THE LONDON SCHOOL OF ECONOMICS AND POLITICAL SCIENCE

Connectivity Properties of Some Transformation Graphs

Somkiat Trakultraipruk

A thesis submitted to the Department of Mathematics
of the London School of Economics
for the degree of Doctor of Philosophy, London, January 2013

To my parents

Declaration

I certify that the thesis I have presented for examination for the MPhil/PhD degree of the London School of Economics and Political Science is solely my own work other than where I have clearly indicated that it is the work of others (in which case the extent of any work carried out jointly by me and any other person is clearly identified in it).

I warrant that this authorisation does not, to the best of my belief, infringe the rights of any third party.

Abstract

Many combinatorial problems can be formulated as "can we transform configuration 1 into configuration 2 if certain transformations are allowed?" In order to study such questions, we introduce a so-called *transformation graph*. This graph has the set of all possible configurations as its vertex set, and there is an edge between two configurations if one configuration can be obtained from the other by one of the allowed transformations. Then a question like "can we go from one configuration to another one" becomes a question about connectivity properties of transformation graphs.

In this thesis, we study the following types of transformation graphs in particular:

Labelled Token Graphs: Here configurations are arrangements of labelled tokens on a given graph, and we can go from one arrangement to another one by moving one token at a time along an edge of the given graph. We classify all cases when labelled token graphs are connected, and classify all pairs of arrangements that are in the same component. We also look at the problem how hard it is to determine the length of the shortest path between two arrangements.

Strong k-Colour Graphs: For this transformation graph, the configurations are the proper vertex-colourings of a given graph with k colours, in which all k colours are actually used. We call such a colouring a strong k-colouring. We study the problem when we can transform any strong k-colouring into any other one by recolouring one vertex at a time, always maintaining a strong k-colouring. For

certain classes of graphs, we can completely determine when the transformation graph of strong k-colourings is connected.

Acknowledgements

This thesis would not have been possible without the guidance and help of several individuals who in one way or another contributed and extended their valuable assistance in the preparation and completion of this study.

First, I would like to thank my first and second supervisors, Professor Jan van den Heuvel and Professor Graham Brightwell (Department of Mathematics, London School of Economics and Political Science, United Kingdom), whose encouragement and assistance helped me to complete this study.

Professor Mark Ellingham, Professor Ralph McKenzie, and Professor Michael D. Plummer (Department of Mathematics, Vanderbilt University, United States), who supported me through my Master's degree.

Associate Professor Wanida Hemakul and Assistant Professor Chariya Uiyyasathian (Department of Mathematics, Chulalongkorn University, Thailand), who offered constant support and guidance, and are the people who made it possible for me to study in both the US and UK. I will always owe them a huge debt of gratitude.

Mr Somsak Anansuvanchai, Ms Tassanee Arayatragullikit, Mr Banchar Arnonkijpanich, Assistant Professor Vannee Netsinghanart, Mr Somkiat Panoi, Associate Professor Tiang Poomsa-ard, and Mr Jirayuth Wathveerapong (Department of Mathematics, Khon Kaen University, Thailand), who were a big influence on me during my first degree, and I will always be grateful to them for their advice and encouragement.

Mr Som Prangnakhon, Mr Suthep Prasong, Ms Kobkue Ratchakit, and Mr Manote Ratchakit (Nangrong School, Buriram, Thailand), who encouraged my interest in Mathematics and English through their excellent teaching, fostering in me the desire to pursue both to a higher level.

Development and Promotion of Science and Technology Talents Project (DPST), Higher Education Strategic Scholarships for Frontier Research Network (SFR), and The Royal Thai Government, who have supported me throughout my studies, and without their assistance, it would not have been possible for me to have gone as far as I have in my academic career.

Jorge, Ahmad, Yavor, Pa Van, P'Noi, all my friends in London and Leeds, and friends at Old Siam, Thai Cottage, Thai Square, Manorom, The Cos Bar, and The Bok Bar, who made my time in the UK unforgettable. Thank you for your support and friendship.

P'Mee, P'Phong, P'Noi, P'Jang, P'Top, P'Aekk, and all my friends at Siam Cuisine and Siam Cafe, who helped and advised me in personal and professional matters. Thank you for making my time in the US so enjoyable.

Noi, Joe, Yok, Saaw, Nui, Palm, and David, who are my best friends forever. Thank you for always being there, and for always making me smile. (",)

Ming and Oomimm, I will never forget the time we spent together and the happiness that brought me.

Jeck Heng and Sim Muay, who have always given me good advice.

Finally, I would like to thank the people I love the most, Tia, Maa, Ree, Mart, and Pen, whose support and love have guided me through the good times and the bad. There are no seconds in the day in which I don't love you.

Contents

CHAPTER 1

Introduction

The primary focus of this thesis is to study "Transformation Graphs". But what are they? And why should we study them? It might be good to start the thesis by answering these two questions. We may answer the first question by using the following definition.

Let O be a collection of objects and T a collection of transformations, where each transformation changes one object into another object. The transformation graph has O as its vertex set, and there is an edge between two objects if one object can be obtained from the other by some transformation from T.

With the definition above, it may not be that clear what transformation graphs really are. So here we give some examples of transformation graphs.

The graph $TG(v, e)$

Let $G(v, e)$ be the class of all graphs with v vertices and e edges. Define the graph $TG(v, e)$ to be the graph having $G(v, e)$ as its vertex set, and two vertices are adjacent if one vertex can be obtained from the other by adding an edge and deleting another edge.

The (Δ, n)-Graph

Let $G(\Delta, n)$ be the class of all graphs with n vertices, and with maximum degree Δ. Define the (Δ, n)-Graph to be the graph having $G(\Delta, n)$ as its vertex set, and two vertices are adjacent if one vertex can be obtained from the other by either adding or deleting an edge.

We can see that these two examples have the vertex sets coming from the class of graphs with some specific properties. However, there are some other transformation graphs, getting the vertex sets from different ways, e.g., *Token Graphs*.

Token Graphs (Fabila-Monroy et al. [11])
For a graph G and positive integer k, let the *token graph* $F_k(G)$ be the graph whose vertices are k-subsets of $V(G)$, and two k-subsets are adjacent in $F_k(G)$ if their symmetric difference (the set of vertices which are in either of the sets but not in both sets) is a pair of adjacent vertices in G.

We have extended the study of Fabila-Monroy et al. by generalising the problem to any number of tokens. Additionally, tokens can be identical or different. For more details, see *Labelled Token Graphs* in Chapter 2.

Next, we will answer the second question, "why should we study transformation graphs?". A possible answer is the following. In some problems, we cannot solve them directly, or they are not easy to be solved directly, but when we transform them to other problems, they might be solved easier.

The *15-puzzle* is a sliding puzzle that consists of 15 squares numbered from 1 to 15 that are placed in a 4×4 box leaving one position out of the 16 empty. The object of the puzzle is to reposition the squares from a given arrangement into the configuration β shown in Figure 1.1 by making sliding moves that use the empty space.

Example 1.1
Let α and β be the two configurations of the 15-puzzle problem, which are shown in Figure 1.1. Is it possible to reach β from α?

This question can be easily answered by using labelled token graphs, and the answer will be "NO". See the arguments at the end of Chapter 2.

2	1	4	3
6	5	8	7
10	11	12	9
14	15	13	

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	

FIGURE 1.1: Configurations α and β of the *15-Puzzle*

1.1 Preliminaries

In this section we present some mathematical terminology and notation, which are used in this thesis. Most of them will be standard and can be found in any textbook on graph theory, such as [10] and [27].

Basic Concepts in Graph Theory

A graph G is defined by two finite sets $V(G)$ and $E(G)$, where an element of $V(G)$ is called a *vertex*, and an element of $E(G)$ is called an *edge*. Each edge is a two-element subset of $V(G)$. An edge between vertices u and v is denoted by uv. The *degree* of a vertex v in G, denoted by $d_G(v)$, is the number of edges of G incident with v. We omit the subscript G when it is clear from the context. The numbers of vertices and edges in G are denoted by $n(G)$ and $e(G)$, respectively.

A *walk* on n vertices is a sequence $v_1, v_2, \ldots, v_n$ of vertices where any two consecutive vertices are adjacent. A *path* on n vertices, denoted by P_n, is a walk with n vertices such that the vertices are all different. The *size* and the *length* of P_n are n and $n-1$, respectively. A *cycle* on n vertices, denoted by C_n, is a sequence $v_1, v_2, \ldots, v_n, v_{n+1}$, where any two consecutive vertices are adjacent, and the vertices are all different except $v_1 = v_{n+1}$.

The *distance* between two vertices in a graph is the length of a shortest path connecting them.

A graph is *connected* if for any two vertices there is a path between them. A *subgraph* of a graph G is a graph H such that $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$. A *spanning subgraph* of G is a subgraph with vertex set $V(G)$. An *induced subgraph* of a graph G is a graph H such that for any pair of vertices x and y of H, xy is an edge of H if and only if xy is an edge of G. A *component* is a maximal connected subgraph.

A *vertex cut* of a graph G is a set of vertices S such that $G \backslash S$ has more components than G, where $G \backslash S$ denotes the graph obtained after deleting the vertices in S, together with the edges incident with these vertices. If $S = \{v\}$ is a vertex cut, we call v a *cut-vertex*. An edge e is a *cut-edge* of G if $G \backslash \{e\}$ (the graph obtained after deleting the edge e) has more components than G.

A *block* is a maximal connected subgraph without cut-vertices.

A graph is k-*connected* if it has at least $k+1$ vertices and there is no vertex cut with $k-1$ or fewer vertices. Here is a theorem about k-connected graphs, which can be found in any textbook of graph theory.

Theorem 1.2 (Global Version of Menger's Theorem)
A graph G with $n(G) \geq k+1$ is k-connected if and only if for every $u, v \in V(G)$ there exist k internally disjoint paths from u to v.

The *graph Cartesian product* $G \Box H$ of graphs G and H is the graph such that $V(G \Box H) = \{(u, v) : u \in V(G) \text{ and } v \in V(H)\}$, and two vertices (u_1, v_1) and (u_2, v_2) are adjacent in $G \Box H$ if and only if either $u_1 = u_2$ and v_1 is adjacent with v_2 in H, or $v_1 = v_2$ and u_1 is adjacent with u_2 in G. A *grid graph* is the graph Cartesian product of two paths.

A *complete graph* is a graph where any two vertices are adjacent. We denote a complete graph with n vertices by K_n. A *bipartite graph* is a graph without odd cycles. For a bipartite graph, we can partition the vertices into two sets X and Y such that every edge has one end vertex in X and one in Y. A *complete bipartite graph* has an edge from every vertex in X to every vertex in Y. We denote the complete bipartite graph with p vertices in X and q vertices in Y by $K_{p,q}$. A *planar graph* is a graph that can be drawn on the plane such that its edges intersect only at their endpoints. A *tree* is a connected graph with no cycles. And the graph θ_0 is the graph shown in Figure 1.2.

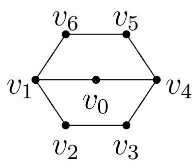


FIGURE 1.2: The graph θ_0

Permutation Groups

A *permutation* of a set is a bijection from that set onto itself. The set of all permutations of any given set S forms a group, with composition of functions as product and the identity as identity element. This group is the *symmetric group* of S. A *transposition* is a permutation which exchanges two elements and keeps all others fixed. Note that every permutation can be written as a product of transpositions. A permutation is *odd* if it can be written as a product of an odd number of transpositions; otherwise, it is an *even* permutation. Since bijections have inverses, so do permutations. The *inverse* of a permutation α, denoted by α^{-1}, is the permutation such that $\alpha(x) = y$ if and only if $\alpha^{-1}(y) = x$ for any elements x, y.

Complexity and Algorithms

In this thesis we need a number of concepts from algorithmic complexity theory. The formal definitions of these concepts would be beyond the scope of this thesis. In this section we give some informal descriptions only; more formal definitions can be found in any textbook such as [9] and [14].

The *time complexity* of an algorithm quantifies the amount of time taken by the algorithm to run as a function of the length of the string representing the input. The time complexity of an algorithm is commonly expressed using *big O* notation, which is defined as the following.

For functions $f, g : \mathbb{Z}^+ \to \mathbb{R}$, we say that $f(n) = O(g(n))$ (pronounced $f(n)$ is big O $g(n)$) if there are constants $N \geq 1$ and $C > 0$ such that for all $n \geq N$ we have $|f(n)| \leq C|g(n)|$. For example, if the time required by an algorithm on all inputs of size n is at most $7n^3 + 5n$, its time complexity is said to be $O(n^3)$.

An algorithm is said to be of *polynomial time* if its running time is upper bounded by a polynomial expression in the size of the inputs for the algorithm, i.e., if the time complexity is $O(n^k)$ for some constant k, where n is the size of the inputs.

A *decision problem* is a problem whose output is a single Boolean value: *YES* or *NO*.

Let A and B be two decision problems. A *reduction* from A to B is a function f which transforms inputs of A to equivalent inputs of B. That is given an input x

to the problem A, f will produce an input $f(x)$ to the problem B, such that x is a "YES" input of A if and only if $f(x)$ is a "YES" input of B. This reduction can be used to define complexity classes on a set of problems. Intuitively, if a problem A is reducible to a problem B, then an algorithm for solving B efficiently (if it existed) could also be used as a subroutine to solve A efficiently provided the reduction can be done efficiently as well. Thus solving A cannot be harder than solving B.

The set P is the set of decision problems that can be solved in polynomial time, and the set *NP* is the set of decision problems with the following properties: if the answer is "YES", then there is a certificate of that yes-answer which can be checked in polynomial time. A problem is *NP-hard* if a polynomial-time algorithm for the problem would imply a polynomial-time algorithm for every problem in *NP*. A problem is *NP-complete* if it is both *NP*-hard and an element of *NP*.

Note that to prove a problem is *NP*-complete, all we need to do are:
(1) showing that the problem is in *NP*,
(2) reducing a known *NP*-complete problem to it, and
(3) showing that the reduction is a polynomial-time computable function.

To give some examples of *NP*-complete problems, we define a *Boolean formula* to be an expression written using only *AND*, *OR*, *NOT*, variables, and parentheses; e.g., $(a \vee b \vee \bar{c}) \wedge ((a \wedge \bar{b}) \vee (c \wedge \bar{d}))$. A *literal* is either a variable or its negation. A Boolean formula is in *conjunctive normal form* if it is a conjunction (AND) of several clauses, each of which is a disjunction (OR) of one or more literals; e.g., $\underbrace{(a \vee b \vee \bar{c})}_{\text{clause}} \wedge (b \vee \bar{c}) \wedge (\bar{a} \vee d) \wedge (a \vee \bar{b} \vee d)$.

The following are *NP*-complete problems.

SAT

Input: A Boolean formula;

Question: Is there a truth assignment for the variables which satisfies the formula?

3-SAT

Input: A Boolean formula which is in conjunctive normal form with 3 literals per clause;

Question: Is there a truth assignment for the variables which satisfies the formula?

3-Exact-Cover

Input: A set $U = \{u_i\}_{i=1}^{3n}$ and a collection $S = \{s_j\}_{j=1}^{m}$ of 3-element subsets of U;

Question: Is there a subcollection $S' \subseteq S$ such that every element in U occurs in exactly one member of S'?

The *complement* of a decision problem is the decision problem, which results from replacing the problem by the negation of that problem. For example, one important problem is to consider if a number is a prime number. Its complement is to ask if a number is a composite number (a number which is not prime).

A decision problem is a member of *coNP* if and only if its complement is in *NP*. Each *coNP-complete* problem is the complement of an *NP*-complete problem.

A decision problem is in *PSPACE* if it can be solved by an algorithm using polynomial storage space. A decision problem is *PSPACE-complete* if it is in *PSPACE*, and every problem in *PSPACE* can be reduced to it in polynomial time.

Minors

Let G be a graph. A *vertex-deletion* of G means removing a vertex from the vertex set of G and removing all edges incident to it. An *edge-deletion* of G means removing an edge from the edge set of G. An *edge-contraction* of an edge uv means replacing the vertices u and v by a new vertex w, then joining w to all vertices that were adjacent to u or v, and finally replacing any multiple edges which arise by a single edge.

A graph H is a *minor* of a graph G if H can be obtained from G by a sequence of vertex-deletions, edge-deletions, and edge-contractions. Note that in the definition, we allow a sequence of length zero, so every graph is a minor of itself.

In this thesis, it is common to assume that all graphs are connected. Then if H is a minor of a connected graph G, H can be obtained from G by a sequence of edge-deletions and edge-contractions only (since a vertex-deletion can be obtained by an edge-contraction and some edge-deletions in a connected graph).

Orderings

A relation $\leq$ on a set X is a *quasi-ordering* if it is reflexive ($x \leq x$ for all $x \in X$) and transitive (if $x \leq y$ and $y \leq z$, then $x \leq z$ for all $x, y, z \in X$). And $\leq$ is a *well-quasi-ordering* if it is a quasi-ordering, and for any infinite sequence of elements $x_0, x_1, x_2, \ldots$ from X, there are two indices $i < j$ such that $x_i \leq x_j$. Note that sometimes we use $x \geq y$ to denote $y \leq x$, and $y < x$ to denote $y \leq x$ and $y \neq x$.

A quasi-ordering is *without infinite descent* if there are no infinite strictly decreasing sequences $x_1 > x_2 > x_3 > \cdots$. An *antichain* $A \subseteq X$ is a subset such that no two elements in A are comparable.

Let P be a property defined on elements of a set X. We say that P is *closed* under a quasi-ordering $\leq$ (or $\leq$-*closed*) if for every two elements $x, y \in X$ we have that if x has property P and $y \leq x$, then y also has property P.

Let $\leq$ be a well-quasi-ordering on a set X, and P a $\leq$-closed property on X. Then an element $x \in X$ is said to be *P-forbidden* if it is a minimal element in X under $\leq$, which does not have property P.

Labelled Token Graphs

Let G be a graph, and $k_1, k_2, \ldots, k_p$ positive integers for some $p \in \mathbb{Z}^+$. We assume we have $k_1 + k_2 + \cdots + k_p$ labelled tokens, where there are k_i tokens with label i, for $i = 1, 2, \ldots, p$. A *token configuration* is an arrangement of all these tokens on the vertices of G such that no two tokens are placed on the same vertex.

The *labelled token graph* of G with a multiset of tokens $(k_1, k_2, \ldots, k_p)$ is the graph whose vertices correspond to all possible different token configurations, and two token configurations are adjacent if one token configuration can be reached from the other by moving one token along an edge of G. We denote the labelled token graph by $T(G; (k_1, k_2, \ldots, k_p))$.

Strong k-Colour Graphs

For a positive integer k, a *k-vertex-colouring* of a graph G is a function $f : V(G) \to \{1, 2, \ldots, k\}$. A colouring is *proper* if adjacent vertices have different labels. A graph is *k-colourable* if it has a proper k-vertex-colouring. The smallest number of colours needed to colour a graph is called its *chromatic number*.

The *strong k-colour graph* of a graph G and integer k, denoted by $S_k(G)$, is the graph that has the proper k-vertex-colourings of G using exactly k colours as its vertices, and two such colourings are joined by an edge if they differ in the colour on one vertex only.

1.2 Outline of the Thesis

In Chapter 2, we work on labelled token graphs, and we prove two main theorems. The first theorem gives sufficient and necessary conditions of labelled token graphs being disconnected. The other main result tells exactly the situation whether two given token configurations are in the same component of a labelled token graph or not.

Chapter 3 contains some miscellaneous results on labelled token graphs. We first consider the computational hardness of finding the shortest path between two token configurations in labelled token graphs. Then we propose another way to prove being disconnected of labelled token graphs by considering whether a given graph contains certain forbidden minors or not, and we prove that the number of these forbidden minors is finite for all the numbers of tokens.

In the last chapter, we study strong k-colour graphs. We give some general results and characterise when strong k-colour graphs are connected for some classes of graphs.

In the appendix, we talk about the graph θ_0 again because it is the only 2-connected non-bipartite graph so that not all its labelled token graphs are connected. We show the components of labelled token graphs of θ_0 in a compact form, which will be described later.

The results in Chapter 2 are joint work with Professor Graham Brightwell and Professor Jan van den Heuvel, whilst the results in Chapter 3 are joint work with Professor Jan van den Heuvel. Chapter 4 can be found in [25], and we are preparing publication of the results from Chapters 2 and 3 in [4] and [26], respectively.

Chapter 2

Labelled Token Graphs

There are many problems and games, concerned with moving objects around. Examples are puzzle games, transportation, manufacturing, scheduling, control flow and management of memory in computing systems. These problems have been extensively studied.

Let G be a graph, and $k_1, k_2, \ldots, k_p$ positive integers for some positive integer p. We have a number of classes of tokens, say k_1 tokens with label 1, k_2 tokens with label 2, etc. Tokens with the same label are indistinguishable. A *token configuration* is an arrangement of all tokens on vertices of G such that no two tokens are placed on the same vertex. We use Greek letters $\alpha, \beta, \ldots$ for token configurations, and we denote by $\alpha(v) = t$ that in the token configuration α, the vertex v contains a token with label t. A vertex which does not contain any token is said to be *unoccupied*, and we sometimes say that it contains an *empty token*, denoted by ϕ.

The *labelled token graph* of G with a multiset of tokens $(k_1, k_2, \ldots, k_p)$ is the graph whose vertices correspond to token configurations, and two token configurations are adjacent if one token configuration can be reached from the other by moving one token along an edge of G. We denote the labelled token graph as $T(G; (k_1, k_2, \ldots, k_p))$, and we always assume that $k_1 \geq k_2 \geq \cdots \geq k_p$. For convenience purposes, when the tokens are all different ($k_1 = k_2 = \cdots = k_p = 1$), we will denote $T(G; (k_1, k_2, \ldots, k_p))$ as $T(G; p)$.

To calculate the number of vertices in $T(G;(k_1,k_2,\ldots,k_p))$ of a given graph G and positive integers $k_1,k_2,\ldots,k_p$, let $k=k_1+k_2+\cdots+k_p$. We assume that $k\leq n(G)-1$ to avoid trivial cases. Then

$$n(T(G;(k_1,k_2,\ldots,k_p)))=\frac{n(G)!}{(n(G)-k)!k_1!k_2!\cdots k_p!}.$$

When all tokens are different, we have that

$$n(T(G;p))=\frac{n(G)!}{(n(G)-p)!}.$$

Next, we calculate the number of edges in labelled token graphs. Here we consider only the case that all tokens are different. For general cases, we cannot find a similar formula easily. Since moving a token along each edge of G represents $\frac{\binom{p}{1}(n(G)-2)!}{((n(G)-2)-(p-1))!}$ edges in $T(G;p)$, we have that

$$e(T(G;p))=e(G)\cdot\left[\frac{p\cdot(n(G)-2)!}{(n(G)-p-1)!}\right].$$

With only one token, the resulting labelled token graph is isomorphic to G, i.e.,

$$T(G;1)\cong G.$$

When there are $n(G)-1$ different tokens, a token configuration is a bijective mapping $f:\left\{v_1,v_2,\ldots,v_{n(G)}\right\}\to\{1,2,\ldots,n(G)-1,\phi\}$. Thus we may consider a token configuration as a permutation of the set of tokens. As standard, we denote the inverse of a permutation α by α^{-1}, and use the same notation for the type of token configurations in this paragraph.

2.1 Literature

In this thesis we use the term "token" as a description for a moveable object. Other literatures looking at similar problems use terms such as *agent*, *pebble*, *bean*, *robot*, or *vehicle* to describe the moveable object. Some papers consider problems in which the tokens are all different [1, 18, 28], whilst others consider problems where all tokens are the same [11]. We study problems in which tokens can be identical or distinct, which can be also seen in [13, 16]. And the rules of moving tokens in those studies may be the same or different. For example, in [1, 11, 16, 18, 28], a token can be moved from one vertex to one of its unoccupied neighbours.

Wilson [28] generalised the classic problem, the 15-Puzzle, on 2-connected graphs with n vertices and $n-1$ different tokens.

Theorem 2.1 (Wilson [28])
Let G be a 2-connected graph, which is not a cycle or the graph θ_0. Then the token graph $T(G; n(G)-1)$ is connected, unless G is bipartite. In the latter case, $T(G; n(G)-1)$ has exactly two components, and token configurations α and β are in the same component if and only if they have unoccupied vertices at even distance in G and $\alpha\beta^{-1}$ is an even permutation, or they have unoccupied vertices at odd distance in G and $\alpha\beta^{-1}$ is an odd permutation. The labelled token graph $T(\theta_0; 6)$ has six components.

Kornhauser et al. [18] followed on from Wilson's study by generalising the problem for all graphs and any number of tokens. They also presented a polynomial time algorithm and gave $O(n^3)$ upper and lower bounds for deciding the reachability of any two token configurations.

Auletta et al. [1] also studied on the reachability of any two token configurations, but focused on trees. They gave a linear decision algorithm for those cases.

Goraly and Hassin [16] studied problems where tokens can be identical or distinct, as described earlier. They proved that the reachability of any two token configurations can be decided in linear time.

Fabila-Monroy et al. [11] introduced the word "Token Graph", which we use in this thesis although they only consider the case that all tokens are identical. They gave tight lower and upper bounds on the diameter, and tight lower bounds on the connectivity of their token graphs. They also gave some results on cliques, chromatic numbers, and Hamiltonian paths of their token graphs.

Fujita et al. [13] generalised the problems of moving tokens by defining a move to be an exchange of two tokens with distinct labels on the two endvertices of a common edge (where an unoccupied vertex is supposed to have an empty token). They studied connectivity properties of such token graphs.

Wu and Grumbach [29] studied on motion planning on directed graphs (graphs whose every edge has a direction from one vertex to the other). Their objective is to move a special token (called *robot*) from a source vertex to another destination vertex while the other tokens are considered as just obstacles. They proved that this problem on acyclic (without any directed cycles) strongly connected directed graphs can be decided in time $O(nm)$, where n and m are the numbers of vertices and edges, respectively. And it can be decided in time $O(n^2m)$ on any directed graphs.

2.2 Main Results on Connectivity of Labelled Token Graphs

One of our main results is showing and proving sufficient and necessary conditions of being disconnected of labelled token graphs. If $p = 1$, then the tokens are all the same. Thus the labelled token graph is disconnected if and only if the original graph is disconnected. So we always assume that $p \geq 2$. Under that assumption, if $n(G) = 2$, the labelled token graph is disconnected. Hence we also assume that $n(G) \geq 3$.

