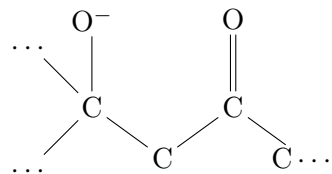
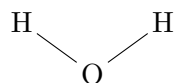
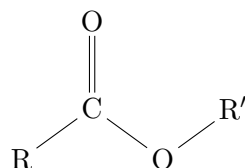
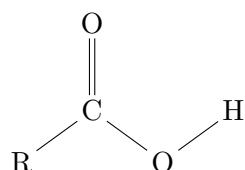


Pattern 5. Carbon Dioxide

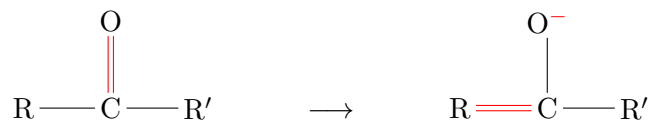


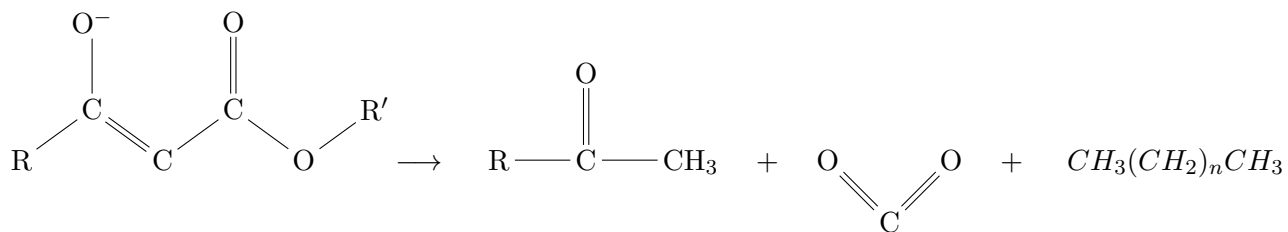
Pattern 6. Carbon Chain



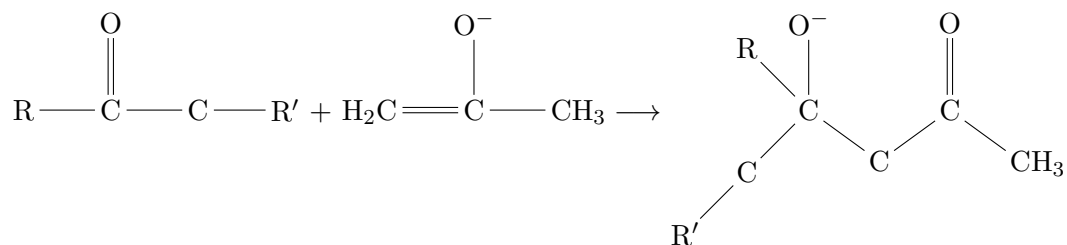
Pattern 7. Acetone Enolate**Pattern 8.** Diacetone Alcohol**Pattern 9.** Water**Pattern 10.** Ester**Pattern 11.** Carboxylic Acid**5.2.3 Input: Reaction Rules**

From the patterns we generate a list of reaction rules. The Reaction number and the patterns involved in each reaction are summarized in Table 5.1. Colours are used in each reaction to show the parts of each pattern that are undergoing transformation.

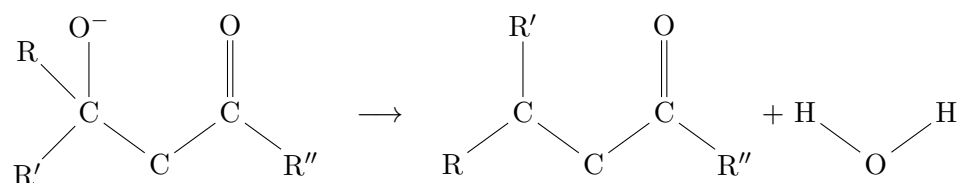
Reaction 1. Keto-Enol Tautomerization**Reaction 2.** Decarboxylation