Let G be a connected graph. A *separating path* of size one in G is a cut-vertex. A separating path of size two in G is a cut-edge $P = v_1v_2$ such that both components of $G - v_1v_2$ have at least two vertices. A separating path of size $l \geq 3$ in G is an induced path $P = v_1v_2 \dots v_{l-1}v_l$ such that $G \backslash \{v_2, v_3, \dots, v_{l-1}\}$ has exactly two components, one contains v_1 and the other contains v_l, and each of which has at least two vertices. When a separating path P is of size at least two, we say that G has two *P-components* (the components described above).

Theorem 2.2

Let G be a graph with $n(G) \geq 3$, and $k_1 \geq k_2 \geq \cdots \geq k_p$ positive integers for some integer $p \geq 2$. Then $T(G; (k_1, k_2, \dots, k_p))$ is disconnected if and only if at least one of the following conditions holds:

1. *G is disconnected;*

2. *$k_1 + k_2 + \cdots + k_p = n(G)$;*

3. *G is a path;*

4. *G is a cycle with $p = 2$ and $k_2 \geq 2$, or G is a cycle with $p \geq 3$;*

5. *G is the graph θ_0 (shown in Figure 2.1), and $(k_1, k_2, \ldots, k_p)$ is one of the following: (2,2,2), (2,2,1,1), (2,1,1,1,1), or (1,1,1,1,1,1);*

6. *G is a 2-connected bipartite graph other than a cycle, $k_1 + k_2 + \cdots + k_p = n(G) - 1$, and $k_1 = k_2 = \cdots = k_p = 1$ (tokens are all different);*

7. *G is a connected graph with connectivity 1 other than a path, containing a separating path of size $n(G) - (k_1 + k_2 + \cdots + k_p)$.*

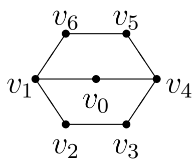


FIGURE 2.1: The graph θ_0

The other main result answers whether two given token configurations α and β are in the same component of the labelled token graph or not. We can see that α and β are in the same component if and only if on each component G_i of G, $\alpha|_{G_i}$ (the token configuration on G_i induced by α) is reachable from $\beta|_{G_i}$. Hence without loss of generality, we can again assume that G is connected. Next, if $k_1+k_2+\cdots+k_p = n(G)$, different token configurations are in different components of the labelled token graph. So we also suppose that $k_1+k_2+\cdots+k_p \leq n(G)-1$.

According to the graph θ_0 in Figure 2.1, a token configuration α of θ_0 is *standard* if v_1 is unoccupied in α, and it is *(s,t)-standard* if it is standard and the tokens s and t are placed on the vertices v_6 and v_5, respectively. We call the cycle $v_1v_2v_3v_4v_0$ on θ_0 as the *lower 5-cycle*, the cycle $v_1v_0v_4v_5v_6$ as the *upper 5-cycle*, and the cycle $v_1v_2v_3v_4v_5v_6$ as the *outside 6-cycle*.

Let G be a connected graph with connectivity 1, $n(G) - (k_1 + k_2 + \cdots + k_p) = l \geq 2$, and $P_1, P_2, \ldots, P_m$ all the separating paths of size l in G. For each $i = 1, 2, \ldots, m$, let $G_{i,1}$ and $G_{i,2}$ be the two P_i-components of G. Given a token configuration α, let α_i be a token configuration obtained from α by moving some tokens (if necessary) to make all the vertices on P_i unoccupied.

Let G be a connected graph with connectivity 1, $n(G) - (k_1 + k_2 + \cdots + k_p) = 1$, and B a block in G. Then B contains at least one cut-vertex of G. Let v_B be one of these cut-vertices. Given a token configuration α, let α_{v_B} be a token configuration obtained from α by moving some tokens (if necessary) to make v_B unoccupied.

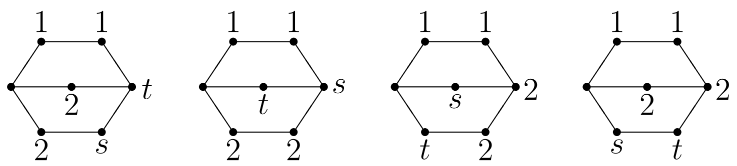
We denote the multiset of all the tokens used in a token configuration α by $\tau(\alpha)$. For example, if α is any of the token configurations in Figure 2.4, then $\tau(\alpha) = \{1, 1, 2, 2, 3, 3\} = (2, 2, 2)$.

Theorem 2.3

Let G be a connected graph with $n(G) \geq 3$, $k_1 \geq k_2 \geq \cdots \geq k_p$ positive integers for some integer $p \geq 2$, and $k_1 + k_2 + \cdots + k_p \leq n(G) - 1$. Then two token configurations α and β are in the same component of $T(G; (k_1, k_2, \ldots, k_p))$ if and only if at least one of the following conditions holds:

1. *$T(G; (k_1, k_2, \ldots, k_p))$ is connected;*
2. *G is a path, and the orders of tokens on G of α and β are the same;*
3. *G is a cycle, and the cyclic orders of tokens on G of α and β are the same;*
4. *G is the graph θ_0, and*
   (a) *$(k_1, k_2, \ldots, k_p) = (2, 2, 2)$ or $(2, 2, 1, 1)$, and for any (1,1)-standard token configurations α' and β' which can be reached from α and β, respectively, we have that α' and β' are in the same group from the following two groups:*

Group a_1*: (1,1)-standard token configurations of which the cyclic order of tokens on the lower 5-cycle is* $(2, 2, s, t)$*, where* $s, t \in \{3, 4\}$*. I.e., token configurations which have the following forms:*



Group a_2*: (1,1)-standard token configurations of which the cyclic order of tokens on the lower 5-cycle is* $(2, s, 2, t)$*, where* $s, t \in \{3, 4\}$*,*

(b) $(k_1, k_2, \ldots, k_p) = (2, 1, 1, 1, 1)$*, and for any (1,1)-standard token configurations* α' *and* β' *which can be reached from* α *and* β*, respectively, we have* α' *and* β' *are in the same group from the following three groups:*

Group b_1*: (1,1)-standard token configurations of which the cyclic order of tokens on the lower 5-cycle is* $(2, 3, 4, 5)$ *or* $(2, 5, 4, 3)$*;*

Group b_2*: (1,1)-standard token configurations of which the cyclic order of tokens on the lower 5-cycle is* $(2, 4, 3, 5)$ *or* $(2, 5, 3, 4)$*;*

Group b_3*: (1,1)-standard token configurations of which the cyclic order of tokens on the lower 5-cycle is* $(2, 3, 5, 4)$ *or* $(2, 4, 5, 3)$*;*

(c) $(k_1, k_2, \ldots, k_p) = (1, 1, 1, 1, 1, 1)$*, and for any (1,6)-standard token configurations* α' *and* β' *which can be reached from* α *and* β*, respectively, we have* α' *and* β' *are in the same group from the following six groups:*

Group c_1*: (1,6)-standard token configurations of which the cyclic order of tokens on the lower 5-cycle is* $(2, 3, 4, 5)$*;*

Group c_2*: (1,6)-standard token configurations of which the cyclic order of tokens on the lower 5-cycle is* $(2, 5, 4, 3)$*;*

Group c_2*: (1,6)-standard token configurations of which the cyclic order of tokens on the lower 5-cycle is* $(2, 4, 3, 5)$*;*

Group c_4*: (1,6)-standard token configurations of which the cyclic order of tokens on the lower 5-cycle is* $(2, 5, 3, 4)$*;*

Group c_5*: (1,6)-standard token configurations of which the cyclic order of tokens on the lower 5-cycle is* $(2, 3, 5, 4)$*;*

Group c_6*: (1,6)-standard token configurations of which the cyclic order of tokens on the lower 5-cycle is* $(2, 4, 5, 3)$*.*

5. *G is a 2-connected bipartite graph other than a cycle, there are $n(G) - 1$ different tokens, and one of the following holds:*

   (a) *α and β have their unoccupied vertices at even distance in G, and $\alpha\beta^{-1}$ is an even permutation;*

   (b) *α and β have their unoccupied vertices at odd distance in G, and $\alpha\beta^{-1}$ is an odd permutation.*

6. *G is a connected graph with connectivity 1 other than a path, $n(G)-(k_1+k_2+\cdots+k_p) = l \geq 2$, $P_1, P_2, \ldots, P_m$ are all the separating paths of size l in G, and $\tau(\alpha_i|_{G_{i,1}}) = \tau(\beta_i|_{G_{i,1}})$ and $\tau(\alpha_i|_{G_{i,2}}) = \tau(\beta_i|_{G_{i,2}})$ for all $i = 1, 2, \ldots, m$.*

7. *G is a connected graph with connectivity 1 other than a path, $n(G) - (k_1 + k_2 + \cdots + k_p) = 1$, for each block B in G, $\tau(\alpha_{v_B}|_B) = \tau(\beta_{v_B}|_B)$, and at least one of the following conditions holds:*

   (a) *$T(B; \tau(\alpha_{v_B}|_B))$ is connected;*

   (b) *B is a cycle, and the cyclic orders of tokens of $\alpha_{v_B}|_B$ and $\beta_{v_B}|_B$ are the same;*

   (c) *B is the graph θ_0, and $\alpha_{v_B}|_B$ and $\beta_{v_B}|_B$ satisfy 4(a), 4(b), or 4(c) above;*

   (d) *B is a 2-connected bipartite graph other than a cycle, there are $n(B)-1$ different tokens in $\alpha_{v_B}|_B$ and $\beta_{v_B}|_B$, and $\alpha_{v_B}|_B \cdot (\beta_{v_B}|_B)^{-1}$ is an even permutation.*

2.3 Proof of Theorem 2.2

In some figures which follow, we add a subscript to a label to see how the tokens are moved; however, tokens with the same label (but with different subscripts) should be interpreted as being indistinguishable.

Let α be any token configuration of a graph G, and v_1 and v_2 vertices on G. Let β be a token configuration of G such that $\beta(v_1) = \alpha(v_2)$, $\beta(v_2) = \alpha(v_1)$, and $\beta(v) = \alpha(v)$ for all $v \neq v_1, v_2$. We say that tokens $\alpha(v_1)$ and $\alpha(v_2)$ can be *improperly swapped* in α if there is a sequence of moves to get β from α. We call such a swap an *improperly empty-token swap* when we improperly swap a token with an empty token. Note that if the tokens are all different, an improper swap of two tokens is just swapping the positions of them.

In Figure 2.2, we can see that in α the tokens 2 and 3 can be improperly swapped by moving the tokens on the cycle until we get the token configuration β.

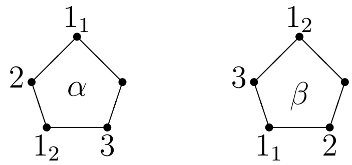


FIGURE 2.2: Token configurations α and β

Lemma 2.4

Let G be a connected graph, and $k_1, k_2, \ldots, k_p$ positive integers for some integer p. Then $T(G; (k_1, k_2, \ldots, k_p))$ is connected if and only if any two tokens in any token configuration can be improperly swapped.

Proof. If the labelled token graph is connected, then any two tokens in any token configuration can be improperly swapped since improper swap is a special case of transformation from one token configuration to another one in the labelled token graph. On the other hand, we suppose that any two tokens in any token configuration can be improperly swapped. Given any two token configurations α and β, we can assume that the sets of all unoccupied vertices in α and β are the same (otherwise, we move some tokens in β to get another token configuration which we want and then call it β). Then β is reachable from α by a sequence of improper swaps of two tokens. $\square$

Since we can improperly swap two tokens by doing three consecutive improperly empty-token swaps, we get the following lemma.

Lemma 2.5

Let G be a connected graph, and $k_1, k_2, \ldots, k_p$ positive integers for some integer p. Then $T(G; (k_1, k_2, \ldots, k_p))$ is connected if and only if any token and any empty token in any token configuration can be improperly empty-token swapped.

The following Properties are easily proved by using Lemma 2.4.

Proposition 2.6

Let $p \leq q$, and $k_i \leq l_i$ for all i. If $T(G; (l_1, l_2, \ldots, l_q))$ is connected, then the token graph $T(G; (k_1, k_2, \ldots, k_p))$ is connected.

Proposition 2.7

For each $i = 1, 2, \ldots, p$, we refine k_i as a summation of positive integers, say $k_i = k_{i,1} + k_{i,2} + \cdots + k_{i,n_i}$ for some positive integer n_i. Let $(l_1, l_2, \ldots, l_q)$ be the non-increasing reordering of $(k_{1,1}, k_{1,2}, \ldots, k_{1,n_1}, \ldots, k_{p,1}, k_{p,2}, \ldots, k_{p,n_p})$, where $q = n_1 + n_2 + \cdots + n_p$. If $T(G; (l_1, l_2, \ldots, l_q))$ is connected, then $T(G; (k_1, k_2, \ldots, k_p))$ is connected.

Proposition 2.8

Let H be a connected graph, which contains a subgraph G. If $T(G;(k_1,k_2,\ldots,k_p))$ is connected, then $T(H;(k_1,k_2,\ldots,k_p))$ is connected.

Lemma 2.9

Let G be a cycle. Then $T(G;(k_1,k_2,\ldots,k_p))$ is connected if and only if $p=1$, or if $p=2$, $k_1 \leq n(G)-2$, and $k_2=1$.

Next, we consider the graph θ_0. According to Figure 2.1, we call v_1 and v_4 *middle vertices* of θ_0. Applying an operation U (respectively, L) on a standard token configuration α is making 5 moves of the tokens on the upper (respectively, lower) 5-cycle anti-clockwise to get another standard token configuration (see Figure 2.3 for examples).

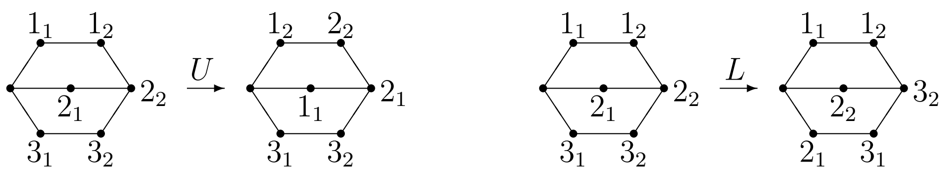


FIGURE 2.3: Operations U and L

On a standard token configuration, making 6 moves of the tokens on the outside 6-cycle anti-clockwise can be replaced by using the operation U and then L. Thus two standard token configurations are in the same component of the labelled token graph if and only if one configuration can be obtained from the other by a finite sequence of operations U and L.

Lemma 2.10

The labelled token graphs $T(\theta_0;(2,2,2))$, $T(\theta_0;(2,2,1,1))$, $T(\theta_0;(2,1,1,1,1))$, and $T(\theta_0;(1,1,1,1,1,1))$ are all disconnected.

Proof. By Proposition 2.7, if $T(\theta_0;(2,2,2))$ is disconnected, then the other labelled token graphs are also disconnected. So it suffices to show that $T(\theta_0;(2,2,2))$ is disconnected. Note that every token configuration in $T(\theta_0;(2,2,2))$ has a path to some standard token configuration. There are $\frac{6!}{2!2!2!} = 90$ standard token configurations in total, which are shown in Figures 2.4 and 2.5.

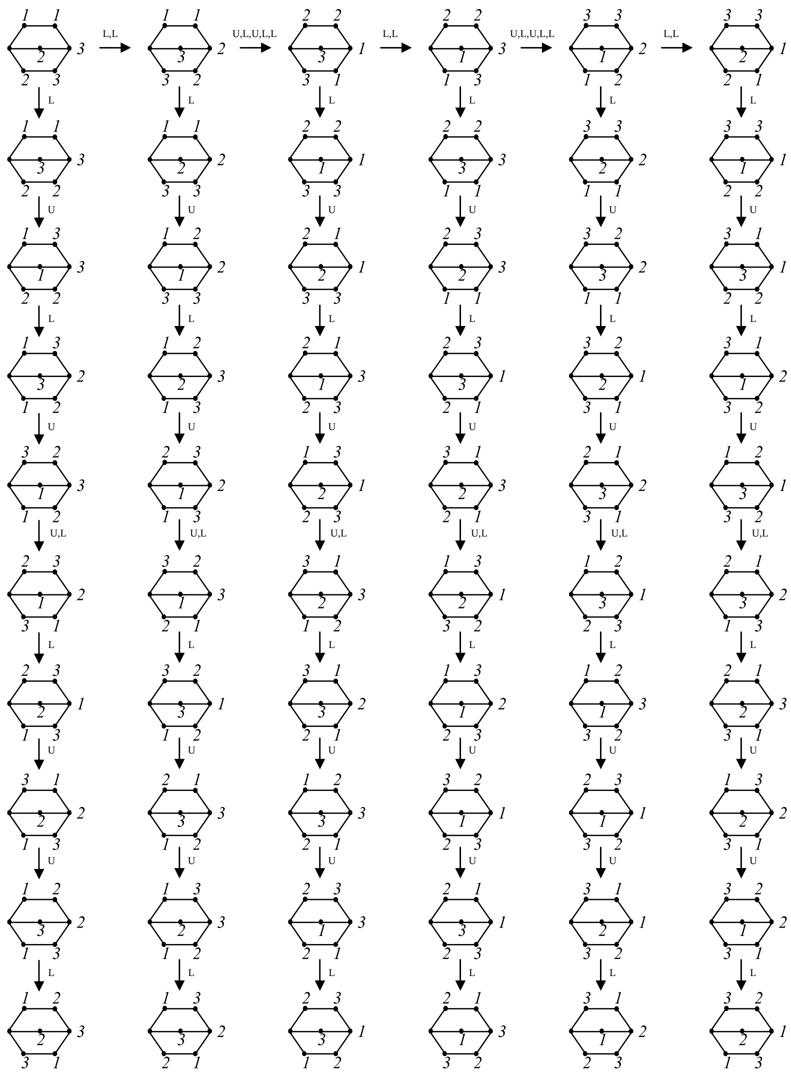


FIGURE 2.4: The 60 standard token configurations of Group a_1 in $T(G;(2,2,2))$

We can see that standard token configurations in the same figure are in the same component since they are joined by a path which can be generated by the given

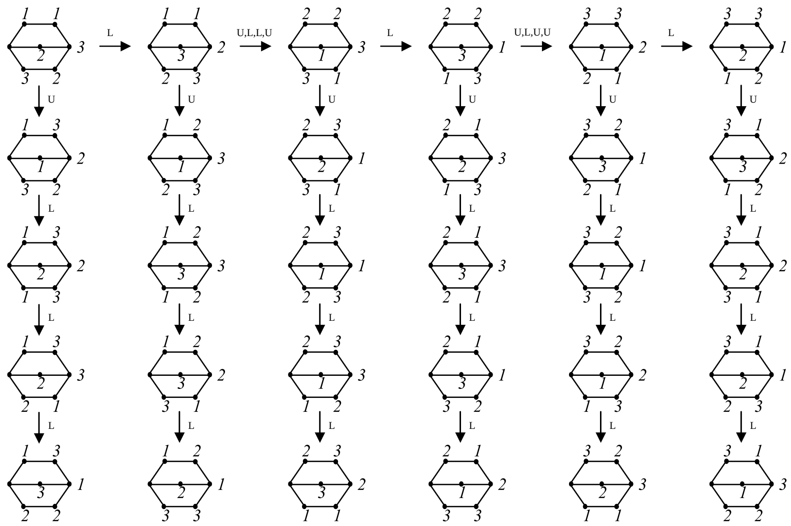


FIGURE 2.5: The 30 standard token configurations of Group a_2 in $T(G;(2,2,2))$

operations. Thus $T(\theta_0;(2,2,2))$ has at most 2 components. As described previously, two standard token configurations are in the same component if and only if one can be obtained from the other by using operations U and L only. However, by applying the operations U and L to each standard token configuration in Figures 2.4 and 2.5, we always get another token configuration in the same figure. Therefore, $T(\theta_0;(2,2,2))$ is not connected and has 2 components. $\square$

By considering all the standard token configurations, we find that $T(\theta_0;(2,2,1,1))$ has 2 components, $T(\theta_0;(2,1,1,1,1))$ has 3 components, and $T(\theta_0;(1,1,1,1,1,1))$ has 6 components. The last case was already proved by Wilson [28].

Lemma 2.11

The labelled token graph $T(\theta_0;(3,1,1,1))$ is connected.

Proof. We prove by using Lemma 2.4. Let $\{1_1,1_2,1_3,2,3,4\}$ be the multiset of tokens, and α any token configuration in $T(\theta_0;(3,1,1,1))$.

First, we show that a token 1 can be improperly swapped with a token different from 1. The following are the steps to improperly swap the tokens 1_1 and 2. For other pairs, we can do in a similar way.

Step 1: Move tokens so that 3 occupies v_0.

Step 2: Move tokens on the 6-cycle until 4 is on a middle vertex, and the other middle vertex is unoccupied. Then move 3 to this vertex.

Step 3: Move tokens on the 5-cycle which 2 is on until 1 and 2 are on the middle vertices, so we now have that $1_1, 1_2, 1_3$, and 2 are on the same 5-cycle.

Step 4: Improperly swap 1_1 and 2.

Step 5: Move all the tokens we moved in the previous steps backward to make 3 and 4 go back to their initial positions.

Next, we show that any two tokens $i, j \in \{2, 3, 4\}$ can be improperly swapped. Here we show only that the tokens 2 and 3 can be improperly swapped. For other pairs, we can so do in a similar way. Let $\alpha(v_x) = 1_1$, $\alpha(v_y) = 2$, and $\alpha(v_z) = 3$ for some $x, y, z \in \{0, 1, \ldots, 6\}$. We first improperly swap 1_1 and 2. Note that now 1_1 may or may not be on v_y. We then improperly swap the token 1 occupying v_y with 3. Finally, we improperly swap the token 1 occupying v_z with 2. □

Lemma 2.12

The labelled token graph $T(\theta_0; (1, 1, 1, 1, 1))$ is connected.

Proof. We prove by using Lemma 2.4. Let α be any token configuration in $T(\theta_0; (1, 1, 1, 1, 1))$, and t_1 and t_2 tokens in α. We first move some tokens to make t_1 occupy v_0. Then rotate the tokens on the outer cycle until there is an unoccupied vertex, which is adjacent to t_1, t_2, and another unoccupied vertex. We now can easily swap t_1 and t_2. Then move all the tokens we moved in the previous steps backward until we get the token configuration, which is the same as α except that t_1 and t_2 are swapped. $\square$

We now have all the lemmas which we need for the case G is the graph θ_0. Next, we consider 2-connected graphs, and finally just connected graphs.

Lemma 2.13

Let G be a 2-connected graph different from a cycle or the graph θ_0, $k_1 + k_2 + \cdots + k_p \leq n(G) - 1$, and $k_1 \geq 2$. Then $T(G; (k_1, k_2, \ldots, k_p))$ is connected.

Proof. It is enough to show the case $k_1 + k_2 + \cdots + k_p = n(G) - 1$ only. By Theorem 2.1, we only have to consider the case that G is bipartite, and we have that $T(G; (k_1, k_2, \ldots, k_p))$ has at most two components. Let α and β be any token configurations in $T(G; (k_1, k_2, \ldots, k_p))$. By considering all tokens are different, let α' and β' be token configurations in $T(G; n(G) - 1)$ induced by α and β, respectively.

Case 1: α' and β' have their unoccupied vertices at odd distance.

If $\alpha'(\beta')^{-1}$ is an odd permutation, we are done. Suppose this is not the case. Let α_0 be the same token configuration as α except that the tokens 1_1 and 1_2 are swapped. Thus α_0 and β are in the same component of $T(G; (k_1, k_2, \ldots, k_p))$. Then we are done since α and α_0 are the same token configuration in $T(G; (k_1, k_2, \ldots, k_p))$.

Case 2: α' and β' have their unoccupied vertices at even distance.

Similar to Case 1, if $\alpha'(\beta')^{-1}$ is an even permutation, then we are done. Otherwise, let α_0 be the same token configuration as α except that the tokens 1_1 and 1_2 are swapped. $\square$

Lemma 2.14

Let G be a 2-connected graph different from a cycle. If $k \leq n(G)-2$, then $T(G;k)$ is connected.

Proof. By Lemma 2.12, we have that $T(\theta_0;(1,1,1,1,1)) = T(\theta_0;5)$ is connected. Then, by Proposition 2.6, we are done if G is the graph θ_0. Now we suppose that G is not θ_0. By Lemma 2.13, $T(G;(\underbrace{2,1,\ldots,1}_{n(G)-2 \text{ terms}}))$ is connected. Then $T(G;k)$ is also connected by Proposition 2.6. □

Lemma 2.15

Let H be a 2-connected graph, and G the graph obtained from H by adding one vertex, which is connected to a vertex of H. Then $T(G;n(G)-2)$ is connected.

Proof. Let uv be the added edge, where $d(u)=1$ and $d(v) \geq 2$, and α any token configuration in $T(G;n(G)-2)$.

Case 1: The graph H is a cycle.

We may assume that in α all tokens are on the cycle (otherwise, we can move some tokens to get it since G is connected). It is enough to show that any two consecutive tokens on the cycle can be swapped. Suppose we want to swap consecutive tokens t_1 and t_2. Then we move the tokens on the cycle until v is unoccupied and is between t_1 and t_2. Note that u is now unoccupied. Then we can easily swap t_1 and t_2 by moving through the vertex u.

Case 2: The graph H is not a cycle.

By Lemma 2.14, $T(H;n(H)-2)$ is connected. Thus any two tokens in $\alpha|_H$ can be swapped. Since H is 2-connected, $d_H(v) \geq 2$. Let x and y be neighbors of v in H. Since G is connected, we may assume that v and x are the two unoccupied vertices in α. We can see that the tokens on u and on y can be swapped by moving through the vertex x. Hence any two tokens in α can be swapped, so we are done. $\square$

Lemma 2.16

Let G_1 and G_2 be 2-connected graphs containing a vertex v_1 and v_2, respectively, and G the graph resulting from combining G_1 and G_2 by identifying the vertices v_1 and v_2. Then $T(G; n(G) - 2)$ is connected.