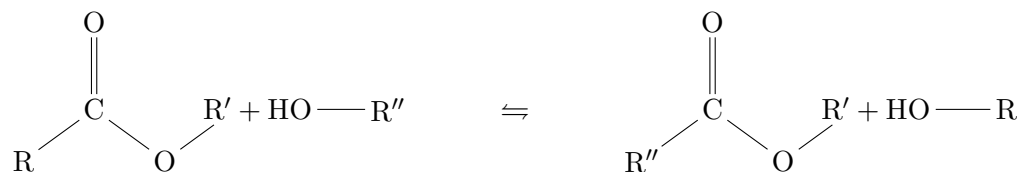
Reaction 3. Aldol Addition



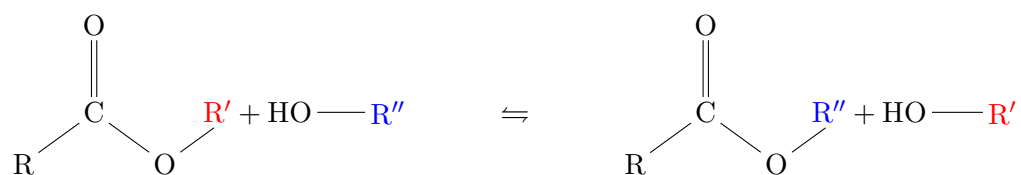
Reaction 4. Aldol Condensation



Reaction 5. Alcoholysis



Reaction 6. Ester Exchange (Transesterification)

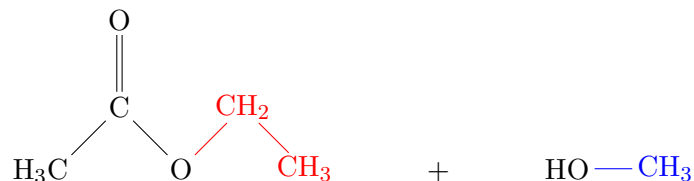


Reaction	Reactants	Products
1	Pattern 1	Pattern 3
2	Pattern 4	Pattern 1 + Pattern 5 + Pattern 6
3	Pattern 1 + Pattern 7	Pattern 8
4	Pattern 8	Pattern 1 + Pattern 9
5	Pattern 3 + Pattern 2	Pattern 1
6	Pattern 10 + Pattern 2	Pattern 2 + Pattern 10

Table 5.1: Summary of Reactions.

Example 5.1. The ester exchange transformation between ethyl acetate and methanol, and the resulting products.

One of the reactions that occurs as a part of this chemical network is an ester exchange between ethyl acetate and methanol, shown below. In Line 2 of the algorithm, Ullman's algorithm would identify pattern 2 in methanol and pattern 10 in ethyl acetate. Both species are shown below.



The R groups of each molecule are highlighted in their corresponding adjacency matrices shown below. The adjacency matrix for methanol has the R group highlighted in blue and the atom where it attaches to the pattern is highlighted in green.

$$\begin{array}{c} C \quad C \quad C \quad C \quad O \quad O \\ C \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 2 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

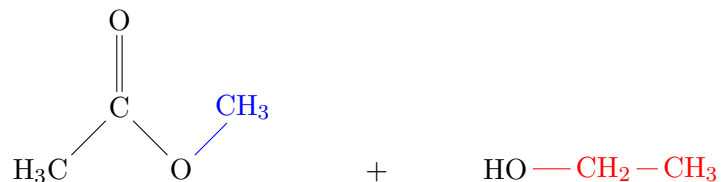
$$\begin{array}{c} C \quad O \quad H \\ C \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \\ O \\ H \end{array}$$

In order to more easily visualize the transformation for ethyl acetate, we can permute the rows and columns of the adjacency matrix to move the submatrix corresponding to the R group to the top left hand corner of the matrix. Note that the only other non zero entry in the first two rows and columns is the atom where that group connects to the rest of the molecule. This entry is highlighted in green.

$$\begin{array}{c} C \quad C \quad C \quad C \quad O \quad O \\ C \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 2 & 1 \\ 0 & 0 & 0 & 2 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \end{array}$$

$$\begin{array}{c} C \quad O \quad H \\ C \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \\ O \\ H \end{array}$$

After the candidate species have been identified, the transformation is applied. From the list of reaction rules, we know that the reaction results in the R groups of the two molecules being exchanged, resulting in the products shown below.