Proof. Let u_1 be a vertex in G_1 which is adjacent to v_1, and u_2 a vertex in G_2 which is adjacent to v_2. Let v be the vertex in G, which results from identifying the vertices v_1 and v_2. Thus v is adjacent to u_1 and u_2 in G. Let H_1 and H_2 be the subgraphs of G induced by $V(G_1) \cup \{u_2\}$ and $V(G_2) \cup \{u_1\}$, respectively. By Lemma 2.15, $T(H_1; n(H_1) - 2)$ and $T(H_2; n(H_2) - 2)$ are connected. Let α be any token configuration in $T(G; n(G) - 2)$. We can assume that v and u_1 are unoccupied in α; otherwise, we can move some tokens in α to get another token configuration such that v and u_1 are unoccupied. Since $T(H_1; n(H_1) - 2)$ is connected, any two tokens on the vertices of H_1 in α can be swapped. Also, since $T(H_2; n(H_2) - 2)$ is connected, any two tokens on the vertices of H_2 in α can be swapped. Since H_1 and H_2 share the token on u_2, any two tokens in α can be swapped. $\square$

The block-cutvertex graph of a connected graph G, denoted by $bc(G)$, is the graph whose vertices are the blocks and cut-vertices of G, and the edges of $bc(G)$ join cut-vertices with those blocks to which they belong. An endblock of G is a block, which is of degree 1 in $bc(G)$. Thus every endblock contains only one cut-vertex of G.

Lemma 2.17

Let G be a connected graph with connectivity 1, and l a positive integer. If G does *not contain a separating path of size l, then there exists an endblock B in G such that $G\backslash(V(B)\backslash\{v_B\})$ is connected and does not contain a separating path of size l, where v_B is the cut-vertex of G in B.*

Proof. Let B_0 be any endblock in G. Take an endblock B of maximum distance from B_0 in $bc(G)$. Thus $G\backslash(V(B)\backslash\{v_B\})$ is connected, where v_B is the cut-vertex of G in B. Suppose for a contradiction that $G\backslash(V(B)\backslash\{v_B\})$ has a separating path P of size l. Since G has no separating path of size l, v_B is an interior vertex of P. Then there are two P-components. One component contains B_0, and the other contains another endblock, which is farther from B_0 than B is. This is a contradiction. $\square$

We call an endblock B according to Lemma 2.17, a *good endblock.*

Proof of Theorem 2.2. ($\Leftarrow$) It is easy to see that if G is disconnected, $k_1 + k_2 + \cdots + k_p = n(G)$, or G is a path, then $T(G;(k_1, k_2, \ldots, k_p))$ is disconnected. By Lemmas 2.9, 2.10, and Theorem 2.1, we are done when the fourth, fifth, or sixth condition holds.

Next, we suppose that G is a connected graph with connectivity 1, other than a path, $n(G) - (k_1 + k_2 + \cdots + k_p) = l \geq 1$, and G contains a separating path $P = v_1v_2 \ldots v_l$. If $l = 1$, let ${G_1}^*$ be any component of $G\backslash v_1$, and G_1 and G_2 the subgraphs of G induced by $V({G_1}^*)\cup\{v_1\}$ and $V(G)\backslash V({G_1}^*)$, respectively. If $l \geq 2$, let G_1 and G_2 be the two P-components of G such that G_1 contains v_1 and G_2 contains v_l.

We prove the labelled token graph is disconnected by showing that there are two token configurations, which are in different components of $T(G;(k_1, k_2, \ldots, k_p))$.

Case 1: $n(G_1) \geq k_1$.

Let α be a token configuration such that $v_1, v_2, \ldots, v_l$ are all unoccupied, and all the tokens with label 1 are in $G_1 \backslash v_1$. Let u be a vertex such that $\alpha(u) = 1$, and v a vertex in $G_2 \backslash v_l$. Thus $\alpha(v) = c$ for some $c \neq 1$. Let β be the same token configuration as α except that $\beta(u) = c$ and $\beta(v) = 1$. Since the token 1 on u in α cannot be moved to the vertex v (it can go as far as to v_l), β is not reachable from α.

Case 2: $n(G_1) < k_1$.

Let α be a token configuration such that $v_1, v_2, \ldots, v_l$ are all unoccupied, and each vertex in $G_1 \backslash v_1$ contains a token 1. Let $u \in V(G_1) \backslash \{v_1\}$ (so $\alpha(u) = 1$), and v a vertex in $G_2 \backslash v_l$ such that $\alpha(v) = c$ for some $c \neq 1$. Let β be the same token configuration as α except that $\beta(u) = c$ and $\beta(v) = 1$. Since the token c on v in α cannot be moved to the vertex u (it can go as far as to v_1), β is not reachable from α.

($\Rightarrow$) We prove necessity by contrapositive. Suppose that all the conditions 1-7 do not hold. Thus G is connected, $k_1 + k_2 + \cdots + k_p \leq n(G) - 1$, and G is not a path. Lemma 2.9 completes the proof when G is a cycle. If G is the graph θ_0, we have $T(\theta_0; (3, 1, 1, 1))$ and $T(\theta_0; (1, 1, 1, 1, 1))$ are connected by Lemmas 2.11 and 2.12. For the other numbers of tokens, we are done by Properties 2.6 and 2.7. If G is a 2-connected graph, then we are done by Theorem 2.1 and Lemmas 2.13 and 2.14.

We now suppose that G is a connected graph with connectivity 1, which does not contain a separating path of size l, where $l = n(G) - (k_1 + k_2 + \cdots + k_p)$. Since G has connectivity 1, G contains a separating path of size 1. Thus $l \geq 2$. We will prove that $T(G; n(G) - l)$ is connected by induction on the number of blocks in G, and then we will have $T(G; (k_1, k_2, \ldots, k_p))$ is connected by Proposition 2.7.

Base Step: There are two blocks in G.

Since G is not a path, G has to be a graph in Lemma 2.15 or 2.16. Then $T(G; n(G)-2)$ is connected. Since $l \geq 2$, $T(G; n(G)-l)$ is also connected.

Induction Step: There are $m \geq 3$ blocks in G.

Let B be a good endblock in G and H the subgraph of G induced by $[V(G) \backslash V(B)] \cup \{v_B\}$, where v_B is the cut-vertex of G in B. By Lemma 2.17, we have that H is connected and has no separating path of size l. It is easy to see that H has $m-1 \geq 2$ blocks, so H has connectivity 1. Then by the induction hypothesis, $T(H; n(H)-l)$ is connected.

Let v'_B be a vertex of H which is adjacent to v_B, and B' the subgraph of G induced by $V(B) \cup \{v'_B\}$. By Lemma 2.15, $T(B'; n(B')-2)$ is connected, so $T(G; n(B')-2)$ is connected by Proposition 2.8.

Case 1: $n(H) \leq l$.

Then $n(G)-l = (n(H)+n(B')-2)-l \leq (l+n(B')-2)-l = n(B')-2$. Therefore, $T(G; n(G)-l)$ is connected by Proposition 2.6.

Case 2: $n(H) \geq l+1$.

We claim that H is not a path. Indeed, if H is the path $v_1 v_2 \cdots v_{l+1} \cdots v_m$ for some $m \geq l+1$, then $v_2 v_3 \cdots v_{l+1}$ is a separating path of size l in G, which is a contradiction.

If $d_H(v_B) \geq 2$, choose $u = v_B$, $x = v'_B$, and y another neighbour of v_B in H. Otherwise, let u be the vertex nearest to v_B in H such that $d_H(u) \geq 3$ (u exists since H is not a path). Then there is only one path from u to v_B, and it is of size less than l since G does not contain a separating path of size l. Then let x and y be neighbours of u in H which are farther from v_B than u is.

Let α be any token configuration in $T(G; n(G) - l)$. We can assume that in α, there are $n(B') - 2$ tokens on B' such that v_B and v'_B are unoccupied, and there are $n(H) - l \geq 1$ tokens on H such that x and the vertices on the path from u to v_B are all unoccupied while the vertex y is occupied. Otherwise, we can always move some tokens to get a token configuration with such properties from α since G is connected. Since $T(B'; n(B') - 2)$ and $T(H; n(H) - l)$ are connected, any two tokens on B' and any two tokens on H can be improperly swapped in α. So we need to show only that in α there is a token on B' and a token on H which can be swapped.

Let t_1 be the token which occupies y, and t_2 the token on B' which is on a neighbour of v_B. We now can improperly swap the tokens t_1 and t_2 by moving through the vertex x. This completes the proof of Theorem 2.2. $\square$

We go back to Lemma 2.14 now. The proof of this lemma as previously given uses the result of Theorem 2.1, which was proved by using algebraic methods. Here we give another proof, using graph theoretical methods only.

Another Proof of Lemma 2.14. We prove the lemma by using Lemma 2.5. Let G be a 2-connected graph different from a cycle, α any token configuration with at most $n(G) - 2$ different tokens, v any unoccupied vertex in α, and u any occupied vertex in α, say u contains a token t. We will show that in α, the tokens t and ϕ (the empty token on v) can be improperly empty-token swapped.

Since G is 2-connected, there are two internally disjoint paths P_1 and P_2 between u and v. If one of these two paths does not contain any token, then we can easily move t from u to v, so we are done. Suppose this is not the case. Since $k \leq n(G) - 2$, there is an unoccupied vertex w such that $w \neq v$. Let s_1 and s_2 be the tokens which are nearest to v on P_1 and P_2, respectively.

Case 1: There is a vertex of G which is not on $P_1 \cup P_2$.

Subcase 1.1: The vertex w is not on $P_1 \cup P_2$.

Take a path P_w from w to some vertex x on $P_1 \cup P_2$. Let y be the vertex on P_w which is adjacent to x. Then do the following (if necessary),

(1) move the tokens on P_w forward to w to make y unoccupied,

(2) move the tokens on $P_1 \cup P_2$ in anticlockwise direction until t is at x,

(3) move t from x to y along P_w,

(4) move the tokens on $P_1 \cup P_2$ in anticlockwise direction until x is between the tokens s_1 and s_2,

(5) move t from y back to x along P_w,

(6) move the tokens on $P_1 \cup P_2$ in clockwise direction until they are at their destination positions,

(7) move the tokens which we moved in (1) back to their initial positions.

Subcase 1.2: The vertex w is on $P_1 \cup P_2$, but there is another vertex z which is not on $P_1 \cup P_2$.

We can suppose that the vertices which are not on $P_1 \cup P_2$ are all occupied by tokens. Otherwise, we are done by Subcase 1.1. Take a path P_z from z to some vertex x on $P_1 \cup P_2$. Let s be the token on P_z which is the nearest to x but not on x. To use Subcase 1.1, we will move s to the vertex w.

We do the following (if necessary),

(1) move the tokens on $P_1 \cup P_2$ in anticlockwise direction until the free space on w arrives at x,

(2) move s to x along P_z,

(3) move the tokens on $P_1 \cup P_2$ in clockwise direction until s arrives at w.

Now we can use Subcase 1.1 to move t from u to v, and then we move the token s back to its initial position.

Case 2: All vertices are on $P_1 \cup P_2$.

Since G is not a cycle, there exists a chord xy on $P_1 \cup P_2$ for some $x, y \in V(G)$. We move the tokens on $P_1 \cup P_2$ in anticlockwise direction until the token t arrives at x or the empty token ϕ (on v initially) arrives at y. At the time of arrival, we may assume that t is on a vertex z_1, and ϕ is on a vertex z_2. If $z_1 = x$ and $z_2 = y$, we can easily move t from x to y along the chord, and then move the tokens on $P_1 \cup P_2$ in clockwise direction until we get the wanted token configuration. Otherwise, we can find two internally disjoint paths P_1' and P_2' from z_1 to z_2 such that there is another vertex which is not on $P_1' \cup P_2'$. By Case 1, we can move t from z_1 to z_2. Finally, we move the tokens on $P_1 \cup P_2$ in clockwise direction until t arrives at v, and the other tokens are back to their initial positions. $\square$

2.4 Proof of Theorem 2.3

Note that in this section, we use some terminology and notation from the previous section. According to the graph θ_0 in Figure 2.1, we denote a token configuration α of θ_0 which has $\alpha(v_i) = t_i$ for all $i = 0, 1, \ldots, 6$, by $[t_0, t_1, t_2, t_3, t_4, t_5, t_6]$. For example, if α is the top left token configuration in Figure 2.4, then $\alpha = [2, \phi, 2, 3, 3, 1, 1]$.

Let G be a connected graph with connectivity 1, $n(G) - (k_1 + k_2 + \cdots + k_p) = 1$, and v_B a cut-vertex of a block B in G. Given a token configuration α, we can see that even though α_{v_B} is not unique, $\tau(\alpha_{v_B}|_B)$ is unique. This means tokens in $\tau(\alpha_{v_B}|_B)$ can be moved to vertices in B only. We get the following lemma.

Lemma 2.18

Let G be a connected graph with connectivity 1, $n(G) - (k_1 + k_2 + \cdots + k_p) = 1$, and α and β token configurations of G. Then α is reachable from β if and only if $\alpha_{v_B}|_B$ is reachable from $\beta_{v_B}|_B$ for every block B in G.

Proof of Theorem 2.3. ($\Leftarrow$) It is easy to see that α and β are in the same component of $T(G; (k_1, k_2, \ldots, k_p))$ if one of the first three conditions holds.

Suppose that G is the graph θ_o and $(k_1, k_2, \ldots, k_p) = (2, 2, 2)$. We have proved that $T(\theta_0; (2, 2, 2))$ is not connected and has 2 components. One component contains 60 standard token configurations, shown in Figure 2.4 (and some other token configurations). We can see that all the (1,1)-standard token configurations of which the cyclic order of tokens on the lower 5-cycle is $(2, 2, 3, 3)$ are in this component. The other component contains 30 standard token configurations, shown in Figure 2.5 (and some other token configurations), and all the (1,1)-standard token configurations of which the cyclic order of tokens on the lower 5-cycle is $(2, 3, 2, 3)$ are in this component. Thus, if α' and β' are in the same group (stated in 4(a) in the theorem), then α and β are in the same component of $T(\theta_0; (2, 2, 2))$.

Next, we suppose that $(k_1, k_2, \ldots, k_p) = (2, 2, 1, 1)$. We relabel the two tokens with label 3 in all standard token configurations in Figure 2.4 into two possible ways, say using labels 3 and 4. Then we get two copies of Figure 2.4. One copy contains the standard token configuration $[2, \phi, 2, 3, 4, 1, 1]$, and the other contains $[2, \phi, 2, 4, 3, 1, 1]$. We do a similar relabelling in Figure 2.5, so we get two copies of Figure 2.5. One copy contains $[2, \phi, 3, 2, 4, 1, 1]$, and the other contains $[2, \phi, 4, 2, 3, 1, 1]$. Next, we prove that the standard token configurations from the two copies of Figure 2.4 are in the same component of $T(\theta_0; (2,2,1,1))$ by showing a path from $[2, \phi, 2, 3, 4, 1, 1]$ to $[2, \phi, 2, 4, 3, 1, 1]$. This path can be obtained by the following sequence of operations: $L, L, U, L, U, L, U, L, U, U, L, L, U, U, U$. Similarly, we give a path from $[2, \phi, 3, 2, 4, 1, 1]$ to $[2, \phi, 4, 2, 3, 1, 1]$ to show that the standard token configurations from the two copies of Figure 2.5 are in the same component. This path can be obtained by the following sequence of operations: $L, U, L, L, U, L, U, L, U, U, L, U, L, L, L, U, U, U, L, U, U, U$. Thus we have two groups of standard token configurations in $T(\theta_0; (2,2,1,1))$ such that standard token configurations which are in the same group are in the same component of $T(\theta_0; (2,2,1,1))$. We can see that one group, called Group A_1, contains all the (1,1)-standard token configurations of which the cyclic order of tokens on the lower 5-cycle is $(2,2,3,4)$ or $(2,2,4,3)$. And the other group, called Group A_2, contains all the (1,1)-standard token configurations of which the cyclic order of tokens on the lower 5-cycle is $(2,3,2,4)$ or $(2,4,2,3)$. Hence, if α' and β' are in the same group (stated in 4(a) in the theorem), then α and β are in the same component of $T(\theta_0; (2,2,1,1))$.

Now we suppose that $(k_1, k_2, \ldots, k_p) = (2,1,1,1,1)$. Then we relabel the two tokens with label 2 in all standard token configurations in Group A_1 into two possible ways, say using labels 2 and 5. So we get two copies of standard token configurations from Group A_1, called Groups B_1 and B_2, and two copies of standard token configurations from Group A_2, called Groups B_3' and B_3''. Without loss of generality, we suppose that Group B_3' contains $[2, \phi, 4, 5, 3, 1, 1]$, and Group B_3'' contains $[5, \phi, 4, 2, 3, 1, 1]$. Then we show that there are three groups of standard token configurations in $T(\theta_0; (2,1,1,1,1))$ such that standard token configurations in the same group are in the same component. For this, it suffices to show a path from $[2, \phi, 4, 5, 3, 1, 1]$ to $[5, \phi, 4, 2, 3, 1, 1]$, obtained by the sequence of operations: $L, U, L, L, U, L, U, L, U, U, L, L, U, U, L, L, L, U, U, U, L, L, L, U, U, U, L, L, U, U, U, L,$

L, L. Then we use B_3 to denote the group formed from the combination of B_3' and B_3''. Again, if α' and β' are in the same group (stated in 4(b) in the theorem), then α and β are in the same component of $T(\theta_0; (2,1,1,1,1))$.

We next suppose that $(k_1, k_2, \ldots, k_p) = (1,1,1,1,1,1)$. By relabelling the two tokens with label 1 in all standard token configurations in Groups B_1, B_2, and B_3 into two possible ways, say using labels 1 and 6, we get two copies of standard token configurations from each group. So we now have six groups of standard token configurations such that standard token configurations in the same group are in the same component of $T(\theta_0; (1,1,1,1,1,1))$. Similarly, if α' and β' are in the same group (stated in 4(c) in the theorem), then α and β are in the same component of $T(\theta_0; (1,1,1,1,1,1))$. Thus we are now done when G is the graph θ_0.

When G is a 2-connected bipartite graph other than a cycle and there are $n(G)-1$ different tokens, we are done by Theorem 2.1.

Next, we suppose that G is a connected graph with connectivity 1, other than a path, $n(G) - (k_1 + k_2 + \cdots + k_p) = l \geq 2$, $P_1, P_2, \ldots, P_m$ are all the separating paths of size l in G, and $\tau(\alpha_i|_{G_{i,1}}) = \tau(\beta_i|_{G_{i,1}})$ and $\tau(\alpha_i|_{G_{i,2}}) = \tau(\beta_i|_{G_{i,2}})$ for all $i = 1, 2, \ldots, m$. We prove that α and β are in the same component of the labelled token graph by induction on the number of separating paths of size l in G.

Base Step: There is only one separating path $P_1 = v_{1,1}v_{1,2}\ldots v_{1,l}$ of size l in G.

Let $G_{1,1}$ and $G_{1,2}$ be the two P_1-components of G. If $G_{1,1}$ or $G_{1,2}$ is a path, then it has to be a path of length 2; otherwise, there are at least two separating paths of size l in G. Let $G'_{1,1}$ and $G'_{1,2}$ be the subgraphs of G induced by $V(G_{1,1}) \cup \{v_{1,2}, v_{1,3}, \ldots, v_{1,l}\}$ and $V(G_{1,2}) \cup \{v_{1,1}, v_{1,2}, \ldots, v_{1,l-1}\}$, respectively. Then $G'_{1,1}$ and $G'_{1,2}$ do not contain a separating path of size l, $\tau(\alpha_1|_{G'_{1,1}}) = \tau(\beta_1|_{G'_{1,1}})$, and $\tau(\alpha_1|_{G'_{1,2}}) = \tau(\beta_1|_{G'_{1,2}})$. If $G_{1,1}$ is a path of length 2, then there is only one token on

$\alpha_1|_{G'_{1,1}}$ and $\beta_1|_{G'_{1,1}}$, so $\alpha_1|_{G'_{1,1}} = \beta_1|_{G'_{1,1}}$. Obviously, $\alpha_1|_{G'_{1,1}}$ is reachable from $\beta_1|_{G'_{1,1}}$. Suppose $G_{1,1}$ is not a path, so $G'_{1,1}$ is not a path. Since $l \geq 2$, $G'_{1,1}$ cannot be a cycle, the graph θ_0, or a 2-connected bipartite graph. Since $G'_{1,1}$ does not contain a separating path of size l, its labelled token graph is connected by Theorem 2.2. Thus $\alpha_1|_{G'_{1,1}}$ is reachable from $\beta_1|_{G'_{1,1}}$. Similarly, $\alpha_1|_{G'_{1,2}}$ is reachable from $\beta_1|_{G'_{1,2}}$. We can conclude that α_1 is reachable from β_1, so α is reachable from β.

Induction Step: There are $m \geq 2$ separating paths of size l in G.

For each $i = 1, 2, \ldots, m$ let $P_i = v_{i,1}v_{i,2}\ldots v_{i,l}$, $G_{i,1}$ and $G_{i,2}$ the two P_i-components, and $G'_{i,1}$ and $G'_{i,2}$ the subgraphs of G induced by $V(G_{i,1}) \cup \{v_{i,2}, v_{i,3}, \ldots, v_{i,l}\}$ and $V(G_{i,2}) \cup \{v_{i,1}, v_{i,2}, \ldots, v_{i,l-1}\}$, respectively. Then there exists a separating path P_j for some j such that one of $G'_{j,1}$ and $G'_{j,2}$ does not contain a separating path of size l. Without loss of generality, we may assume that $j = 1$ and $G'_{1,1}$ does not contain a separating path of size l. By the same arguments used in the base step, we can have that $\alpha_1|_{G'_{1,1}}$ is reachable from $\beta_1|_{G'_{1,1}}$.

Let H be the subgraphs of G induced by $V(G)\backslash(V(G_{1,1})\backslash\{v_{1,1}\})$. Thus there are $m-1$ separating paths of size l in H. By the induction hypothesis, we have that $\alpha_1|_H$ is reachable from $\beta_1|_H$. We can conclude that α_1 is reachable from β_1, so α is reachable from β.

Finally, we suppose that the condition 7 of the theorem holds. By using the previous cases, we can conclude that $\alpha_{v_B}|_B$ is reachable from $\beta_{v_B}|_B$ for every block B in G. Hence α is reachable from β by Lemma 2.18.

($\Rightarrow$) We prove necessity by contrapositive. Suppose that all the conditions 1-7 do not hold. We then show that α and β are in different components of the labelled token graph. It is easy to get the result if G is a path or a cycle.

First, we suppose G is the graph θ_0. Since $T(G;(k_1,k_2,\ldots,k_p))$ is not connected, $(k_1,k_2,\ldots,k_p)$ is (2,2,2), (2,2,1,1), (2,1,1,1,1), or (1,1,1,1,1,1) by Theorem 2.2. As we described previously, two standard token configurations are in the same component if and only if one configuration can be obtained from the other by a finite sequence of operations U and L only. By applying the operations U and L to each standard token configuration in each group (the groups we constructed previously such as Groups A_1 and A_2), we always get another token configuration in the same group. Thus, if α' and β' are in different groups (the groups we stated in the condition 4 of the theorem such as Groups a_1 and a_2), then α and β are in different components of the labelled token graph.

When G is 2-connected, we are done by Theorem 2.1 and Lemmas 2.13 and 2.14.

Next, we suppose that G is a connected graph with connectivity 1, other than a path, $n(G)-(k_1+k_2+\cdots+k_p)=l\geq 2$, and there is a separating path $P_1=v_{1,1}v_{1,2}\ldots v_{1,l}$ such that $\tau(\alpha_1|_{G_{1,1}})\neq\tau(\beta_1|_{G_{1,1}})$ or $\tau(\alpha_1|_{G_{1,2}})\neq\tau(\beta_1|_{G_{1,2}})$. Without loss of generality, we may assume that $\tau(\alpha_1|_{G_{1,1}})\neq\tau(\beta_1|_{G_{1,1}})$. Hence there is a token t which appears in $\alpha_1|_{G_{1,1}}$ more often than in $\beta_1|_{G_{1,1}}$. To get β_1 from α_1, we need to move at least one token t on $G_{1,1}$ in α_1 into some vertex in $V(G_{1,2})\backslash\{v_{1,l}\}$. This is impossible because of the separating path P_1, so β_1 is not reachable from α_1. Hence α and β are in different components of the labelled token graph.

Finally, we suppose that G is a connected graph with connectivity 1, other than a path, $n(G)-(k_1+k_2+\cdots+k_p)=1$, and there is a block B such that $\tau(\alpha_{v_B}|_B)\neq\tau(\beta_{v_B}|_B)$ or it does not satisfy any cases in the condition 7.

Case 1: $\tau(\alpha_{v_B}|_B)\neq\tau(\beta_{v_B}|_B)$.

Then there is a token t which appears in $\alpha_{v_B}|_B$ more often than in $\beta_{v_B}|_B$. To get β_{v_B} from α_{v_B}, we need to move at least one token t on B in α_{v_B} into some vertex in $V(G)\backslash V(B)$, which is impossible. Hence β_{v_B} is not reachable from α_{v_B}, so α and β are in different components

Case 2: None of the cases 7(a), 7(b), 7(c), or 7(d) holds.

We may assume that $\tau(\alpha_{v_B}|_B) = \tau(\beta_{v_B}|_B)$; otherwise, we are done by Case 1. If B is a path, it has to be of length 2 since it is a block. Then there is only one token in $\alpha_{v_B}|_B$, so $T(B; \tau(\alpha_{v_B}|_B))$ is connected. This contradicts the assumption that B does not satisfy any cases in the condition 7. Thus B is a block different from a path. By using the previous cases, we can conclude that $\alpha_{v_B}|_B$ is not reachable from $\beta_{v_B}|_B$. Thus α is not reachable from β by Lemma 2.18. $\square$

Now, we go back to Example 1.1. According to Figure 1.1, is it possible to reach β from α? To get the answer, we first transform the 15-Puzzle to the grid graph $P_4 \times P_4$. Then we get the corresponding token configurations α' and β' of α and β, respectively, shown in Figure 2.6.

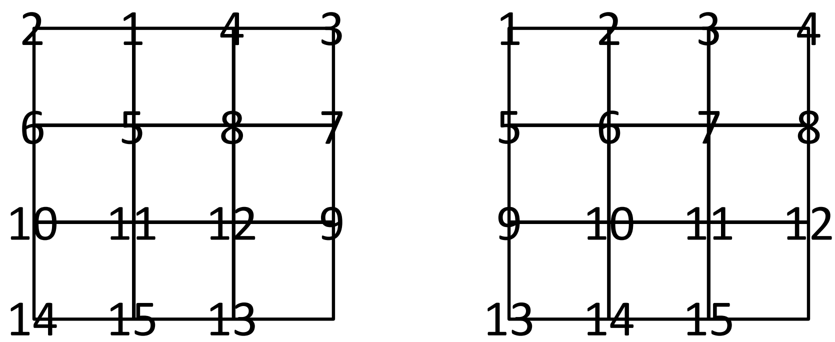


FIGURE 2.6: Corresponding token configurations α' and β'

We can see that α' and β' have empty spaces at distance zero, and $\alpha'(\beta')^{-1} = (1\ 2)(3\ 4)(5\ 6)(7\ 8)(9\ 10)(9\ 11)(9\ 12)(13\ 14)(13\ 15)$ is an odd permutation. By

Theorem 2.3, since $P_4 \times P_4$ is a 2-connected bipartite graph different from a cycle, α' and β' are in different components of the labelled token graph. Thus we cannot go from α to β.

Chapter 3

Miscellaneous Results on Labelled Token Graphs

3.1 Complexity of Finding Shortest Paths in Labelled Token Graphs

In this section, we consider the computational hardness of finding the shortest path between any two token configurations in labelled token graphs.

Goldreich [15] proved that the problem of finding the shortest path between two token configurations in labelled token graphs such that all the tokens are different is NP-hard by reducing 3-Exact-Cover to it.

Papadimitriou et al. [21] proved that the problem of finding the shortest sequence of moves to move the special token from one vertex to another vertex is NP-hard (the other tokens are considered as just obstacles). They proved by reducing from the problem 3-SAT in which each literal occurs at most twice.

We now introduce the following problem, which is called the *Shortest-Token-Move-Sequence (STMS)* problem:

Input: A connected graph G, a multiset of tokens $(k_1, k_2, \ldots, k_p)$, two token configurations α and β, and a positive integer N;

Question: Is there a path of length at most N from α to β in the labelled token graph $T(G; (k_1, k_2, \ldots, k_p))$?

Theorem 3.1

Restricted to the case that the multiset of tokens is (k) *(i.e., all tokens are the same), the problem* STMS *is in* P*.*

Proof. Given two token configurations α and β of k identical tokens on G, let $U = \{u_1, u_2, \ldots, u_k\}$ be the set of vertices containing a token in α, and $V = \{v_1, v_2, \ldots, v_k\}$ be the one in β. We form a complete bipartite graph $K_{k,k}$ with the parts U and V. We define the weight of each edge u_iv_j to be the length of a shortest path from u_i to v_j in G. It is well-known that a perfect matching with minimum weight can be found in polynomial time (for instance, using the Hungarian method; see, e.g., Schrijver [24]). Let M be such a minimum-weight perfect matching.

Without loss of generality, we may assume that $M = \{u_1v_1, u_2v_2, \ldots, u_kv_k\}$. For each i, let P_i be the path corresponding to the edge u_iv_i, w_i the weight of u_iv_i, and W the total weight in M. It is obvious that to get β from α, we need at least W steps. Next, we show that we can find a path from α to β in the labelled token graph of length exactly W by induction on the total weight W.

Base Step: $W = 0$.

Then $U = V$, so α and β are the same token configuration. Thus there is of course a path of length 0 from α to β in the labelled token graph.

Induction Step: $W \geq 1$.

Then there is an edge in M having non-zero weight. Without loss of generality, we may assume that the edge u_1v_1 has weight $w_1 > 0$ (i.e., $u_1 \neq v_1$).

Case 1: $V(P_1) \cap U = \{u_1\}$.

Then we move the token from u_1 along P_1 to v_1. Let α' be the token configuration we just got and M' the perfect matching obtained from M which corresponds to α' and β. We can see that M' has minimum weight in this new situation (otherwise, M is not a minimum weight perfect matching), and the total weight of M' is $W - w_1$. By the induction hypothesis, there is a path of length $W - w_1$ in the labelled token graph from α' to β. Combining the paths from α to α' and from α' to β gives us the wanted path of length W.

Case 2: $[V(P_1) \cap U] \backslash \{u_1\} \neq \phi$.

Without loss of generality, we may assume that u_2 is the vertex in $[V(P_1) \cap U] \backslash \{u_1\}$ nearest to v_1. Note that the path from u_1 to u_2 along P_1 and the path P_2 have the vertex u_2 in common only. Otherwise, there are paths from u_1 to v_2 and from u_2 to v_1 that give another perfect matching having smaller weight than M.

Subcase 2.1: $u_2 \neq v_1$.

Then we define the new paths P_{12} and P_{21} as follows. Let P_{12} be the path formed by going from u_1 along P_1 to u_2 and then continue along P_2 to v_2 (since the path from u_1 to u_2 along P_1 and the path P_2 intersect only at u_2, P_{12} is a path), while P_{21} is just the path from u_2 along P_1 to v_1. It is clear that the sum of the lengths of P_{12} and P_{21} is the same as the sum for P_1 and P_2. We now form a new perfect matching with the minimum weight by removing the edges u_1v_1 and u_2v_2 and adding the edges u_1v_2 and u_2v_1, corresponding to the paths P_{12} and P_{21}, respectively. We can see that u_2 is the only vertex in $V(P_{21}) \cap U$. Since $u_2 \neq v_1$, the edge u_2v_1 has non-zero weight. Then we just follow the steps in Case 1 by moving the token from u_2 along P_{21} to v_1.

Subcase 2.2: $u_2 = v_1$.

Thus $u_2 \neq v_2$. Again, if $V(P_2) \cap U = \{u_2\}$, we are done by following the steps in Case 1. If it is not the case, then we follow the steps in Case 2. We may assume that u_3 is the vertex in $[V(P_2) \cap U] \backslash \{u_2\}$ nearest to v_2. Then P_3 cannot contain u_1 or u_2; otherwise, we can find another perfect matching having smaller weight than M. If $u_3 \neq v_2$, we are done using Subcase 2.1. Otherwise, we continue with u_4. We keep doing this until we get some $l \in \{1, 2, \ldots, k\}$ such that $V(P_l) \cap U = \{u_l\}$ and $u_l \neq v_l$. This must happen since U is finite. Then we finish the proof by following the steps in Case 1 (move the token from u_l to v_l along P_l). □

Recall that Goldreich [15] proved *STMS* is *NP*-complete for the case Wilson considered (i.e., the tokens are all different). So somewhere between all tokens the same and all tokens different, the problem switches from being in *P* to being *NP*-complete. The following theorem shows that it happens as soon as not all tokens are identical. We prove this by applying the ideas of the proof of Theorem 1 of Papadimitriou et al. [21] by reducing from the problem 3-SAT. In their proof, they only move the unique special token from one vertex to the destination vertex and do not mind where the other tokens are at the end. In our proof, at the end we need to move the other tokens to their destinations as well.

Theorem 3.2

Restricted to the case that there is one special token and the other tokens are all identical, the problem STMS *is* NP-*complete.*

Proof. Recall that Kornhauser et al. [18] proved that the problem to decide if there is a path between any two token configurations, is decidable in polynomial time. They also showed that at most $O((n(G))^3)$ moves are needed between any two reachable token configurations. This shows that *STMS* is in *NP*.

To prove that the problem *STMS* is *NP*-hard, we use a reduction from *3-SAT*. Given an instance of a *3-SAT* Boolean formula which is in conjunctive normal form, we construct a graph G, token configurations α and β, and choose a number N as follows (see Figures 3.1 and 3.2 for an example).

The graph G consists of p variable gadgets (the cycles on the left), where p is the number of variables in the formula, q clause gadgets (the vertices on the right), where q is the number of clauses in the formula, a destination vertex (the rightmost vertex) called v_2, and some paths starting from vertices on variable gadgets.

Let M be the maximum number of occurrence of any literal in the formula. For each variable x, we form a variable gadget by two paths corresponding to each variable and its negated form, called P_x and $P_{\bar{x}}$. The path P_x contains vertices $x_1, x_2, \ldots, x_M$ which separate the path into $M+1$ parts, each of length $3r$, where $r = c(p+q)$ for some positive integer c so that $2r^2 > 5Mrp + 3rp + 5rq + 3r$ (c is large enough). Similarly, the path $P_{\bar{x}}$ contains vertices $\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_M$ which separate the path into $M+1$ parts, each of length $3r$. Since M is the maximum number of the number of occurrence of each literal in the formula, x and $\bar{x}$ occur in at most M clauses. For each appearance of x in the clauses, we form a path of length $2r$ from a vertex in $\{\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_M\}$ to the corresponding clause gadget. For each vertex in $\{\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_M\}$ which does not have a path to any clause gadget, we construct a path of length r from it to a new vertex. We create M paths starting from the vertices on $P_{\bar{x}}$ so that they are mutually disjoint. Similarly, we form M disjoint paths starting from the vertices on P_x.

We construct token configurations α and β of G corresponding to the given formula in similar way as the example shown in Figures 3.1 and 3.2 (this example corresponds to the formula $(x \vee y \vee \overline{z}) \wedge (\overline{x} \vee y \vee z) \wedge (\overline{x} \vee \overline{y} \vee z))$, and choose

$N = 5Mrp + 3rp + 5rq + 3r$. Note that the special token has label t, and all the other tokens are identical.

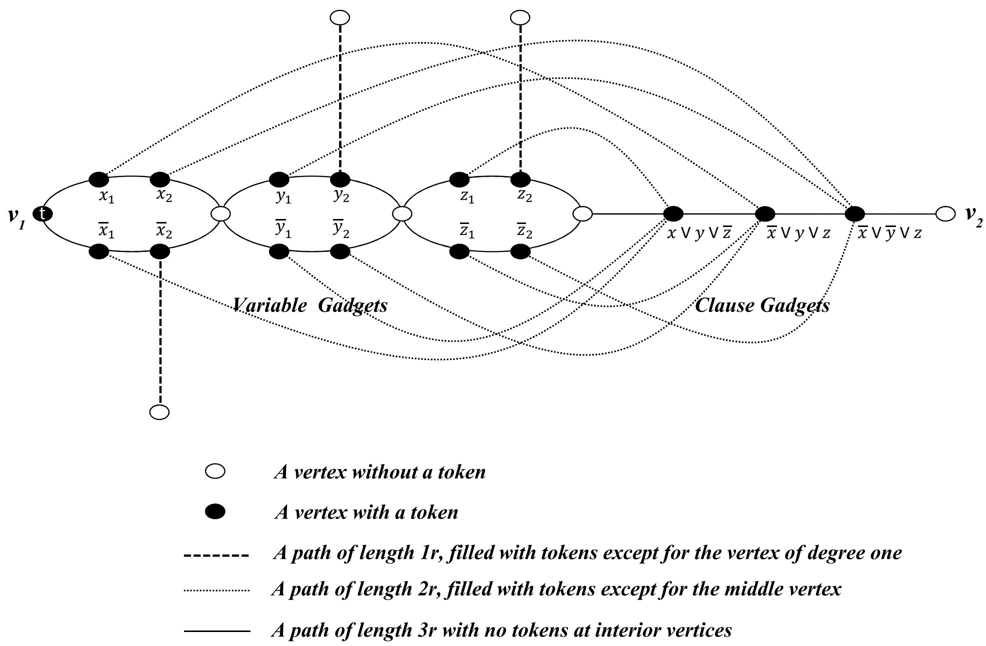


FIGURE 3.1: Starting token configuration α of G, corresponding to the formula $(x \vee y \vee \overline{z}) \wedge (\overline{x} \vee y \vee z) \wedge (\overline{x} \vee \overline{y} \vee z)$

Now, we claim that the formula is satisfiable if and only if we can reach β from α within N steps. First, we suppose that the formula is satisfiable. We will show that within N steps, we can move the token t from the vertex v_1 to the vertex v_2, passing through all the variable gadgets and the clause gadgets, and finally move the other tokens to their final positions.

Consider a variable x, and suppose x is set to be "true" in a feasible assignment. We will move the token t through the variable gadget of x along the path P_x. We need r steps for each token on P_x to be cleared out by pushing the tokens to the end of the r-length paths or to the middle of the $2r$-length paths. Note that

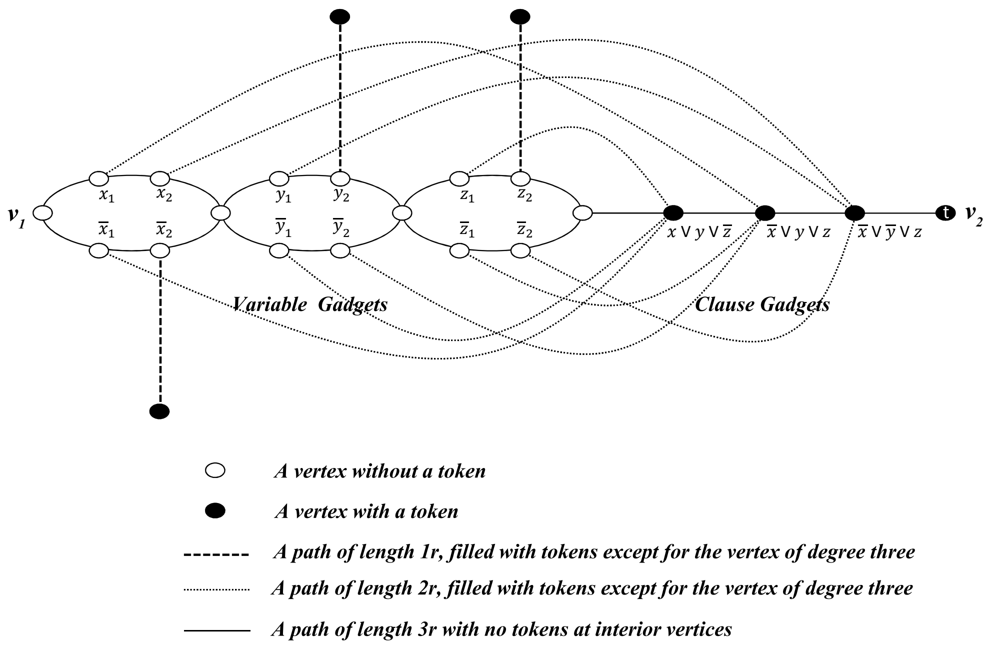


FIGURE 3.2: Final token configuration β of G, corresponding to the formula $(x \vee y \vee \overline{z}) \wedge (\overline{x} \vee y \vee z) \wedge (\overline{x} \vee \overline{y} \vee z)$

any other way requires at least $3r$ steps. If x is "false", we do the same via the path $P_{\overline{x}}$.

Next, we will move t through each clause gadget occupied by a token initially. Since every clause is satisfied, at least one of the three $2r$-length paths connected to each clause gadget has the middle vertex still unoccupied. Then we can move the token on each clause gadget off in r steps by pushing the tokens to the middle on this path, and then t can pass that clause gadget. Hence we can bring the token t from v_1 to v_2 in $[Mrp + (M+1)(3r)p] + [rq + (q+1)(3r)]$ steps, and we need $rq + Mrp$ steps to move the other tokens to their final destinations. Therefore, we can get β from α in exactly N steps.

Conversely, suppose that for every choice of the variables to be true or false, at least one of the clauses is not satisfied. In the step of moving the token on this clause gadget off from the route of t, we need at least $2r$ steps because the middle vertices of all the three $2r$-length paths ending at that clause gadget are no longer empty. Thus the total number of steps to get β from α is more than N steps. Note that to let the token t make a short cut along a path packed with tokens, we need at least $2r^2 > N$ steps.

Since the reduction from the given instance of a *3-SAT* Boolean formula to the graph G, token configurations α and β, and integer N can be done in polynomial time, we can conclude that *STMS* is *NP*-hard, and therefore is *NP*-complete. $\square$

Corollary 3.3

The problem STMS *remains* NP*-complete for any other multiset of tokens which does not restrict the total number of tokens and allow at least two types of tokens.*

3.2 Connectivity of Labelled Token Graphs and Forbidden Minors

In this section, we give another way to consider whether the labelled token graph of a given graph G with a multiset of tokens $(k_1, k_2, \ldots, k_p)$ is disconnected by considering the existence of some forbidden minors. The details can be found in the following theorem.

Theorem 3.4

For each multiset of tokens $(k_1, k_2, \ldots, k_p)$*, there exists a* <u>*finite*</u> *collection of connected graphs* $H_1, H_2, \ldots, H_n$*, such that for each connected graph* G *with* $n(G) > k_1 + k_2 + \cdots + k_p$ *we have*

$T(G;(k_1,k_2,\ldots,k_p))$ is disconnected $\Leftrightarrow$ G has none of $H_1,H_2,\ldots,H_n$ as a minor.

As a consequence, for each multiset $(k_1,k_2,\ldots,k_p)$, there exists a polynomial time algorithm (in $n(G)$) to decide if the labelled token graph of a given graph is disconnected.

Below, we give some theorems which will be used in the proof of Theorem 3.4. They are well-known and can be found, e.g., in [10].

Proposition 3.5

A relation $\leq$ is a well-quasi-ordering on a set X if and only if it is a quasi-ordering without infinite descent and without infinite antichains on elements of X.

Theorem 3.6 (Robertson & Seymour [23])

The minor ordering is well-quasi-ordered on the class of finite graphs.

Proposition 3.7

Let $\leq$ be a well-quasi-ordering on a set X, and P a $\leq$-closed property on elements of X. Then the set of P-forbidden elements is finite.

By Proposition 3.5 and Theorem 3.6, we can see that the minor ordering is a quasi-ordering without infinite descent and without infinite antichains on finite graphs. Since the class of finite connected graphs is a subset of the class of finite graphs, there is neither infinite descent nor infinite antichains on connected graphs under minor ordering. Hence the minor ordering is well-quasi-ordered on the class of finite connected graphs as well. Combining this result with Proposition 3.7 gives the following corollary.

Corollary 3.8

If P is a minor-closed property on connected graphs, then the set of P-forbidden *minors is finite.*

For example, let P be the property of being planar on connected graphs. It is easy to see that P is minor-closed. Hence there are a finite number of P-forbidden minors. According to *Wagner's theorem*, the P-forbidden minors are the complete graph K_5 and the complete bipartite graph $K_{3,3}$.

Theorem 3.9 (Robertson & Seymour [22])
Given a fixed graph H, there exists a polynomial time algorithm to decide if a given graph has H as a minor or not.

Corollary 3.10
If P is a minor-closed property on connected graphs, then there exists a polynomial time algorithm to decide whether a given graph has property P.

Proof of Theorem 3.4. To prove the theorem, it suffices to show that the property of being disconnected of labelled token graphs on connected graphs is closed under minor ordering. We then use Corollary 3.10 to complete the rest of the proof.

Let $(k_1, k_2, \ldots, k_p)$ be a multiset of tokens, G a connected graph, H a connected minor of G, and $k < n(H) \leq n(G)$, where $k = k_1 + k_2 + \cdots + k_p$. Suppose $T(G; (k_1, k_2, \ldots, k_p))$ is disconnected. We will show that $T(H; (k_1, k_2, \ldots, k_p))$ is disconnected by showing the following.

Claim 1: The graph H can be obtained from G by a sequence of edge contractions first and then edge deletions.

For each $X \in V(H)$, let the branch set of X, V_X, be the set of vertices of G, which are contracted to X. Note that the subgraph of G induced by V_X is connected for all $X \in V(H)$, and for all $Y, Z \in V(H)$, $YZ \in E(H)$ if and only if there are $y \in V_Y, z \in V_Z$ such that $yz \in E(G)$. Now let $U = V(G) \setminus \bigcup_{X \in V(H)} V_X$. Choose $v \in U$ such that v is adjacent to some vertex in some branch set. Then we add v to one of these branch sets and keep doing this process until U is empty. We can do this because G is connected and U is finite. Then we can obtain the graph H from G by first contracting all edges in each subgraph induced by each branch set, and then removing all edges which do not appear in H. Note that since G and H are connected, each operation in the sequence to obtain H transforms a connected graph to a connected graph.

Claim 2: Assume F is a connected graph containing an edge uv, $k < n(F)$, and $T(F; (k_1, k_2, \ldots, k_p))$ is disconnected. Then $T(F/uv; (k_1, k_2, \ldots, k_p))$ is disconnected.

Note that F/uv is connected. Let α and β be token configurations of F, which are in different components of $T(F; (k_1, k_2, \ldots, k_p))$. We may assume that in both α and β there is at most one vertex of u and v occupied by a token. Otherwise, since F is connected and $k < n(F)$, we can move some tokens in α (in β) to obtain another token configuration of F in which there is at most one vertex of u and v with a token on it.

Let α' and β' be the token configurations of F/uv induced by α and β, respectively. Then if there is a path from α' to β' in $T(F/uv; (k_1, k_2, \ldots, k_p))$, we can also construct a path from α to β in $T(F; (k_1, k_2, \ldots, k_p))$, which is a contradiction. Hence $T(F/uv; (k_1, k_2, \ldots, k_p))$ is disconnected.

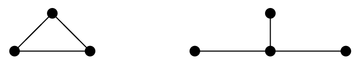
Claim 3: Assume F is a connected graph containing an edge e, $F - e$ is connected, $k < n(F)$, and $T(F; (k_1, k_2, \ldots, k_p))$ is disconnected. Then $T(F - e; (k_1, k_2, \ldots, k_p))$ is disconnected.

Note that each token configuration of F is a token configuration of $F - e$. Since $T(F;(k_1, k_2, \ldots, k_p))$ is disconnected, we can find token configurations α and β of F which are in different components of $T(F;(k_1, k_2, \ldots, k_p))$. Then they are in different components of $T(F - e;(k_1, k_2, \ldots, k_p))$. Hence $T(F - e;(k_1, k_2, \ldots, k_p))$ is disconnected. $\square$

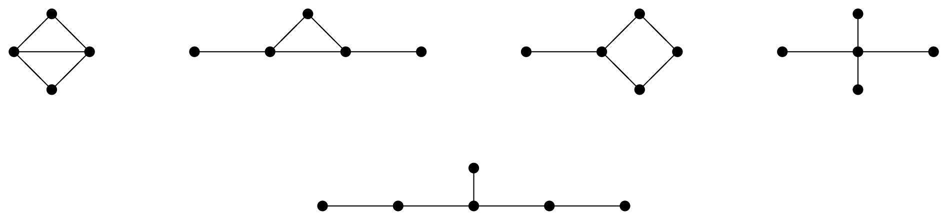
Next, we give some examples of P-forbidden minors, where P is the property of being disconnected of labelled token graphs on connected graphs.

Example 3.11

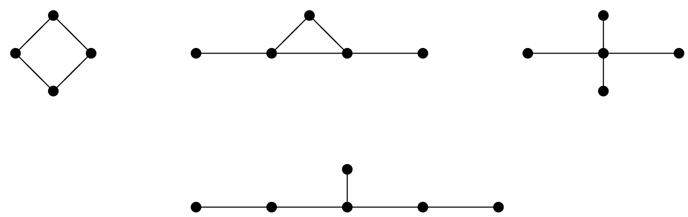
If the multiset of tokens is $(1, 1)$, then there are two P-forbidden minors:



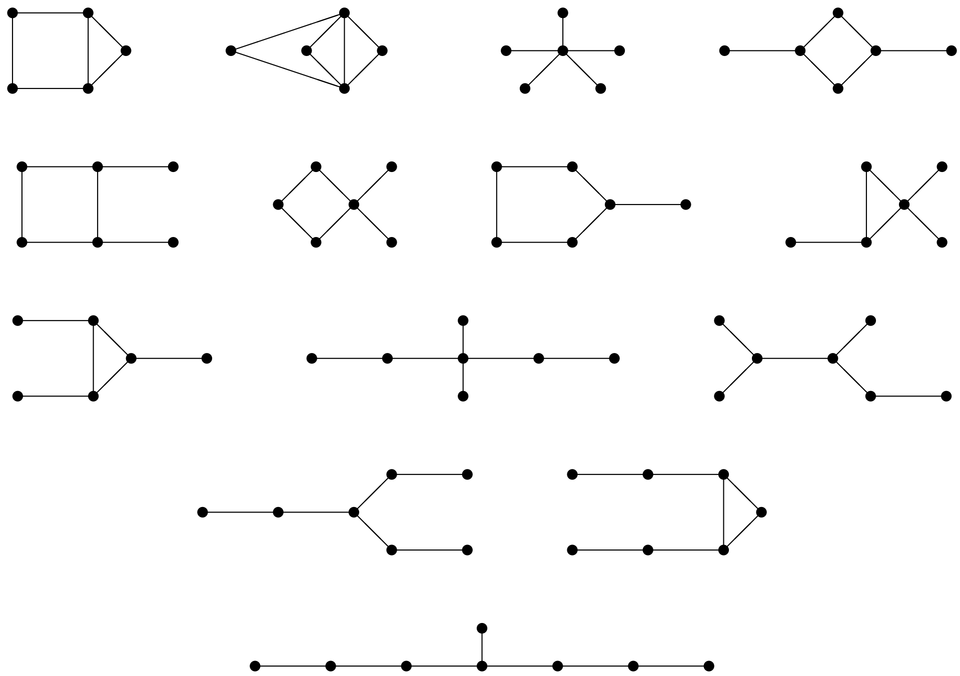
If the multiset of tokens is $(1, 1, 1)$, then there are five P-forbidden minors:



If the multiset of tokens is $(2, 1)$, then there are four P-forbidden minors:



For the case the multiset of tokens is $(1, 1, 1, 1)$, the following graphs are P-forbidden minors, but we are not sure yet if this is the complete list of P-forbidden minors.



Chapter 4

Strong k-Colour Graphs

For a positive integer k and a graph G, the *k-colour graph* of G (introduced by Cereceda et al. [5]) is the graph that has the proper k-vertex-colourings of G as its vertices, and two vertices are joined by an edge if they differ in colour on only one vertex of G. We denote the k-colour graph of G by $C_k(G)$.

We now introduce a subgraph of $C_k(G)$, the *strong k-colour graph* of G, denoted by $S_k(G)$. Its vertex set contains only proper k-colourings in which all k colours actually appear, and we call such a colouring a *strong k-colouring*.

Questions regarding the connectivity of a k-colour graph have applications in reassignment problems of the channels used in cellular networks; see, e.g., [2, 17]. For some applications, it is required that all channels in a range are actually used. Such a labelling is sometimes called a "no-hole" or "consecutive" labelling; see, e.g., [12, 19]. In terms of colourings, this corresponds to a strong k-colouring. And asking questions about the possibility to reassign channels in a cellular network can be done in such a way that all available channels are actually used. These problems can be expressed in finding paths in strong k-colour graphs.

Questions related to the connectivity of a k-colour graph have been studied extensively: [5, 6, 8, 20]. In this note we initiate similar research on the connectivity of strong k-colour graphs: Given a positive integer k and a graph G, is $S_k(G)$ connected? As an example, in Figures 4.1 and 4.2, we show the strong 3-colour graphs of the paths with 4 and 5 vertices, respectively. One is not connected while the other is connected.