	C	C	C	O	O
C	0	0	0	0	1
C	0	0	1	0	0
C	0	1	0	2	1
O	0	0	2	0	0
O	1	0	1	0	0

	C	C	O	H
C	0	1	1	0
C	1	0	0	0
O	1	0	0	1
H	0	0	1	0

5.2.4 Network

This reaction network is bipartite, with the chemical species shown in round nodes, while the reaction nodes are shown in green boxes. The input species, ethyl acetoacetate and methanol, are shown in purple, while the final nonreactive products are shown in red. The intermediate species, which are those chemical species that are generated by the algorithm but also having patterns which allow them to react further, are those species shown in the black circles.

A few reactions have been omitted from this diagram because their products are already a part of the network. For instance, methyl acetoacetate is a candidate for Reaction 6 with methanol, however the products of that reaction would be methyl acetoacetate and methanol which are already in the network.

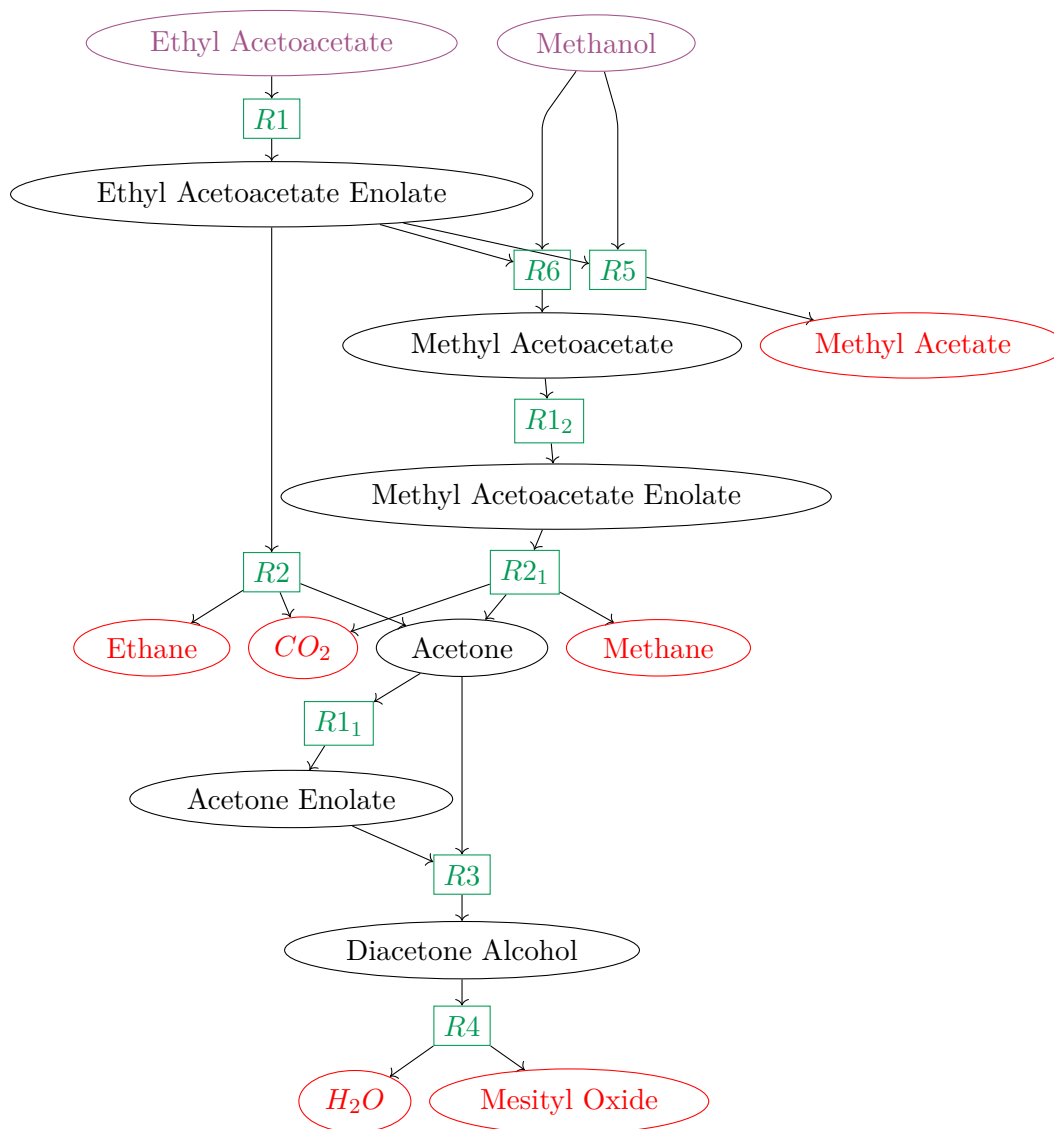


Figure 5.3: The reaction network generated by Orlova's Algorithm generated from ethyl acetoacetate and methanol.

5.2.5 Output: List of Chemical Species

Intermediate Species

Species Name	Patterns	Condensed Formula
Ethyl Acetoacetate Enolate	Pattern 3, Pattern 4	$CH_3C(OH) = CHCOOC_2H_5$
Methyl Acetoacetate	Pattern 1	$CH_3COCH_2COOCH_3$
Methyl Acetoacetate Enolate	Pattern 7, Pattern 4	$CH_3COCH = C(O^-)CH_3$
Acetone	Pattern 1	$(CH_3)_2CO$
Acetone Enolate	Pattern 7	$CH_2^-C(O)CH_3$
Diacetone Alcohol	Pattern 8	$CH_3C(O)CH_2C(OH)(CH_3)_2$

Table 5.2: The reaction intermediates from the reaction network for ethyl acetoacetate and methanol.

Non-Reactive Final Products

Species Name	Patterns	Condensed Formula
Methyl Acetate*	Pattern 1	CH_3COOCH_3
CO_2	Pattern 5	CO_2
Ethane	Pattern 6	CH_3CH_3
Methane	Pattern 6	CH_4
H_2O	Pattern 9	H_2O
Mesityl Oxide * *	Pattern 1	$(CH_3)_2C = CHCOCH_3$

Table 5.3: Final products of the reaction network for ethyl acetoacetate and methanol.

*Methyl acetate can undergo Reaction 6 with methanol, however the products of that reaction will be the same as the inputs.

** Mesityl Oxide is a candidate for Reaction 1, however, its enol form does not react further in this network.

Chapter 6

Conclusion