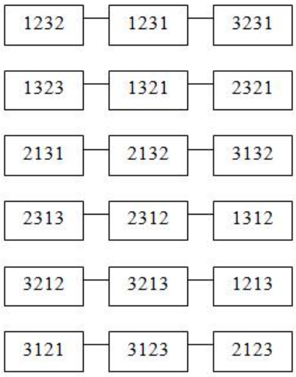


FIGURE 4.1: The strong colour graph $S_3(P_4)$

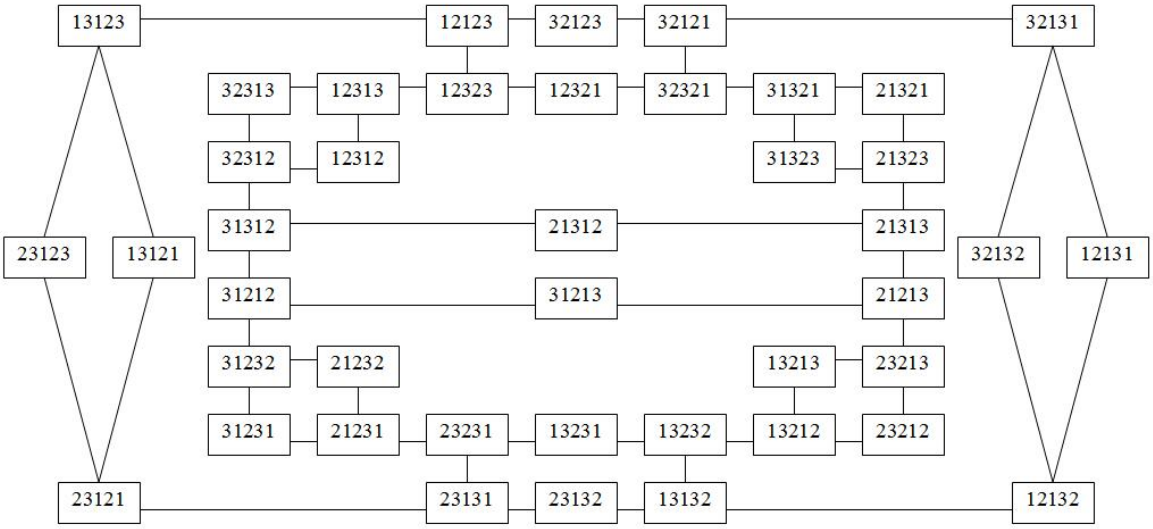


FIGURE 4.2: The strong colour graph $S_3(P_5)$

We usually use lower case Greek letters $\alpha, \beta, \gamma, \ldots$ to denote specific colourings, and lower case Latin $a, b, c, \ldots$ to denote specific colours. And to avoid trivial cases, we will always assume that k is greater than or equal to the chromatic number of G, and the number of vertices of G is at least $k + 1$.

4.1 Literature

The following are known results on k-colour graphs. We make them our goal to study on strong k-colour graphs. However, this thesis contains only some first results on strong k-colour graphs

Cereceda et al. [5] gave some results on the connectivity of k-colour graphs such as if G is a graph with chromatic number $k \in \{2,3\}$, then $C_k(G)$ is not connected. On the other hand, they gave some graphs with chromatic number $k \geq 4$ for which $C_k(G)$ is not connected, and some k-chromatic graphs for which $C_k(G)$ is connected.

In [6], they characterised the bipartite graphs for which $C_3(G)$ is connected. They also proved that the problem to decide the connectedness of the 3-colour graph of a bipartite graph is *coNP*-complete

In [7], they considered the problem of finding a path between two vertex 3-colourings in 3-colour graphs. They proved that it needs polynomial time to decide whether there is a path between them or not. They also showed that if a path exists, the algorithm uses $O(n^2)$ steps.

Bonsma and Cereceda [3] proved that the problem to decide whether there is a path between two vertex k-colourings in k-colour graphs is *PSPACE*-complete for all $k \geq 4$. They also proved that the problem remains *PSPACE*-complete for some specific graphs and the number k such as planar graphs with $4 \leq k \leq 6$, or bipartite planar graphs with $k = 4$.

4.2 General Results on Connectivity of Strong k-Colour Graphs

For all $k \geq 2$ and $m, n \geq 1$, let α be a strong k-vertex-colouring of a complete bipartite graph $K_{m,n}$. Each colour that appears in one part of the partition cannot be used in the other part. Now we choose one colour from each part and recolour the graph by swapping these two colours on each vertex coloured with one of these colours. Let β be the resulting colouring, so β is strong as well. It is easy to see that there is no path in $S_k(K_{m,n})$ from α to β. Thus $S_k(K_{m,n})$ is not connected for all $k \geq 2$ and $m, n \geq 1$. This gives the following lemma.

Lemma 4.1

Let G be a connected, k-colourable graph such that $S_k(G)$ is connected, with $k \geq 2$. Then $|V(G)| \geq k+1$, and G does not contain a complete bipartite graph as a spanning subgraph.

Theorem 4.2

Let G be a connected, k-colourable graph such that $S_k(G)$ is connected, with $k \geq 2$. Suppose the graph G^* is obtained from G by adding a new vertex v^* and joining it to j vertices in $V(G)$, with $1 \leq j \leq k-2$. Then $S_k(G^*)$ is connected.

We will show in the next section that the strong 3-colour graph of the n-vertex path, $S_3(P_n)$, is connected if and only if $n \geq 5$. Now we add a new vertex and join it to the first and the last vertex of the path, forming to an $(n+1)$-vertex cycle. We will show in Section 4.4 that the strong 3-colour graph of the n-vertex cycle, $S_3(C_n)$, is not connected for all n. This example shows that the restriction $j \leq k-2$ of Theorem 4.2 is optimal.

Proof of Theorem 4.2. Let α^* and β^* be strong k-colourings of G^*. We show that there always exists a walk in $S_k(G^*)$ from α^* to β^*. We say that a colouring of G^* is *good* if all k colours appear on $V(G)$.

First, we suppose that α^* and β^* are good. By ignoring the vertex v^*, let α and β be the strong k-colourings of G obtained from α^* and β^*, respectively. Since $S_k(G)$ is connected, there is a path from α to β in $S_k(G)$. We just follow the recolouring steps of that path to form a walk from α^* to β^* in $S_k(G^*)$. The only extra steps happen when we want to recolour a neighbour u of v^* to the same colour as v^*. Since $d_{G^*}(v^*) = j \leq k-2$, we can always recolour v^* to a colour different from any of the colours appearing in its neighbourhood and its current colour. After recolouring v^*, we can recolour u, and continue the walk. This walk in $S_k(G^*)$ finishes in a colouring in which the vertices in $V(G)$ have the same colour as they have in β^*. If necessary, we can do one recolouring of v^* to its colour in β^*, completing the walk in $S_k(G^*)$ from α^* to β^*.

If α^* is not good, then below we show that we can always find a path in $S_k(G^*)$ from α^* to some good colouring (and if necessary, we do the same for β^*). Together with the method described in the previous paragraph, this completes the proof.

So we now assume that in α^* every vertex of G has received one of $k-1$ colours while v^* has the remaining colour. Let W be the set of vertices in $V(G)$ that do not have a unique colour in G for the colouring α^*. Since $|V(G)| \geq k+1$, W is not empty.

Case 1: There is a vertex $w \in W$ not adjacent to v^*.

By the definition of W, there is a vertex $w' \in W$ such that w and w' have the same colour in α^*. Then we recolour w to the same colour as v^*. The resulting colouring is good.

Case 2: All vertices in W are adjacent to v^*.

Additionally, define $U = N(v^*) \backslash W$. and $X = V(G) \backslash N(v^*)$. Note that all vertices in X have a unique colour in α^*.

Subcase 2.1: There is a vertex $x \in X$ which is not adjacent to some vertex $w \in W$.

Again, there is a vertex $w' \in W$ such that w and w' have the same colour in α^*. Then we first recolour w to the same colour as x, and then recolour x to the same colour as v^*. Again, this gives a good colouring.

Subcase 2.2: Every vertex in X is adjacent to every vertex in W.

By Lemma 4.1, U is not empty (otherwise, the pair (X, W) would form the parts of a spanning complete bipartite subgraph of G). Suppose there is some vertex $u \in U$ which is not adjacent to some vertex $w \in W$ and not adjacent to some vertex $x \in X$. Then we can recolour w to the colour of u (this is possible since there is another vertex $w' \in W$ with the same colour as w). Then recolour u to the same colour as x, and finally x to the same colour as v^*. It is easy to check that the remaining colouring is good.

So we are left with the case that each vertex in U is adjacent to every vertex in W or to every vertex in X. Let U_W be the set of vertices in U which are adjacent to every vertex in W, and $U_X = U \setminus U_W$. Then the pair $(X \cup U_W, W \cup U_X)$ forms the parts of a spanning complete bipartite subgraph of G. This case is impossible by Lemma 4.1. $\square$

Theorem 4.3

Let G be a connected k-colourable graph so that $S_k(G)$ is connected, with $k \geq 2$.

Let v be a vertex of G with neighbourhood $N(v)$. Suppose the graph G^* is obtained from G by adding a new vertex v^* and joining v^* to the vertices in N^* for some $N^* \subseteq N(v)$, $N^* \neq \varnothing$. Then $S_k(G^*)$ is connected.

Proof. Let α^* and β^* be strong k-colourings of G^*. We show that there always exists a walk in $S_k(G^*)$ from α^* to β^*. We say that a colouring of G^* is *good* if v and v^* are labelled with the same colour.

First suppose that α^* and β^* are good. By ignoring the vertex v^*, let α and β be the strong k-colourings of G obtained from α^* and β^*, respectively. Since $S_k(G)$ is connected, there is a path from α to β in $S_k(G)$. We just follow the recolouring steps of that path to form a walk from α^* to β^* in $S_k(G^*)$. The only extra step happens when we recolour v. In the next step we immediately recolour v^* to the same colour as v just received. It is easy to check that all these recolourings are allowed and give a walk in $S_k(G^*)$ from α^* to β^*, completing the proof.

Assume that α^* is not good. Below we show that we always can find a path in $S_k(G^*)$ from α^* to some good colouring (and if necessary, we do the same for β^*). Together with the method described in the previous paragraph, this completes the proof.

If there is a vertex $u \in V(G) \setminus \{v\}$ with the same colour as v^*, then we just recolour v^* to the same colour as v. This gives a good colouring.

So we now suppose that in α^* every vertex of G has received one of $k-1$ colours while v^* has the remaining colour. Remind that v and v^* received different colours in α^*. Let W be the set of vertices in $V(G)$ that did not receive a unique colour in G for the colouring α^*. Since $|V(G)| \geq k+1$, W is not empty.

Case 1: There is a vertex $w \in W$ not adjacent to v^*.

By the definition of W, there is a vertex $w' \in W$ such that w and w' have the same colour in α^*. Hence we can recolour w to the same colour as v^*, and then recolour v^* to the same colour as v. The resulting colouring is good.

Case 2: All vertices in W are adjacent to v^*.

Additionally, define $U = N(v^*) \backslash W$. and $X = V(G) \backslash N(v^*)$. Note that all vertices in X have a unique colour in α^*.

Subcase 2.1: There is a vertex $x \in X$ which is not adjacent to some vertex $w \in W$.

Again, there is a vertex $w' \in W$ such that w and w' have the same colour in α^*. Then we first recolour w to the same colour as x. Then recolour x to the same colour as v^*, and finally recolour v^* to the same colour as v. Again, this gives a good colouring.

Subcase 2.2: Every vertex in X is adjacent to every vertex in W.

By Lemma 4.1, U is not empty (otherwise, the pair (X, W) would form the parts of a spanning complete bipartite subgraph of G). Suppose there is some vertex $u \in U$ that is not adjacent to some vertex $w \in W$ and not adjacent to some vertex $x \in X$. Then we can recolour w to the colour of u (this is possible since there is another vertex $w' \in W$ with the same colour as w). Then recolour u to the same colour as x, x to the same colour as v^*, and finally recolour v^* to the same colour as v. It is easy to check that the remaining colouring is good.

So we are left with the case that each vertex in U is adjacent to every vertex in W or to every vertex in X. Let U_W be the set of vertices in U which are adjacent to every vertex in W, and $U_X = U \setminus U_W$. Then the pair $(X \cup U_W, W \cup U_X)$ forms the parts of a spanning complete bipartite subgraph of G. This case is impossible by Lemma 4.1. □

It is easy to see that in a normal colour graph $C_k(G)$, there always is a path from any proper k-vertex-colouring to some strong k-vertex-colouring. This gives the following lemma.

Lemma 4.4

If $S_k(G)$ is connected, then $C_k(G)$ is also connected.

4.3 The Strong k-Colour Graph of Paths

In this section, we prove that the strong k-colour graph of a path with n vertices, $S_k(P_n)$, is connected if and only if $k \geq 3$, $n \geq 5$, and $n \geq k+1$.

First, suppose we colour a path P_n, $n \geq 2$, with two colours. It is easy to see that there are only two strong 2-vertex-colourings of P_n, and they are not adjacent in $S_2(P_n)$. Thus $S_2(P_n)$ is not connected for all $n \geq 2$.

For $k = 3$, we have already seen in Figures 4.1 and 4.2 that $S_3(P_4)$ is not connected, but $S_3(P_5)$ is connected. It is somewhat more work to show that $S_4(P_5)$ is connected.

Proposition 4.5

The strong colour graph $S_4(P_5)$ is connected.

Proof. In any strong 4-colouring of P_5, there are only two vertices with the same colour. Let α be a strong 4-colouring of $P_5 = v_1v_2 \ldots v_5$. We call α an *a-standard colouring* if $\alpha(v_1) = \alpha(v_5) = a$.

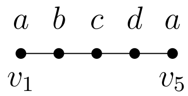
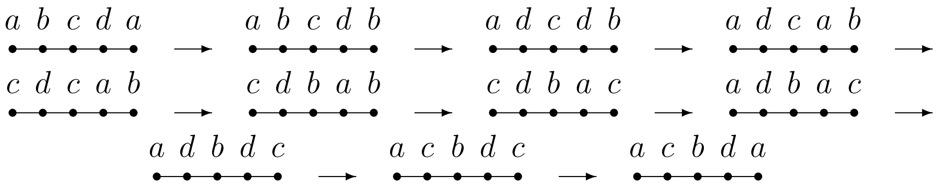


FIGURE 4.3: An a-standard colouring

We will prove the proposition by showing the following three steps.

Step 1: There is a path from (a, b, c, d, a) to any other a-standard colouring.

We will first show that there is a path from (a, b, c, d, a) to (a, c, b, d, a) :

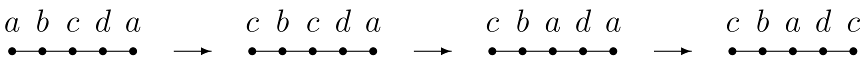


By symmetry, there is a path from (a, b, c, d, a) to (a, b, d, c, a) as well.

Now we consider a-standard colourings as permutations of $\{b, c, d\}$. Note that all these permutations can be generated by the transpositions (b, c) and (c, d). Therefore, since we can find a path from (a, b, c, d, a) to (a, c, b, d, a), and from (a, b, c, d, a) to (a, b, d, c, a), there is a path from (a, b, c, d, a) to any other a-standard colouring.

Step 2: There is a path between any two types of standard colourings.

First, here is a path from an a-standard colouring α to an $\alpha(v_3)$-standard colouring :



Next, a path from an a-standard colouring α to an $\alpha(v_2)$-standard colouring :

$$a\,b\,c\,d\,a \longrightarrow d\,b\,c\,d\,a \longrightarrow d\,b\,c\,b\,a \longrightarrow d\,a\,c\,b\,a \longrightarrow$$
$$d\,a\,c\,b\,d \longrightarrow b\,a\,c\,b\,d \longrightarrow b\,a\,c\,a\,d \longrightarrow b\,d\,c\,a\,d \longrightarrow$$
$$b\,d\,c\,a\,b$$

By symmetry, there is also a path from an a-standard colouring α to an $\alpha(v_4)$-standard colouring.

Step 3: Each strong 4-colouring has a path to some standard colouring.

Let α be a strong 4-colouring of P_5. Then α has one of the following forms:

$$a\,b\,c\,d\,a \qquad a\,b\,c\,a\,d \qquad a\,b\,a\,c\,d \qquad b\,a\,c\,d\,a \qquad b\,a\,c\,a\,d$$
$$b\,c\,a\,d\,a$$

The first form already is an a-standard colouring; for the second and the third ones we just recolour the vertex v_1 to d; while for the fourth and the sixth ones we just recolour the vertex v_5 to b. Finally, for the fifth form the following is a path to a b-standard colouring:

$$b\,a\,c\,a\,d \longrightarrow b\,d\,c\,a\,d \longrightarrow b\,d\,c\,a\,b$$

It is straightforward to see that appropriate renaming of the colours and sequence of the paths in Steps $1-3$ will transform any strong 4-colouring of P_5 into any other strong 4-colouring. □

We extend the last result by showing that $S_k(P_{k+1})$ is connected, for all $k \geq 4$.

Proposition 4.6

For all $k \geq 4$, $S_k(P_{k+1})$ is connected.

Proof. We prove by induction on k. We have already shown the proposition is true for $k = 4$.

Let α and β be strong k-colourings of $P_{k+1} = v_1 v_2 \dots v_{k+1}$, for some $k \geq 5$. We can assume that in α, the vertex v_{k+1} has a unique colour. Otherwise, there is another vertex v_i such that $\alpha(v_i) = \alpha(v_{k+1})$. Then just recolour v_i to a colour different from $\alpha(v_{k+1})$. Without loss of generality, we may assume that this unique colour on v_{k+1} in α is a.

If the vertex v_{k+1} is the only vertex coloured a in β as well, then we can just remove v_{k+1}. Let α' and β' be the strong k-colourings of $P_k = v_1 v_2 \dots v_k$ obtained from α and β, respectively. By the induction hypothesis, $S_{k-1}(P_k)$ is connected. Then there is a path from α' to β' in $S_{k-1}(P_k)$. Using the same steps on P_{k+1} gives a path from α to β in $S_k(P_{k+1})$.

We now assume that in β, v_{k+1} is not coloured a, or it is not the only vertex coloured a. We distinguish 4 cases.

Case 1: In β, v_{k+1} is coloured a, but there is a second vertex v_i coloured a as well.

Then we just recolour v_i to another colour different from a. Thus v_{k+1} is now the only vertex coloured a. We are done by the paragraph above.

Case 2: In β, v_{k+1} and some other vertex v_i have the same colour $b \neq a$, while a third vertex v_j is coloured a.

Subcase 2.1: v_{k+1} and v_j are not adjacent.

Then we just recolour v_{k+1} to a, and we are back to Case 1.

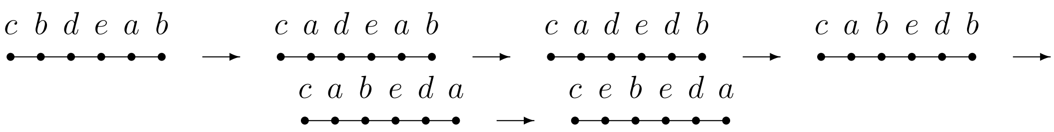
Subcase 2.2: v_{k+1} and v_j are adjacent, i.e., $j = k$.

We call a colouring of $S_k(P_{k+1})$ *good* if we can recolour v_i to a colour which is not in $\{\beta(v_{k-1}), \beta(v_k) = a, \beta(v_{k+1}) = b\}$.

Note that β is good when $k \geq 6$, or $k = 5$ and $i \neq 2$. If β is good, we can recolour v_i to the colour $\beta(v_l)$ for some $l \notin \{i-1, i, i+1, k-1, k, k+1\}$, to obtain the strong k-colouring γ. Let δ be the strong k-colouring of P_{k+1}, obtained from γ by swapping the colours of v_i and v_k. By ignoring the vertex v_{k+1}, we can consider γ and δ as strong $(k-1)$-colourings of P_k. Since $S_{k-1}(P_k)$ is connected, there is a path between these two colourings. We then apply this path to a path in $S_k(P_{k+1})$ from γ to δ.

Next, we will form a path from δ to a colouring in which vertex v_{k+1} is the only vertex coloured a, and then we will have a path from β to this colouring. So we are done by Case 1. In δ, we first recolour v_l to $\beta(v_{k+1}) = b$ and then recolour v_{k+1} to a. Finally, we recolour the vertex v_i (which is previously coloured a) to another colour.

We now suppose that β is not good, i.e., $k = 5$ and $i = 2$. Then here is a path from β to a colouring in which v_{k+1} is the only vertex coloured a.



Case 3: In β, v_i and v_j are coloured a for some $i, j \neq k+1$.

Without loss of generality, we may assume that v_i is not adjacent to v_{k+1}. Then we recolour v_i to $\beta(v_{k+1})$, and we are back to Case 2.

Case 4: In β, v_i and v_j have the same colour $b \neq a$, a third vertex v_ℓ is coloured a, for some $i, j, \ell \neq k+1$.

Without loss of generality, we may assume that v_i is not adjacent to v_{k+1}. Then we recolour v_i to $\beta(v_{k+1})$, and we are back to Case 2. □

By combining these propositions, we get the promised result on the strong colour graph of paths.

Theorem 4.7

The strong colour graph $S_k(P_n)$ is connected if and only if $k \geq 3$, $n \geq 5$ and $n \geq k+1$.

Proof. We already have seen that $S_3(P_4)$ and $S_2(P_n)$, $n \geq 3$, are not connected, while $S_3(P_5)$ and $S_k(P_{k+1})$, $k \geq 4$, are connected. Applying Theorem 4.2 completes the proof. □

4.4 The Strong k-Colour Graph of Cycles

In this section we want to show that the strong k-colour graph of a cycle with n vertices, $S_k(C_n)$, is connected if and only if $k \geq 4$, $n \geq 6$ and $n \geq k+1$. Before we prove the theorem, we give some propositions, which will be used in the proof.

To orient a cycle means to orient each edge on the cycle so that a directed cycle is obtained. If C is a cycle, then by $\overrightarrow{C}$ we denote the cycle with one of the two possible orientations. Given a 3-colouring α using colours $\{1, 2, 3\}$, the weight of an edge $e = uv$ oriented from u to v is

$$w(\overrightarrow{uv}, \alpha) = \begin{cases} +1, & \text{if } \alpha(u)\alpha(v) \in \{12,\ 23,\ 31\}; \\ -1, & \text{if } \alpha(u)\alpha(v) \in \{21,\ 32,\ 13\}. \end{cases}$$

The weight $W(\overrightarrow{C}, \alpha)$ of an oriented cycle $\overrightarrow{C}$ is the sum of the weights of its oriented edges.

Proposition 4.8 (Cereceda et al. [5])

Let α be a 3-colouring of a graph G that contains a cycle C. If $W(\overrightarrow{C}, \alpha) \neq 0$, then $C_k(G)$ is not connected.

Proposition 4.9

For all $n \geq 3$, $S_3(C_n)$ is not connected.

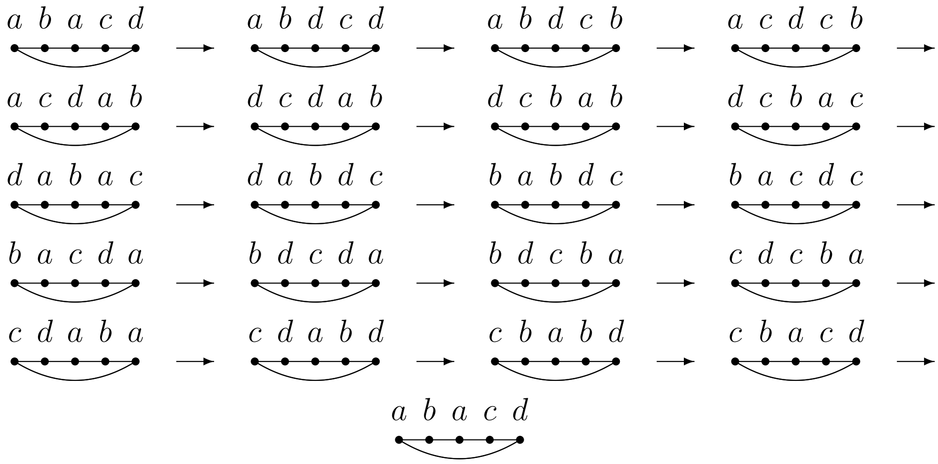
Proof. By Lemmas 4.4 and 4.8, it is enough to find a strong 3-colouring α with $W(\overrightarrow{C_n}, \alpha) \neq 0$. If $n = 3\,\ell$ for some positive integer ℓ, the pattern 1,2,3,1,2,3,...,1,2,3 provides a 3-colouring α of C_n with $W(\overrightarrow{C_n}, \alpha) = n \neq 0$. For $n = 4$, it is easy to see that $S_3(C_4)$ is the graph with 12 isolated vertices. If $n = 3\,\ell + 1 > 4$, then we use the pattern 1,2,3,1,2,3,...,1,2,3,2, which gives $W(\overrightarrow{C_n}, \alpha) = n - 4 \neq 0$. Finally, if $n = 3\,\ell + 2 \geq 5$, we use the pattern 1,2,3,1,2,3,...,1,2,3,1,2, so we get $W(\overrightarrow{C_n}, \alpha) = n - 2 \neq 0$. □

Proposition 4.10

The strong colour graph $S_4(C_5)$ is not connected.

Proof. For any strong 4-colouring of the 5-cycle C_5, there are only two vertices having the same colour. Thus each strong 4-vertex-colouring of C_5 can be recoloured only on these two vertices, and each of these two vertices can be recoloured to only one new colour (since the two different colours of their neighbours are forbidden). This means each colouring has degree two in $S_4(C_5)$.

Straightforward counting shows that $S_4(C_5)$ has 120 vertices. But each colouring in $S_4(C_5)$ is contained in a cycle of length 20. To see this, we start with some strong 4-colouring of C_5 and then recolour:



By symmetry, we immediately get that $S_4(C_5)$ is a disjoint union of six copies of C_{20}, so it is not connected. $\square$

Proposition 4.11

The strong colour graph $S_5(C_6)$ is connected.

Proof. In any strong 5-colouring of the 6-cycle C_6, there are only two vertices having the same colour. Let α be a strong 5-colouring of $C_6 = v_1v_2 \dots v_5v_6v_1$. We call α an *a-standard colouring* if $\alpha(v_1) = \alpha(v_3) = a$.

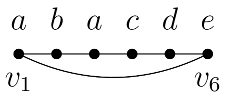
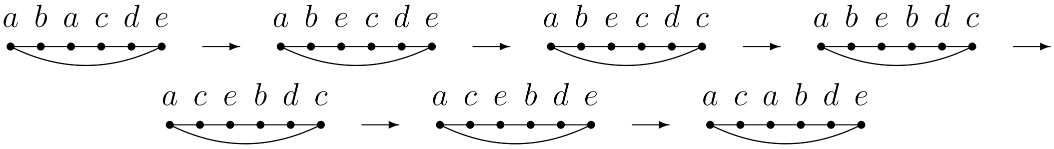


FIGURE 4.4: An a-standard colouring

We will prove the proposition by showing the following three steps.

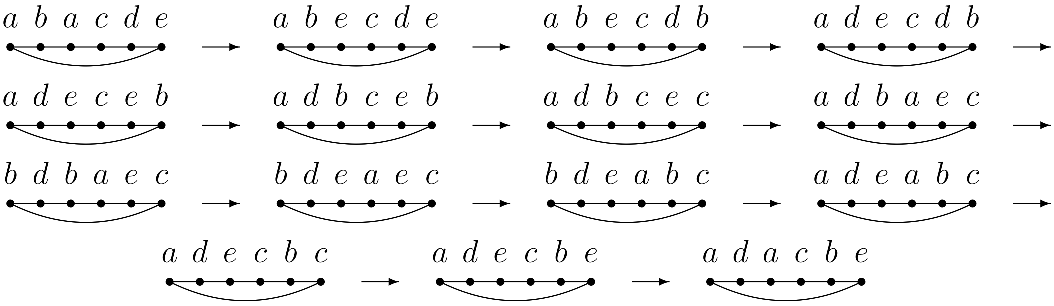
Step 1: There is a path from (a, b, a, c, d, e) to any other a-standard colourings.

First, we will show that there is a path from (a, b, a, c, d, e) to (a, c, a, b, d, e):



By symmetry, there is also a path from (a, b, a, c, d, e) to (a, e, a, c, d, b).

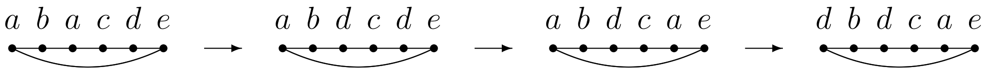
Next, we show that there is a path from (a, b, a, c, d, e) to (a, d, a, c, b, e):



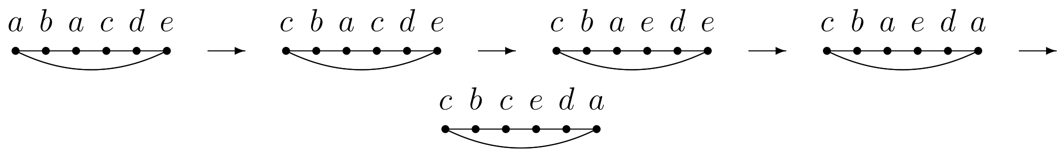
Now we consider a-standard colourings as permutations of $\{b, c, d, e\}$. Note that all these permutations can be generated by the transpositions (b, c), (b, d) and (b, e). Therefore, since we can find a path from (a, b, a, c, d, e) to (a, c, a, b, d, e), from (a, b, a, c, d, e) to (a, e, a, c, d, b), and from (a, b, a, c, d, e) to (a, d, a, c, b, e), there is a path from (e, a, e, b, c, d) to any other a-standard colourings.

Step 2: There is a path between any two types of standard colourings.

First, here is a path from an a-standard colouring α to an $\alpha(v_5)$-standard colouring:

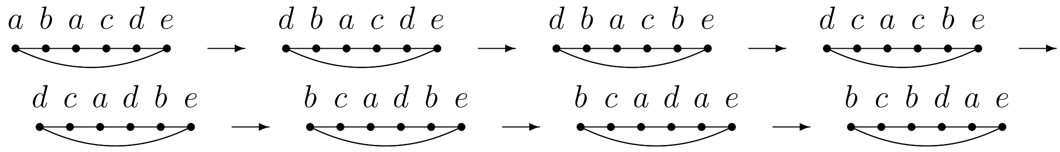


Next, a path from an a-standard colouring α to an $\alpha(v_4)$-standard colouring:



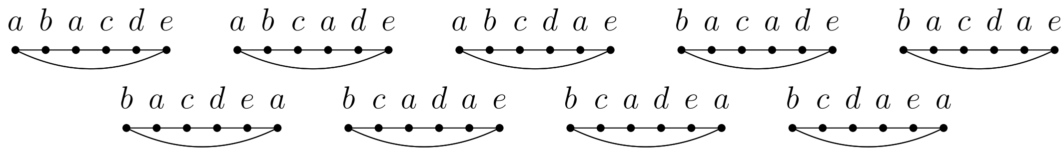
By symmetry, there is also a path from an a-standard colouring α to an $\alpha(v_6)$-standard colouring.

And finally, a path from an a-standard colouring α to an $\alpha(v_2)$-standard colouring:



Step 3: Each colouring has a path to some standard colouring.

Let α be a strong 5-colouring of C_6. Then α has one of the following forms:



The first form is already an a-standard colouring. For the second and the third forms, we just recolour vertex v_1 to c, and for the seventh and eight forms, we just recolour vertex v_3 to b. For all the remaining colourings, we can find a path of length two to some standard colouring. We will leave checking that to the reader.

It is straightforward to see that appropriate renaming of the colours and sequence of the paths in Steps $1-3$ will transform any strong 5-colouring of P_6 into any other strong 5-colouring. $\square$

Theorem 4.12

The strong colour graph $S_k(C_n)$ is connected if and only if $k \geq 4$, $n \geq 6$, and $n \geq k+1$.

Proof. We already have seen that $S_3(C_n)$, $n \geq 3$ and $S_4(C_5)$ are disconnected. By Theorems 4.2 and 4.7 we can obtain that $S_k(C_n)$ is connected for all $k \geq 4$, $n \geq 6$, and $n \geq k+2$. Since $S_5(C_6)$ is connected, all that is left to prove is that $S_k(C_{k+1})$ is connected for all $k \geq 6$.

Let $k \geq 6$ and let α and β be strong k-colourings of $C_{k+1} = v_1 v_2 \ldots v_{k+1} v_1$. In α, there will be a vertex, say v_1, which has an unique colour, say colour a. We say that a strong k-colouring of C_{k+1} is *good* if v_1 is the only vertex in C_{k+1}, which is coloured a. Thus α is good.

If β is good as well, then remove v_1, and let α' and β' be the strong $(k-1)$-colourings of P_k obtained from α and β, respectively. Since $k \geq 6$, by Proposition 4.6 there is a path in $S_{k-1}(P_k)$ from α' to β'. Using the same steps gives a path from α to β in C_{k+1}.

So suppose that β is not good. As we colour the $k+1$ vertices of C_{k+1} with k colours, there are only two vertices having the same colour. We distinguish five cases.

Case 1: In β, v_1 is coloured a, but there is a second vertex v_i coloured a as well.

Then just recolour v_i to another colour. The resulting colouring is good, and we are done by the paragraph above.

Case 2: In β, v_1 and some other vertex v_i have the same colour $b \neq a$, while a third vertex v_j with $j \neq 2, k+1$ is coloured a.

Then we first recolour v_1 to a, and then recolour v_j to another colour. Again, this gives a good colouring, so we are done.

Case 3: In β, v_1 and some other vertex v_i have the same colour $b \neq a$, while a third vertex v_j with $j \in \{2, k+1\}$ is coloured a.

Without loss of generality, we may assume that $j = k+1$.

Subcase 3.1: We have $i \geq 5$.

Now first recolour vertex v_i to colour $\beta(v_3)$, then recolour v_3 to $\beta(v_{k+1}) = a$, v_{k+1} to $\beta(v_4)$, v_4 to $\beta(v_1) = b$, v_1 to a, and finally recolour v_3 to some colour different from a. It is easy to check that the remaining colouring is good.

Subcase 3.2: We have $i \in \{3, 4\}$.

Recall that $j = k+1 \geq 7$. Now first recolour vertex v_i to colour $\beta(v_6)$, then recolour v_6 to $\beta(v_1) = b$. Now v_1 and v_6 have the same colour b, so we are back to Subcase 3.1.

Case 4: In β, v_1 has a unique colour $b \neq a$, while there are two vertices v_i and v_j coloured a.

Subcase 4.1: We have $\{i, j\} = \{2, k+1\}$.

Now first recolour vertex v_2 to $\beta(v_4)$, and then recolour v_4 to $\beta(v_1) = b$. Then we are back to Subcase 3.2.

Subcase 4.2: We have $\{i, j\} \neq \{2, k+1\}$.

Without loss of generality, we may assume that $i \neq 2, k+1$. Then we can recolour v_i

to $\beta(v_1) = b$. This means that v_1 and v_i have the same colour $b \neq a$, so we are back to Case 2 or 3.

Case 5: In β, v_1 has a unique colour $b \neq a$, there is a unique vertex v_i coloured a, and two vertices v_j and v_ℓ have the same colour $c \neq a, b$.

Subcase 5.1: We have $\{j, \ell\} = \{2, k+1\}$.

Since $k + 1 \geq 7$, we must have $i \neq 3$ or $i \neq k$. Without loss of generality, assume that $i \neq 3$. Then recolour v_2 to $\beta(v_i) = a$. This brings us back to Case 4.

Subcase 5.2: We have $\{j, \ell\} \neq \{2, k+1\}$.

Without loss of generality, assume that $j \neq 2, k+1$. Then we can recolour v_j to $\beta(v_1) = b$. This means that v_1 and v_j have the same colour $b \neq a$, and then we are back to Case 2 or 3. $\square$

4.5 The Strong 3-Colour Graph of Trees

The aim of this section is to classify the trees T for which the strong 3-colour graph $S_3(T)$ is connected.

For this we need to consider some special trees. First, in Section 4.2 we saw that the strong k-colour graph of a complete bipartite graph is not connected, so $S_3(K_{1,n})$ is disconnected for all $n \geq 2$.

For $n \geq 1$ and $p, q \geq 2$, let I, Ψ_n and $\Phi_{p,q}$ be the graphs sketched in Figure 4.5, respectively.

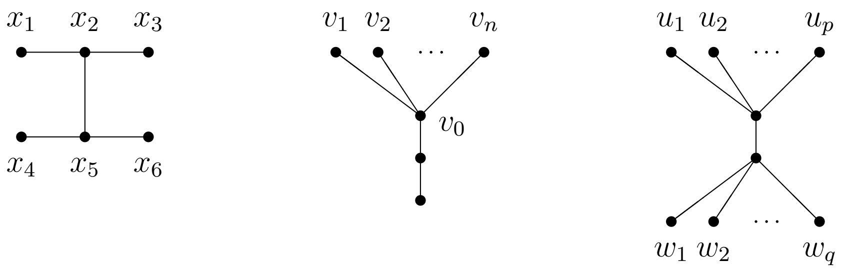


FIGURE 4.5: The graphs I, Ψ_n, and $\Phi_{p,q}$

It is straightforward to check that in any strong 3-colouring of Ψ_n we cannot recolour the vertex v_0 to another colour so that the resulting 3-colouring is strong again. Hence the strong colour graph $S_3(\Psi_n)$ is disconnected for all $n \geq 1$.

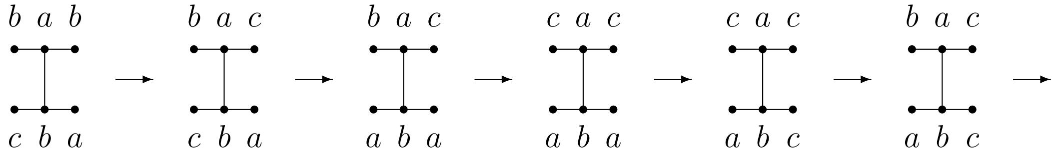
Proposition 4.13

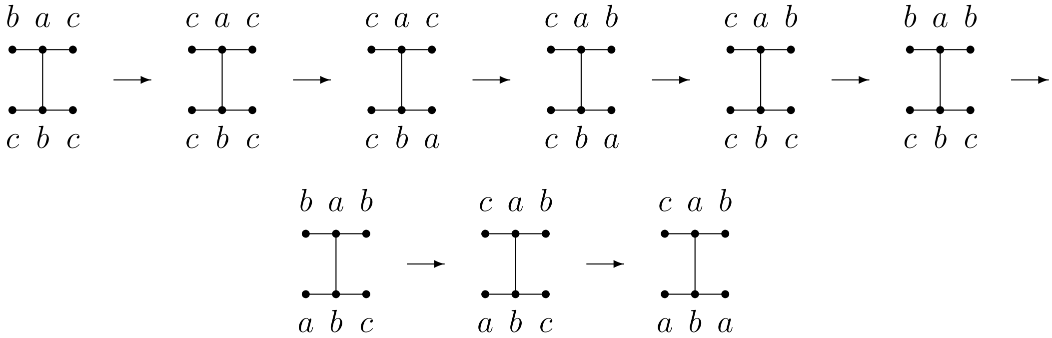
The strong 3-colour graph $S_3(I)$ is connected.

Proof. Let α be a strong 3-colouring of the graph I, with vertex set $\{x_1, x_2, \ldots, x_6\}$ as shown in Figure 4.5. We call α an (a, b)-*colouring* if $\alpha(x_2) = a$ and $\alpha(x_5) = b$. Easy counting shows that for fixed a and b, there are 15 (a, b)-colourings (there are 2 choices for each of the other 4 vertices, but one of the resulting 16 3-colourings is not strong). As there are 6 choices for a pair (a, b) from 3 colours, there are a total of 90 strong 3-colourings of I.

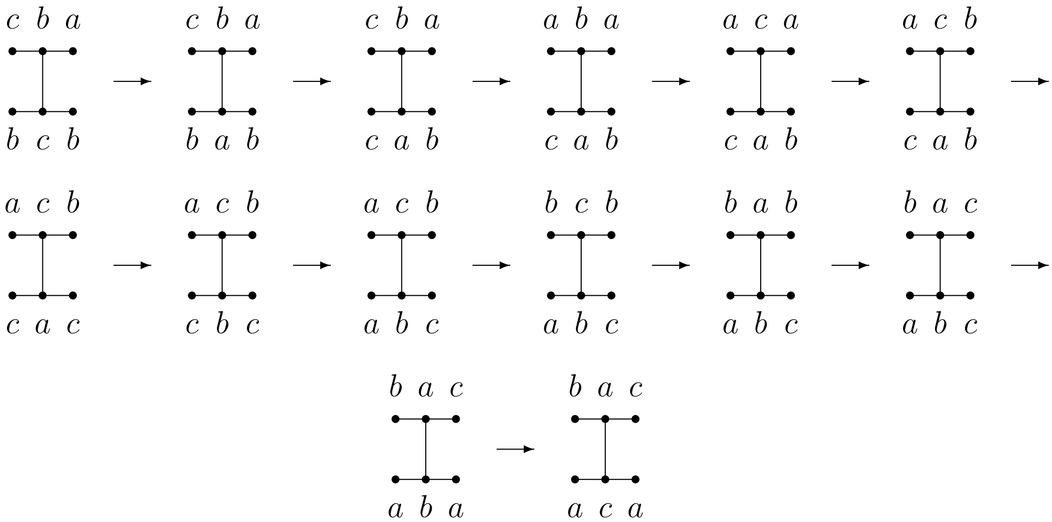
We will prove the proposition by combining the following two steps.

Step 1: For given a and b, there is a path containing all (a, b)-colourings.





Step 2: There is a path containing at least one strong 3-colouring from each type of colourings.



These two steps, together with appropriate renaming of the colours, will give all that is needed to transform any strong 3-colouring of I into any other strong 3-colouring. $\square$

Theorem 4.14

Let T be a tree. Then $S_3(T)$ is connected if and only if T contains P_5 or I as a subgraph.

Proof. Since $S_3(P_5)$ and $S_3(I)$ are connected, one direction is immediately proved by using Theorem 4.2.

For the other direction, suppose that T does not contain P_5, nor I. Since P_5 is a path with 4 edges, the longest path in T can have length at most 3. Thus T has to be one of the following: K_1, P_2, $K_{1,m}$, $m \geq 2$, or Ψ_n, $n \geq 1$. Note that T cannot be $\Phi_{p,q}$ for all $p, q \geq 2$; otherwise, it would contain I as a subgraph. Since K_1 and P_2 have fewer than 3 vertices, T cannot be one of these two graphs. We already saw that $S_3(K_{1,m})$, $m \geq 2$, and $S_3(\Psi_n)$, $n \geq 1$, are disconnected. Therefore, we can conclude that $S_3(T)$ is not connected. $\square$

4.6 Discussion

We have already seen in Lemma 4.4 that if the strong colour graph $S_k(G)$ is connected for some G and k, then so is the normal colour graph $C_k(G)$. In general, the reverse direction is not true. For instance, for all $m, n \geq 2$, for the complete bipartite graphs $K_{m,n}$ we have that for $k \geq 3$, $C_k(K_{m,n})$ is connected, whereas $S_k(K_{m,n})$ is not connected. In fact, we have already seen that if G has a complete bipartite graph as a spanning subgraph, then $S_k(G)$ is never connected. For $k \geq 3$ it is not hard to construct other graphs apart from complete bipartite graphs that have this property, but all examples we know of have a fairly special structure (for instance, see the trees Ψ_n in Figure 4.5). This makes us raise the following question.

Question 4.15

Is it possible to completely describe a class of graphs $\mathcal{H}$ so that if G does not contain a graph from $\mathcal{H}$ as a spanning subgraph, then $C_k(G)$ is connected if and only if $S_k(G)$ is connected?

Bibliography

[1] V. Auletta, A. Monti, M. Parente, and P. Persiano. A linear time algorithm for the feasibility of pebble motion on trees. *Algorithmica*, 23:223–245, 1999.

[2] J. Billingham, R. A. Leese, and H. Rajaniemi. Frequency reassignment in cellular phone networks. Technical report, Smith Institute Study Group Report, 2005. Available from: www.smithinst.ac.uk/Projects/ESGI53/ESGI53-Motorola/Report/.

[3] P. Bonsma and L. Cereceda. Finding paths between graph colourings: PSPACE-completeness and superpolynomial distances. *Theoret. Comput. Sci.*, 410:5215–5226, 2009.

[4] G. Brightwell, J. van den Heuvel, and S. Trakultraipruk. Connectivity properties of labelled token graphs. In Preparation.

[5] L. Cereceda, J. van den Heuvel, and M. Johnson. Connectedness of the graph of vertex-colourings. *Discrete Math.*, 308:913–919, 2008.

[6] L. Cereceda, J. van den Heuvel, and M. Johnson. Mixing 3-colourings in bipartite graphs. *European J. Combin.*, 30:1593–1606, 2009.

[7] L. Cereceda, J. van den Heuvel, and M. Johnson. Finding paths between 3-colorings. *J. Graph Theory*, 67:69–82, 2011.

[8] Y. M. Chee and A. Lim. The algorithmic complexity of colour switching. *Inform. Process. Lett.*, 43:63–68, 1992.

[9] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to algorithms.* MIT Press, Cambridge, MA, third edition, 2009.

[10] R. Diestel. *Graph Theory.* Springer, Berlin, 2010.

[11] R. Fabila-Monro, D. Flores-Penaloza, C. Huemer, F. Hurtado, J. Urrutia, and D. R. Wood. Token graphs. *Graphs Combin.*, 28:365–380, 2009.

[12] G. Fertin, A. Raspaud, and O. Sýkora. No-hole $L(p, 0)$ labelling of cycles, grids and hypercubes. In *Structural information and communication complexity*, volume 3104 of *Lecture Notes in Comput. Sci.*, pages 138–148. Springer, Berlin, 2004.

[13] S. Fujita, T. Nakamigawa, and T. Sakuma. Colored pebble motion on graphs. *European J. Combin.*, 33:884–892, 2012.

[14] M. R. Garey and D. S. Johnson. *Computers and intractability. A guide to the theory of NP-completeness.* A Series of Books in the Mathematical Sciences. W. H. Freeman and Co., San Francisco, Calif, 1979.

[15] O. Goldreich. Finding the shortest move-sequence in the graph-generalized 15-puzzle is np-hard. In *Studies in complexity and cryptography*, volume 6650 of *Lecture Notes in Comput. Sci.*, pages 1–5. Springer, Berlin, 2011.

[16] G. Goraly and R. Hassin. Multi-color pebble motion on graphs. *Algorithmica*, 58:610–636, 2010.

[17] J. Han. Frequency reassignment problem in mobile communication networks. *Comput. Oper. Res.*, 34:2939–2948, 2007.

[18] D. Kornhauser, G. Miller, and P. Spirakis. Coordinating pebble motion on graphs, the diameter of permutation groups, and applications. In *SFCS '84*

Proceedings of the 25th Annual Symposium on Foundations of Computer Science, pages 241–250. IEEE Computer Society, Washington, DC, 1984.

[19] C. Lu, L. Chen, and M. Zhai. Extremal problems on consecutive $L(2,1)$-labelling. *Discrete Appl. Math.*, 155:1302–1313, 2007.

[20] O. Marcotte and P. Hansen. The height and length of colour switching. In *Graph colouring and applications*, volume 23 of *CRM Proc. Lecture Notes*, pages 101–110. Amer. Math. Soc., Providence, RI, 1999.

[21] C. H. Papadimitriou, P. Raghavan, M. Sudan, and H. Tamaki. Motion planning on a graph. In *SFCS '94 Proceedings of the 35th Annual Symposium on Foundations of Computer Science*, pages 511–520. IEEE Computer Society, Washington, DC, 1994.

[22] N. Robertson and P. D. Seymour. Graph minors. XIII. The disjoint paths problem. *J. Combin. Theory Ser. B*, 63:65–110, 1995.

[23] N. Robertson and P. D. Seymour. Graph minors. XX. Wagner's conjecture. *J. Combin. Theory Ser. B*, 92:325–357, 2004.

[24] A. Schrijver. *Combinatorial Optimization*, volume A. Springer, Berlin, 2003.

[25] S. Trakultraipruk. Connectedness of strong k-colour graphs. 2010. arXiv:1009.4440 [math.CO].

[26] J. van den Heuvel and S. Trakultraipruk. On the complexity of finding shortest paths in token graphs. In Preparation.

[27] D. B. West. *Introduction to Graph Theory.* Prentice Hall, New Jersey, second edition, 2001.

[28] R. M. Wilson. Graph puzzles, homotopy, and the alternating group. *J. Combin. Theory Ser. B*, 16:86–96, 1974.

[29] Z. Wu and S. Grumbach. Feasibility of motion planning on directed graphs. In *Theory and applications of models of computation*, volume 5532 of *Lecture Notes in Comput. Sci.*, pages 430–439. Springer, Berlin, 2009.

Appendix

According to Theorem 2.2 we have that the graph θ_0 (which is shown in Figure A.1) is the only 2-connected non-bipartite graph for which not all its labelled token graphs are connected. Additionally, it is claimed that its labelled token graph is disconnected when the multiset of tokens is one of the following: (2,2,2), (2,2,1,1), (2,1,1,1,1), or (1,1,1,1,1,1). To justify this claim we give more detail about how the labelled token graphs of θ_0 look like.

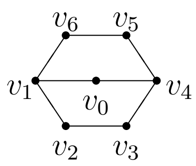


Figure A.1: The graph θ_0

A token configuration of θ_0, using the vertex names in Figure A.1, is said to be *standard* if the vertex v_1 is unoccupied. Performing an operation U (respectively, L) on a standard token configuration is taking 5 moves of the tokens on the upper (respectively, lower) 5-cycle anti-clockwise to get another standard token configuration (see Figure A.2 for examples).

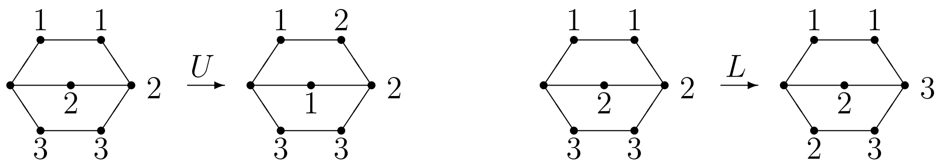


Figure A.2: Operations U and L

We can see that every token configuration has a path to some standard token configuration in the labelled token graph. Thus to see the structure of the labelled token graphs of θ_0, we need to consider standard token configurations only. Here we classify all the standard token configurations of θ_0 into some groups such that standard token configurations are in the same group if and only if they are in the same component of the labelled token graph.

Sometimes we put a dashed box around a standard token configuration in two figures from the same group to denote that these two standard token configurations are in fact the same.

The Structure of $T(\theta_0; (2,2,2))$

The following are the two groups of standard token configurations in the labelled token graph $T(\theta_0; (2,2,2))$.