Chemical graph theory is a field that requires an understanding of both graph theory and chemistry. Questions that are studied in one area, often have parallels to topics of interest in the other. An example of such a parallel is the subgraph isomorphism problem discussed in this paper. The subgraph isomorphism problem is directly related to chemistry, on account of the importance of functional groups in chemical reactions. Some features of chemical graphs, such as the natural partitioning of vertices into groupings based on atom type, make the algorithms used in graph theory particularly effective when applied to chemical graphs. The algorithms described in this paper focus primarily on the structural characteristics of chemical species, however there are applications of these concepts to other aspects of chemistry including reaction rates and kinetics.

There may also be improvements to consider for the reaction network algorithm described by Orlova [11]. For instance, the current algorithm does not track which reactions have already been considered, and so each time a new chemical species is added to the list, each reaction rule will be applied to all species with the requisite patterns, even those that have already been checked. It would be possible to generate a list of species that have already had specific transformations applied so that when a new species is added, only those species that have not reacted with one another will be considered. It is possible, however, that the size of most reaction networks mean that improvements like this do not significantly improve time or efficiency of the algorithm.

Graph theory provides many useful tools and techniques which can be used to effectively model molecular structure and chemical reactions. A few of these techniques were discussed in this paper, however there are many more applications to many different areas of chemistry. The algorithms described in this paper focus primarily on the structural characteristics of chemical species, however there are also applications of these concepts to areas such as reaction rates and reaction kinetics, and the physical and chemical properties of specific molecules.