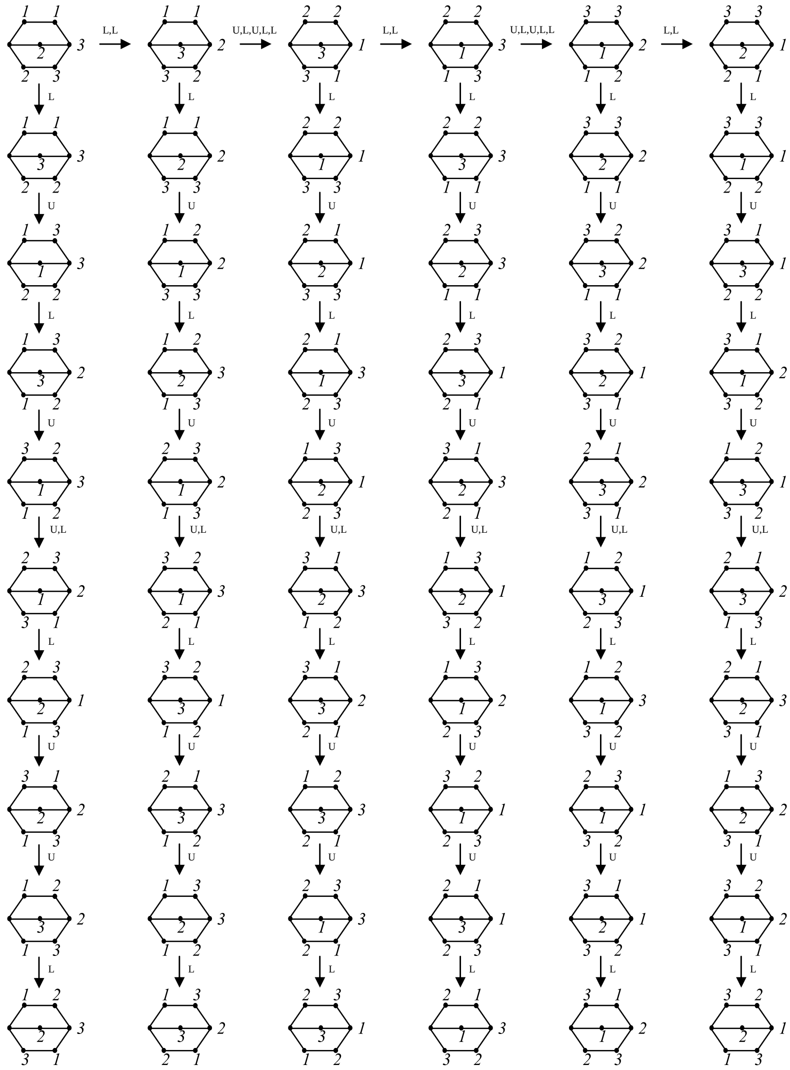


FIGURE A.3: Group 1 in $T(\theta_0; (2,2,2))$

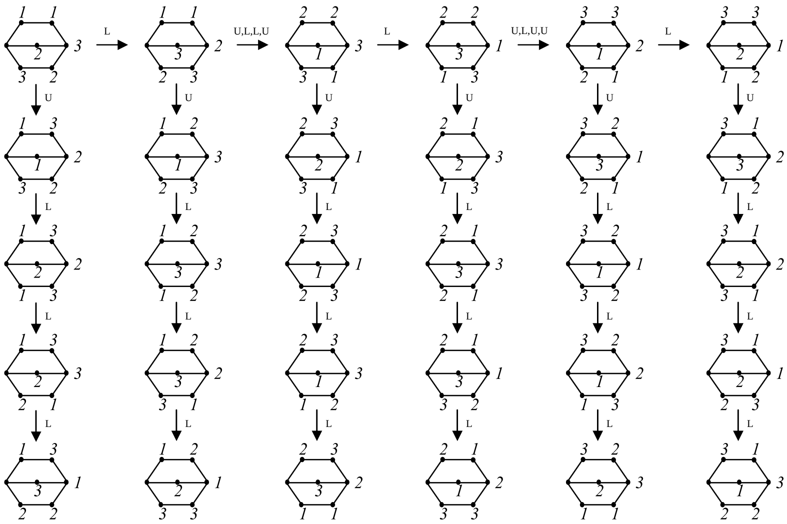


Figure A.4: Group 2 in $T(\theta_0; (2,2,2))$

The Structure of $T(\theta_0; (2, 2, 1, 1))$

The following are the two groups of standard token configurations in the labelled token graph $T(\theta_0; (2, 2, 1, 1))$.

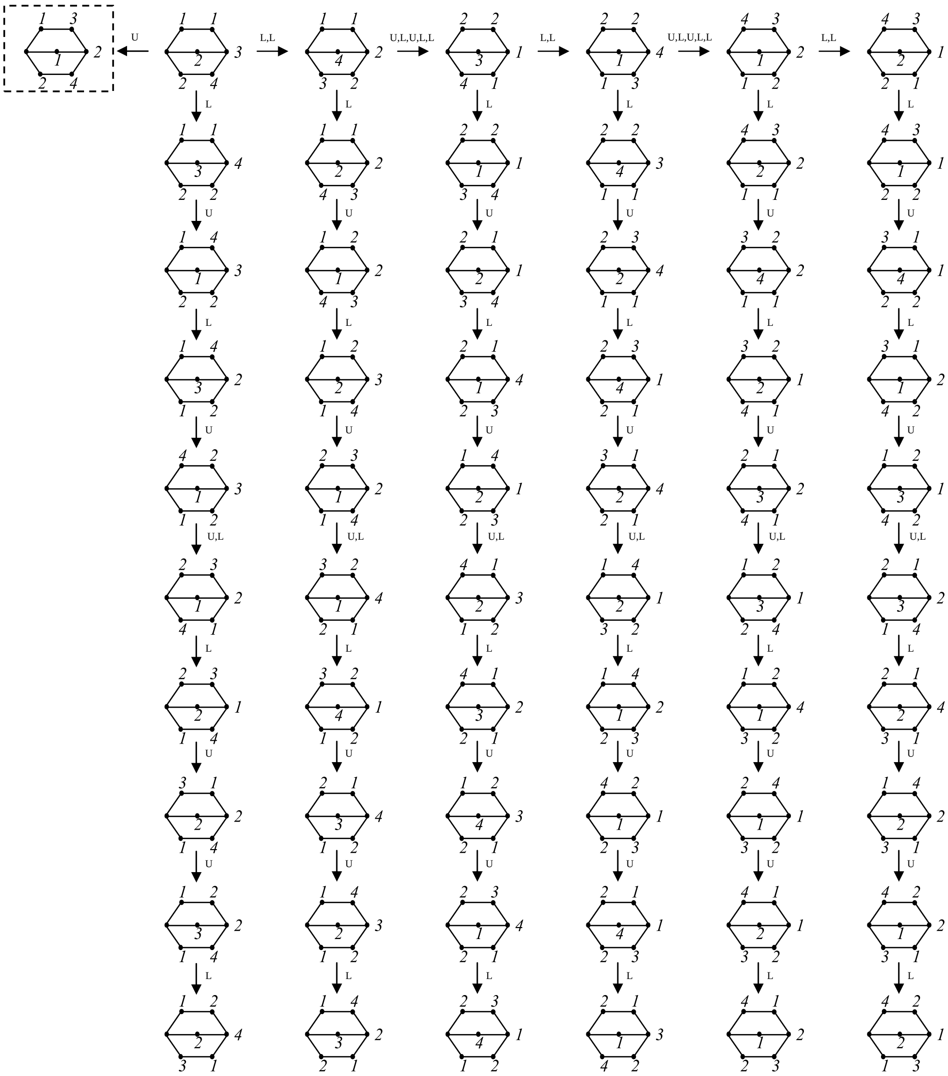


FIGURE A.5: Part 1 of Group A_1 in $T(\theta_0; (2, 2, 1, 1))$

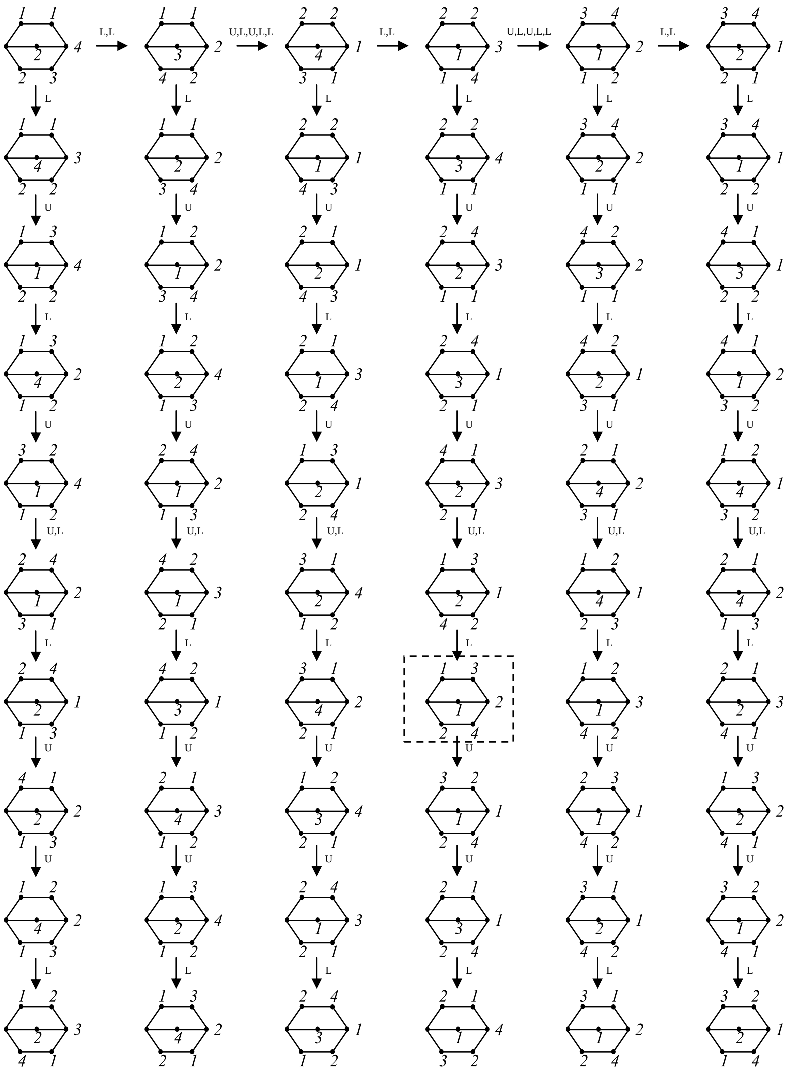


FIGURE A.6: Part 2 of Group A_1 in $T(\theta_0; (2,2,1,1))$

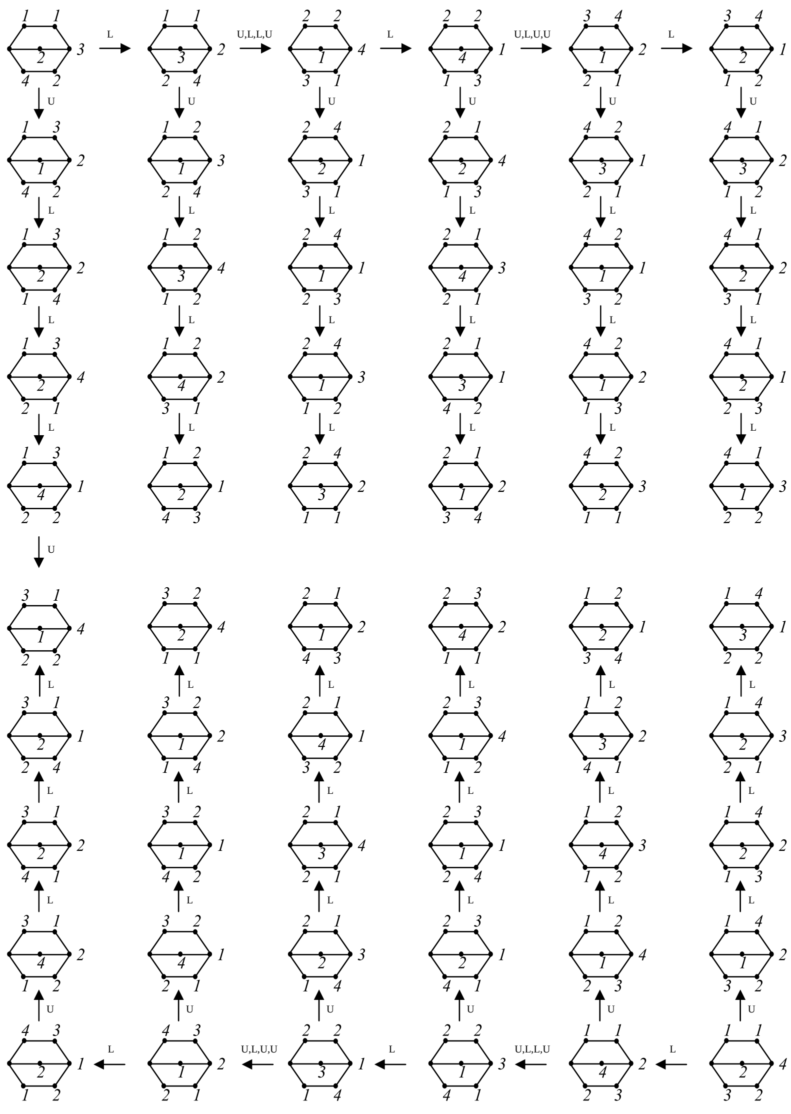


FIGURE A.7: Group A_2 in $T(\theta_0; (2, 2, 1, 1))$

The Structure of $T(\theta_0; (2, 1, 1, 1, 1))$

The following are the three groups of standard token configurations in the labelled token graph $T(\theta_0; (2, 1, 1, 1, 1))$.