Bibliography

- [1] N. Biggs, E.K. Lloyd, and R.J. Wilson. *Graph Theory, 1736-1936*. Clarendon Press, 1986. ISBN: 9780198539162. URL: <https://books.google.ca/books?id=XqYTkOsXmpoC>.
- [2] J. A. Bondy and U. S. R. Murty. *Graph Theory With Applications*. Elsevier Science Publishing Co., 1982.
- [3] Arthur Cayley. “On the mathematical theory of isomers”. In: *The Collected Mathematical Papers*. Cambridge Library Collection - Mathematics. Cambridge University Press, 2009, pp. 202–204.
- [4] *2-BUTENE(624-64-6) 1H NMR*. Chemical Book, CAS DataBase List. URL: https://www.chemicalbook.com/SpectrumEN_590-18-1_1HNMR.htm.
- [5] Robert B. Corey and Linus Pauling. “Molecular Models of Amino Acids, Peptides, and Proteins”. In: *Review of Scientific Instruments* 24.8 (Aug. 1953), pp. 621–627. ISSN: 0034-6748. DOI: 10.1063/1.1770803. eprint: https://pubs.aip.org/aip/rsi/article-pdf/24/8/621/19099249/621\1\1_online.pdf. URL: <https://doi.org/10.1063/1.1770803>.
- [6] D.G. Corneil and C.C. Gotlieb. “An Efficient Algorithm for Graph Isomorphism”. In: *Journal of the Association for Computing Machinery* (1970).
- [7] C. Ebeling. “GeminiII: a second generation layout validation program”. In: *[1988] IEEE International Conference on Computer-Aided Design (ICCAD-89) Digest of Technical Papers*. 1988, pp. 322–325. DOI: 10.1109/ICCAD.1988.122520.
- [8] Ivan Gutman et al. “Why plerograms are not used in chemical graph theory? The case of terminal-Wiener index”. In: *Chemical Physics Letters* 568-569 (2013), pp. 195–197. ISSN: 0009-2614. DOI: <https://doi.org/10.1016/j.cplett.2013.03.045>. URL: <https://www.sciencedirect.com/science/article/pii/S0009261413003771>.
- [9] Compiled by Alan D. McNaught IUPAC and Andrew Wilkinson. *Compendium of Chemical Terminology: IUPAC Recommendations 2nd ed*. Blackwell Scientific Publications, 2019.

- [10] Miles Ohlrich et al. “SubGemini: identifying subcircuits using a fast subgraph isomorphism algorithm”. In: *Proceedings of the 30th International Design Automation Conference*. DAC '93. Dallas, Texas, USA: Association for Computing Machinery, 1993, pp. 31–37. ISBN: 0897915771. DOI: 10.1145/157485.164556. URL: <https://doi.org/10.1145/157485.164556>.
- [11] Yuliia Orlova. “Graph-theoretical approach to algorithmic construction of complex reaction networks”. PhD thesis. Sept. 2020.
- [12] Yuliia Orlova, Ivan Kryven, and Piet D. Iedema. “Automated reaction generation for polymer networks”. In: *Computers & Chemical Engineering* 112 (2018), pp. 37–47. ISSN: 0098-1354. DOI: <https://doi.org/10.1016/j.compchemeng.2018.01.022>. URL: <https://www.sciencedirect.com/science/article/pii/S0098135418300462>.
- [13] J. R. Ullmann. “An Algorithm for Subgraph Isomorphism”. In: *J. ACM* 23.1 (Jan. 1976), pp. 31–42. ISSN: 0004-5411. DOI: 10.1145/321921.321925. URL: <https://doi.org/10.1145/321921.321925>.
- [14] J.J.P. Veerman, T. Whalen-Wagner, and Ewan Kummel. “Chemical reaction networks in a Laplacian framework”. In: *Chaos, Solitons Fractals* 166 (2023), p. 112859. ISSN: 0960-0779. DOI: <https://doi.org/10.1016/j.chaos.2022.112859>. URL: <https://www.sciencedirect.com/science/article/pii/S0960077922010384>.