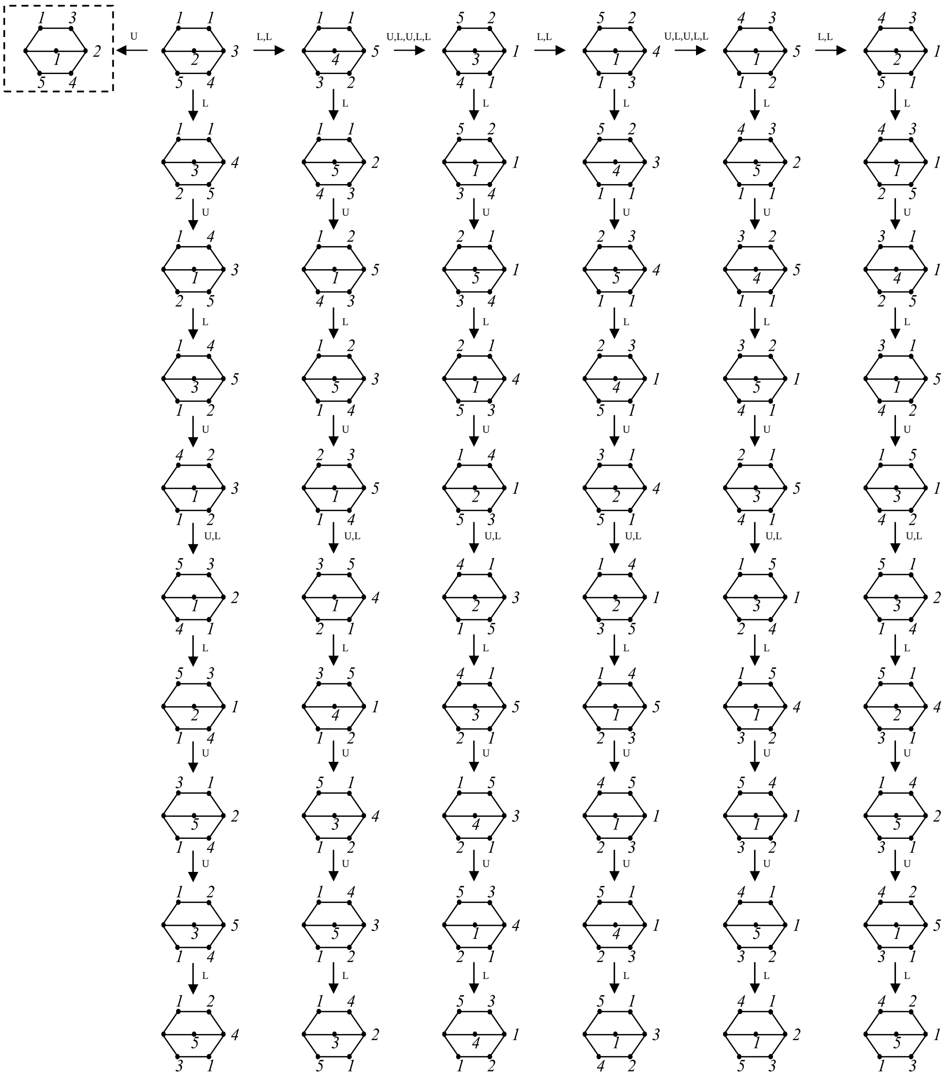


FIGURE A.8: Part 1 of Group B_1 in $T(\theta_0; (2, 1, 1, 1, 1))$

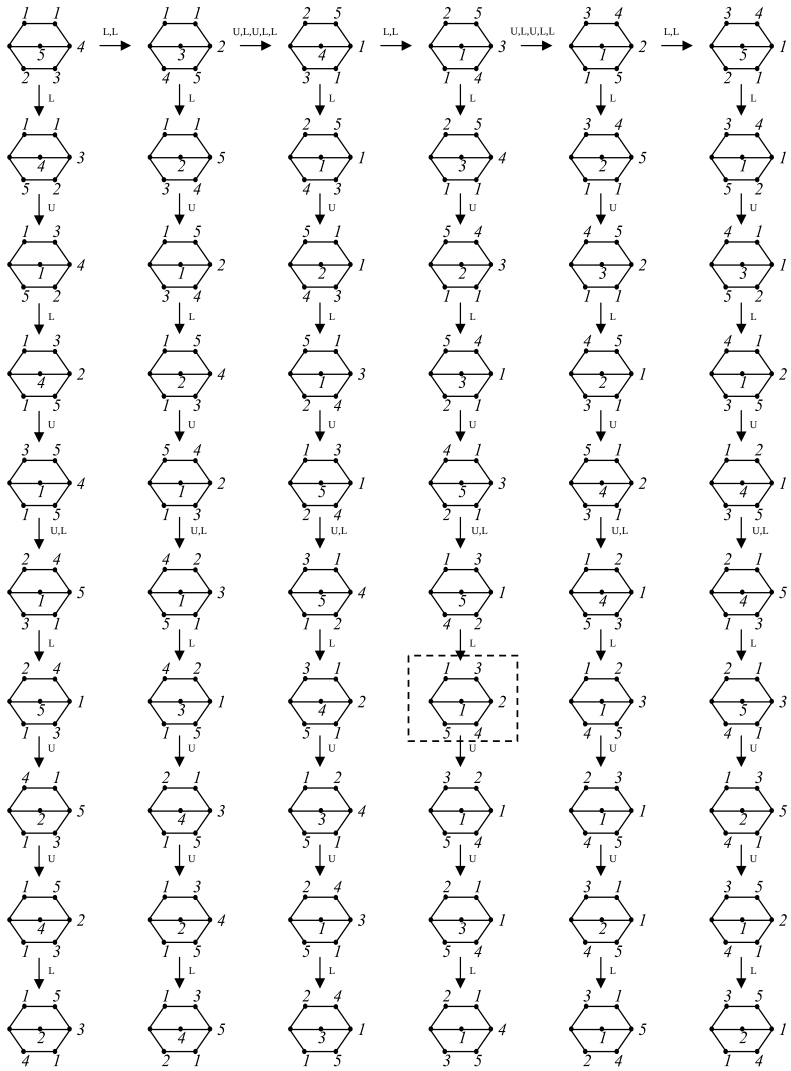


Figure A.9: Part 2 of Group B_1 in $T(\theta_0;(2,1,1,1,1))$

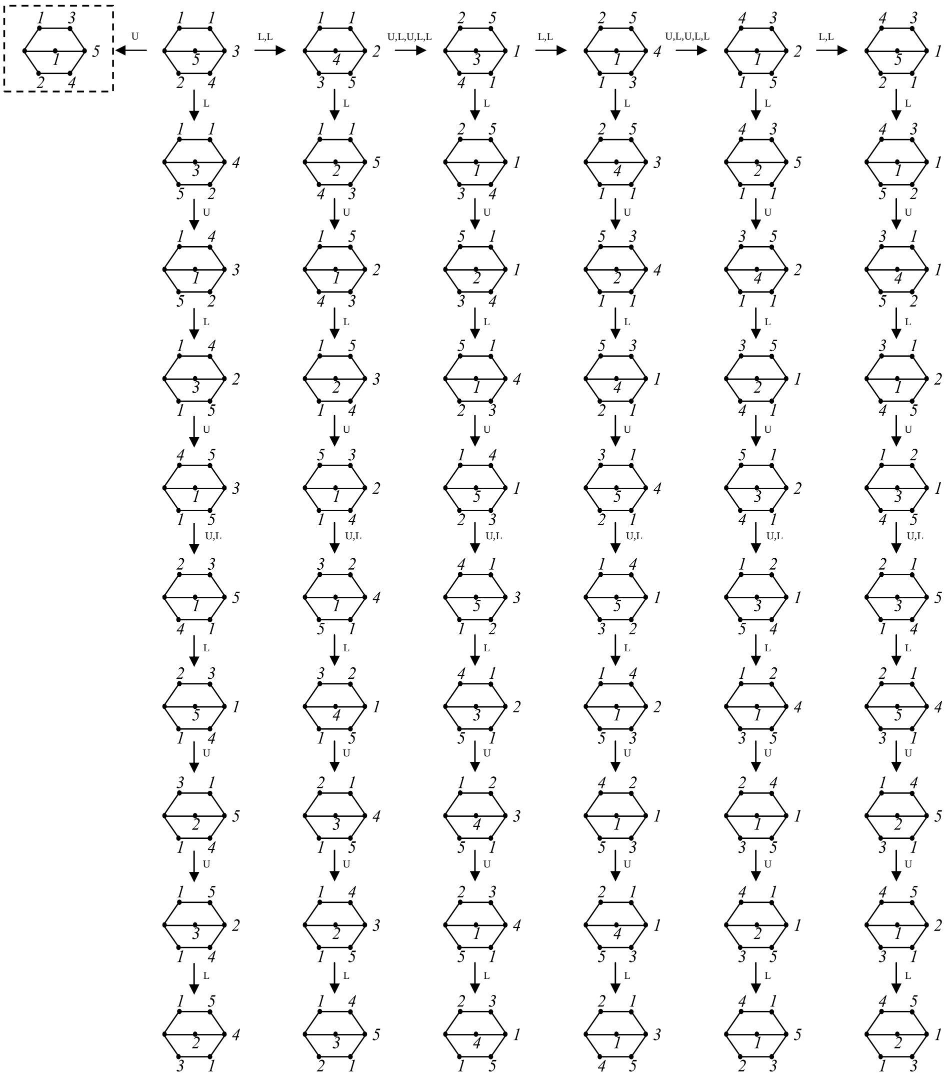


Figure A.10: Part 1 of Group B_2 in $T(\theta_0; (2,1,1,1,1))$

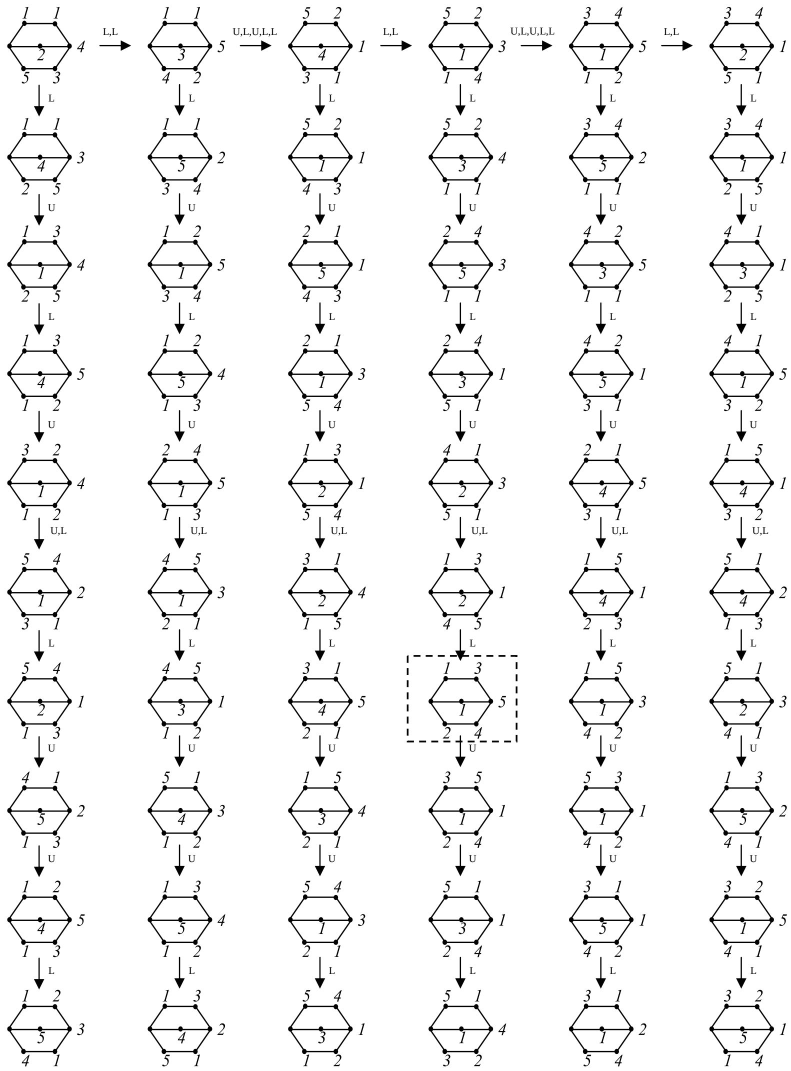


Figure A.11: Part 2 of Group B_2 in $T(\theta_0;(2,1,1,1,1))$

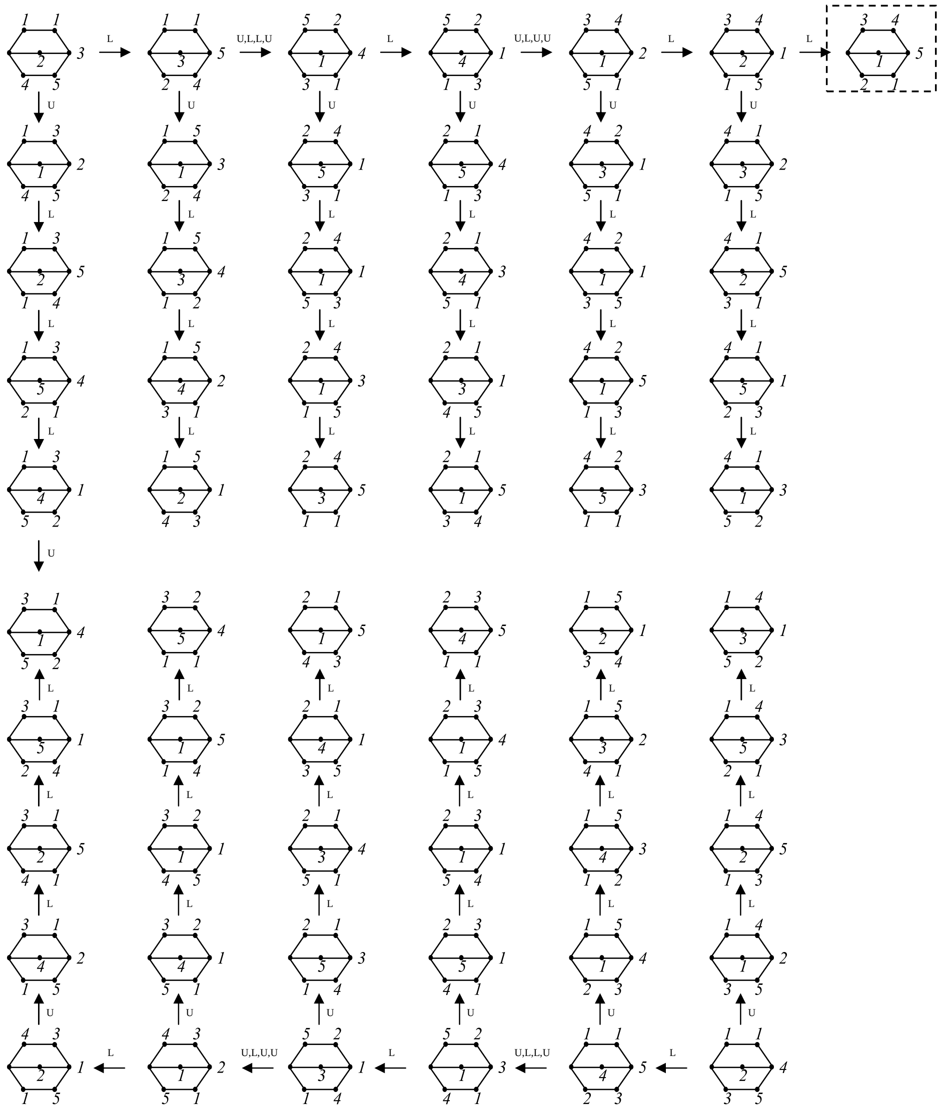


FIGURE A.12: Part 1 of Group B_3 in $T(\theta_0;(2,1,1,1,1))$

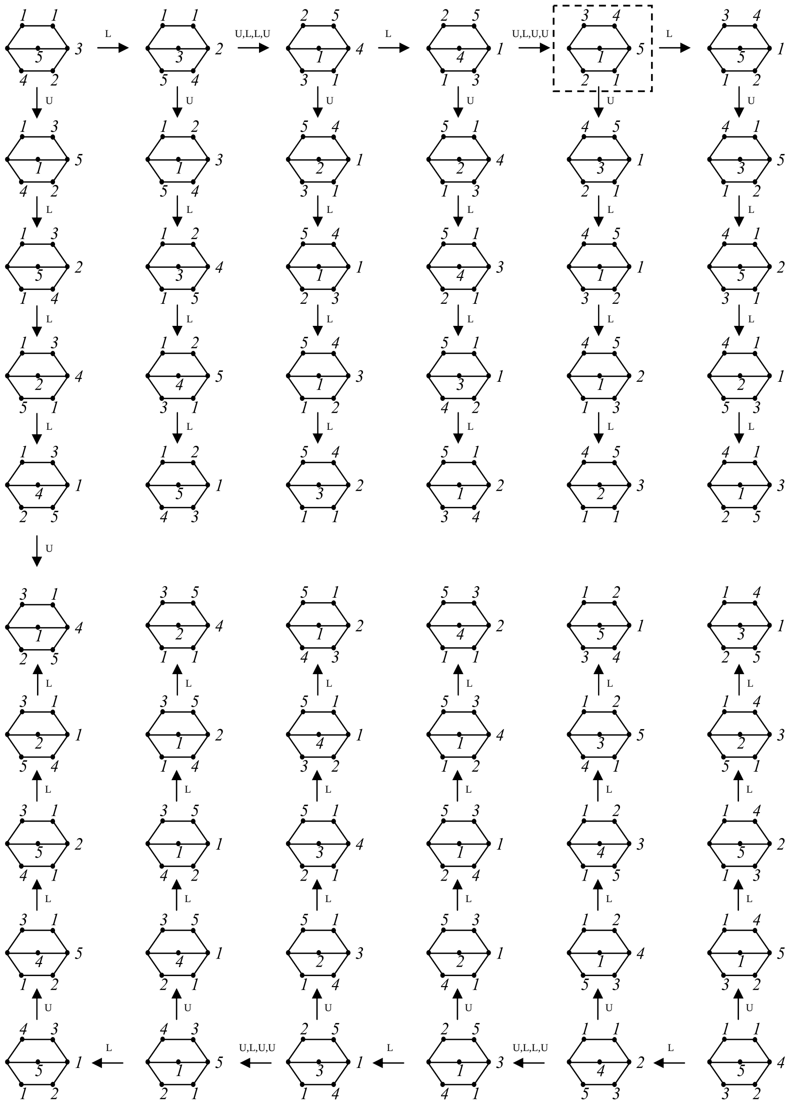


FIGURE A.13: Part 2 of Group B_3 in $T(\theta_0;(2,1,1,1,1))$

The Structure of $T(\theta_0; (1, 1, 1, 1, 1, 1))$

The following are the six groups of standard token configurations in the labelled token graph $T(\theta_0; (1, 1, 1, 1, 1, 1))$.

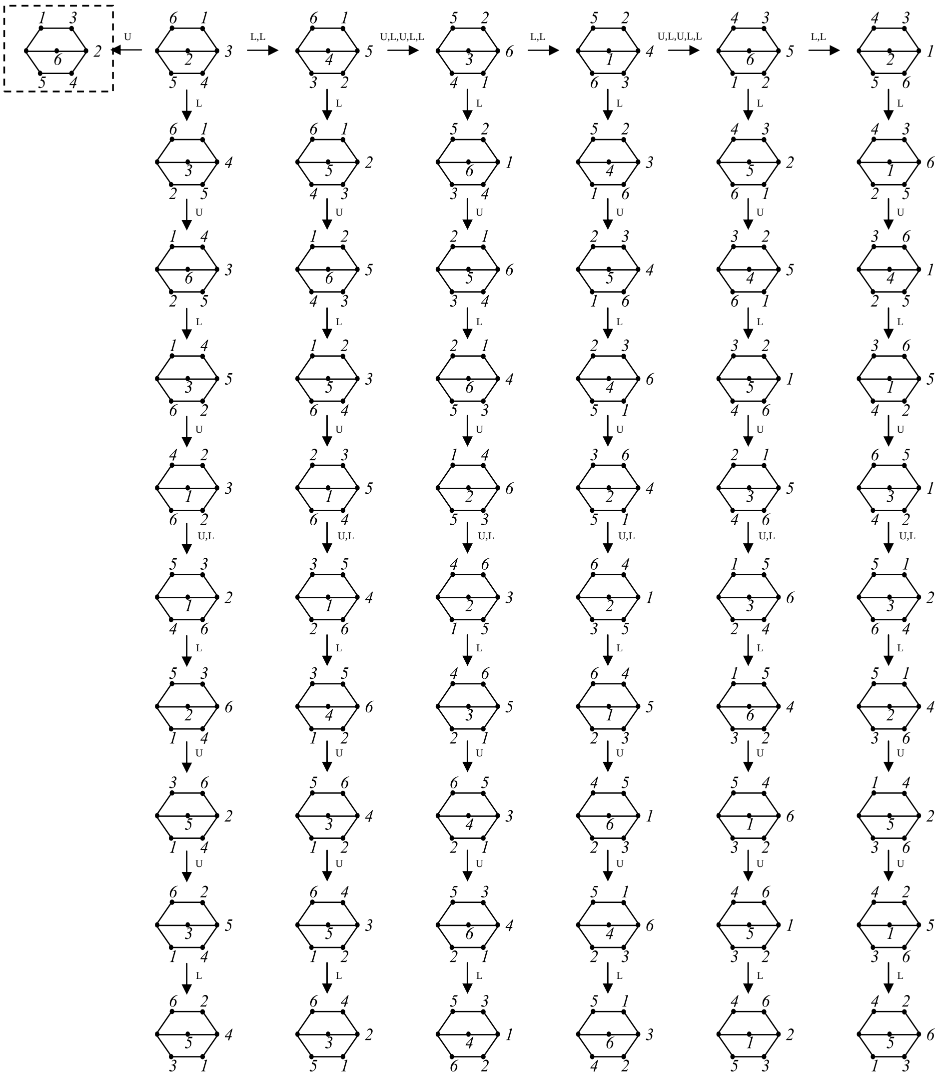


FIGURE A.14: Part 1 of Group C_1 in $T(\theta_0; (1, 1, 1, 1, 1, 1))$

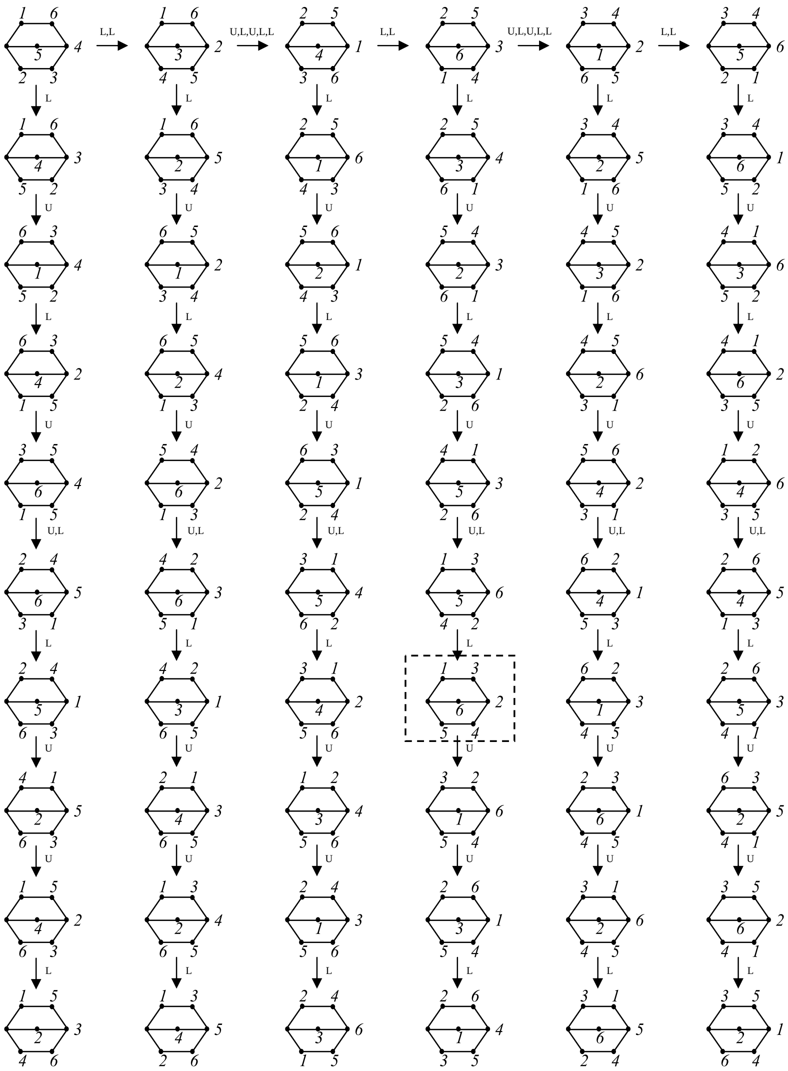


FIGURE A.15: Part 2 of Group C_1 in $T(\theta_0; (1,1,1,1,1,1))$

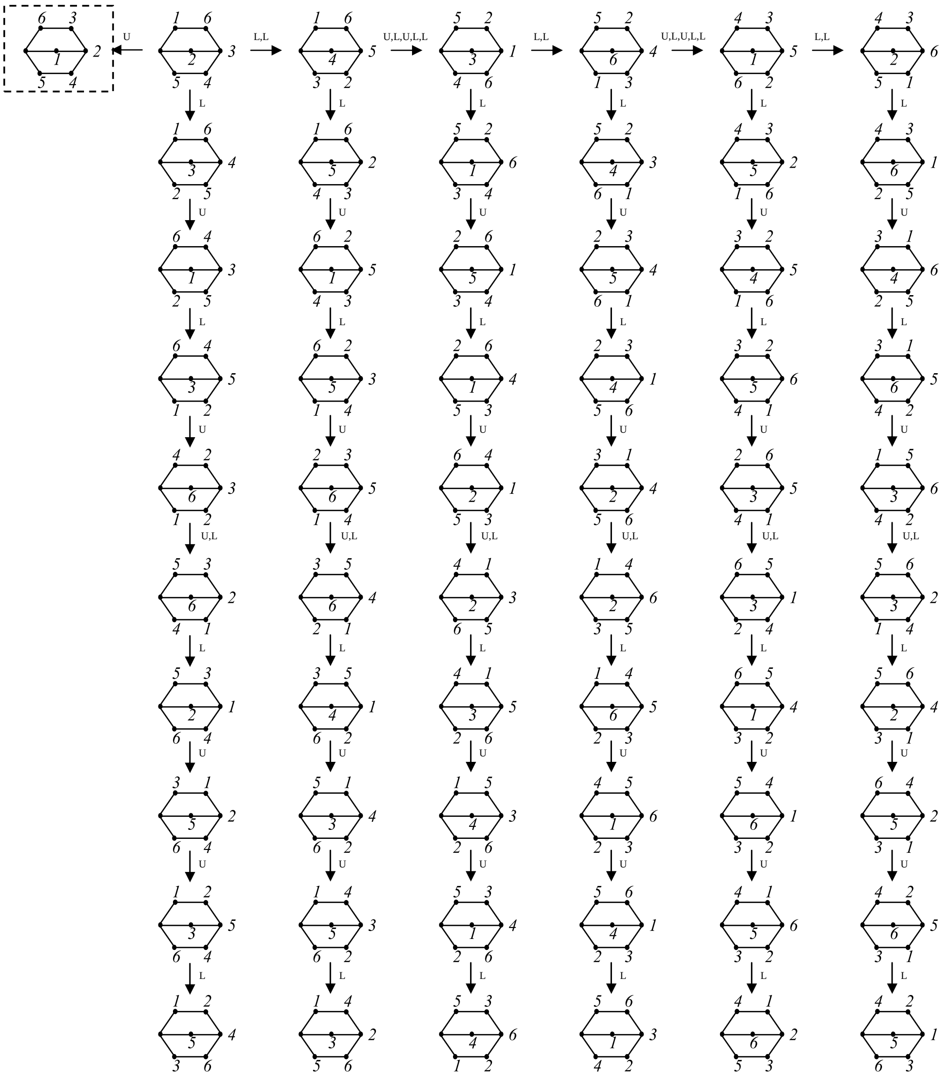


FIGURE A.16: Part 1 of Group C_2 in $T(\theta_0;(1,1,1,1,1,1))$

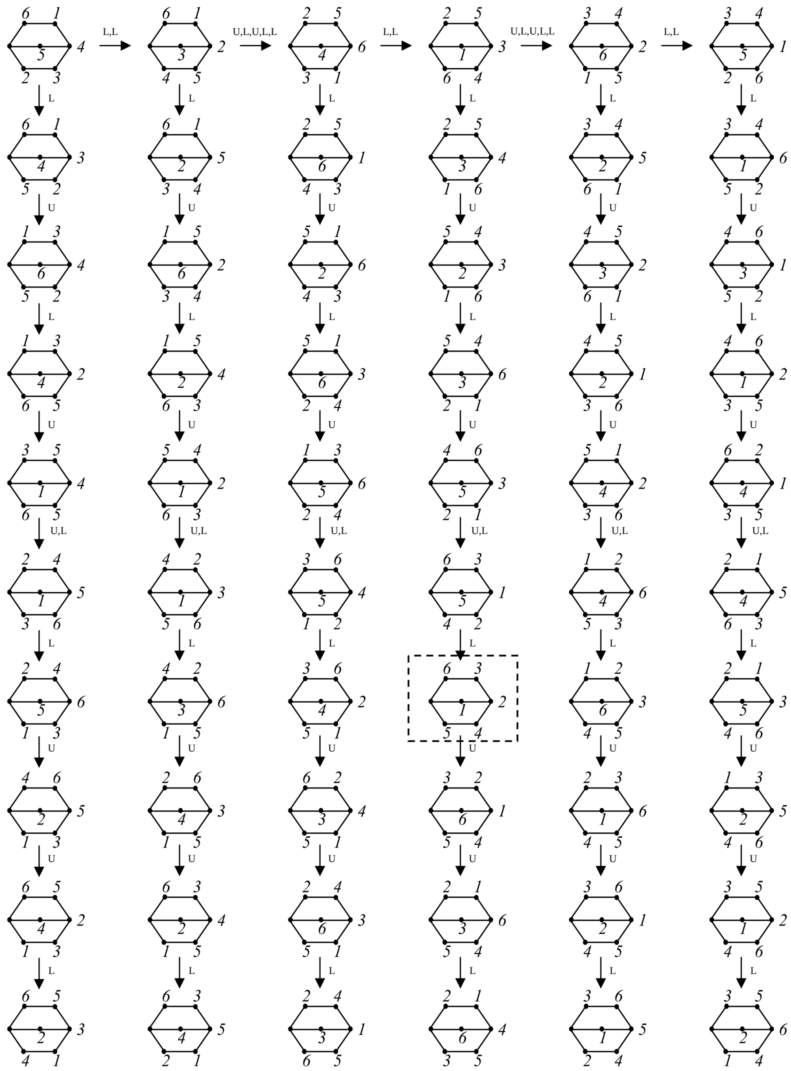


FIGURE A.17: Part 2 of Group C_2 in $T(\theta_0; (1, 1, 1, 1, 1, 1))$

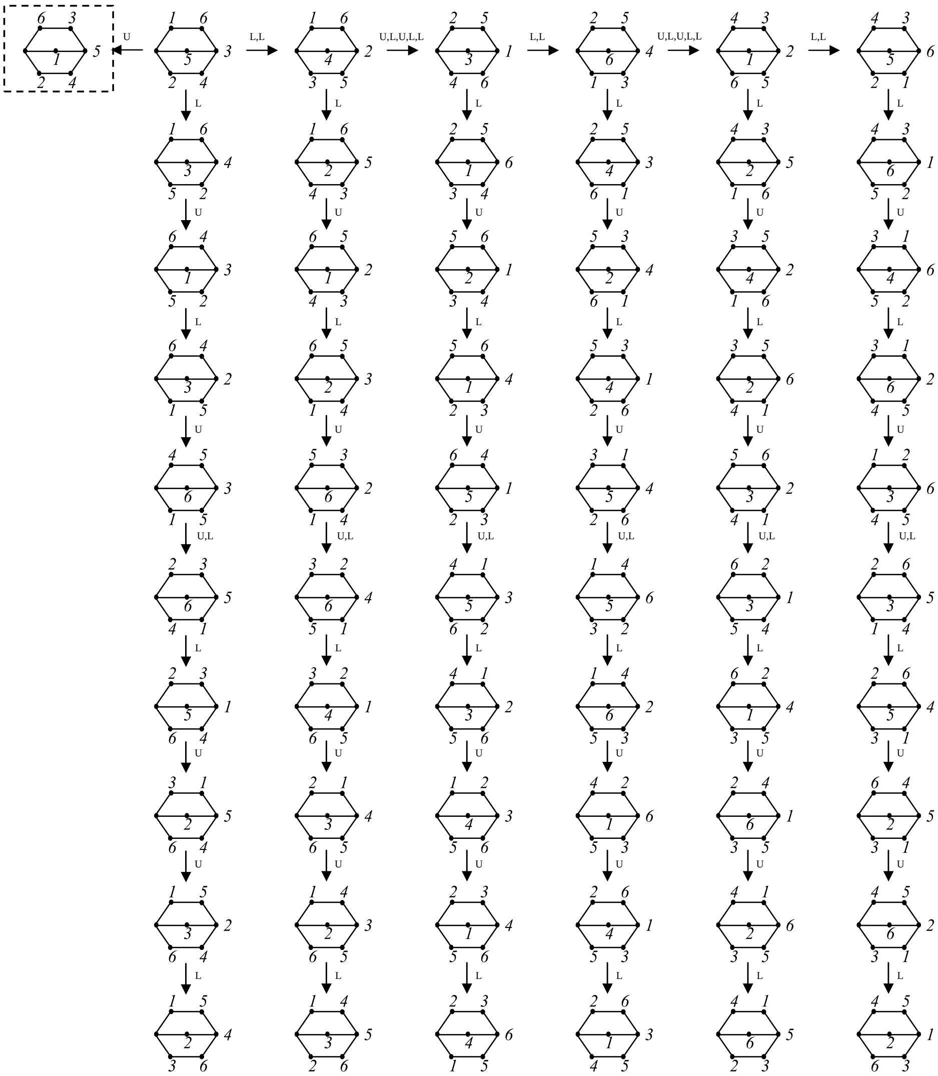


FIGURE A.18: Part 1 of Group C_3 in $T(\theta_0; (1,1,1,1,1,1))$

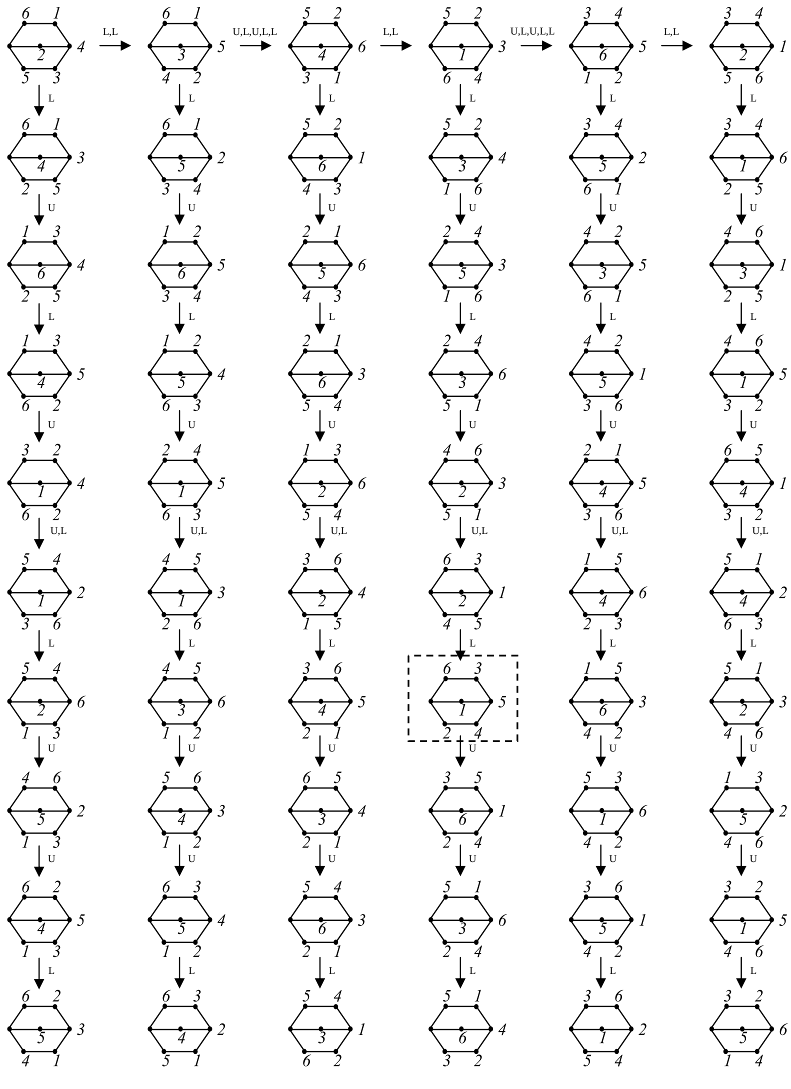


FIGURE A.19: Part 2 of Group C_3 in $T(\theta_0; (1,1,1,1,1,1))$

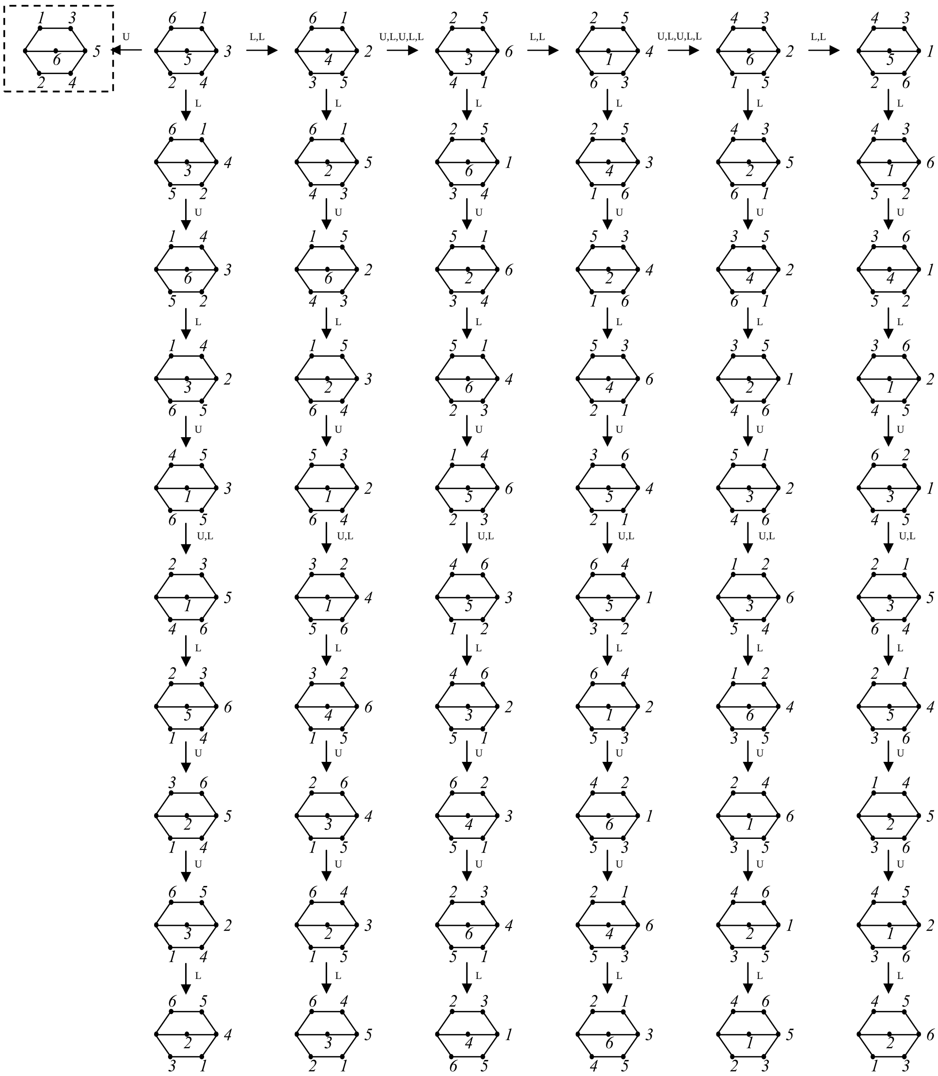


FIGURE A.20: Part 1 of Group C_4 in $T(\theta_0; (1, 1, 1, 1, 1, 1))$

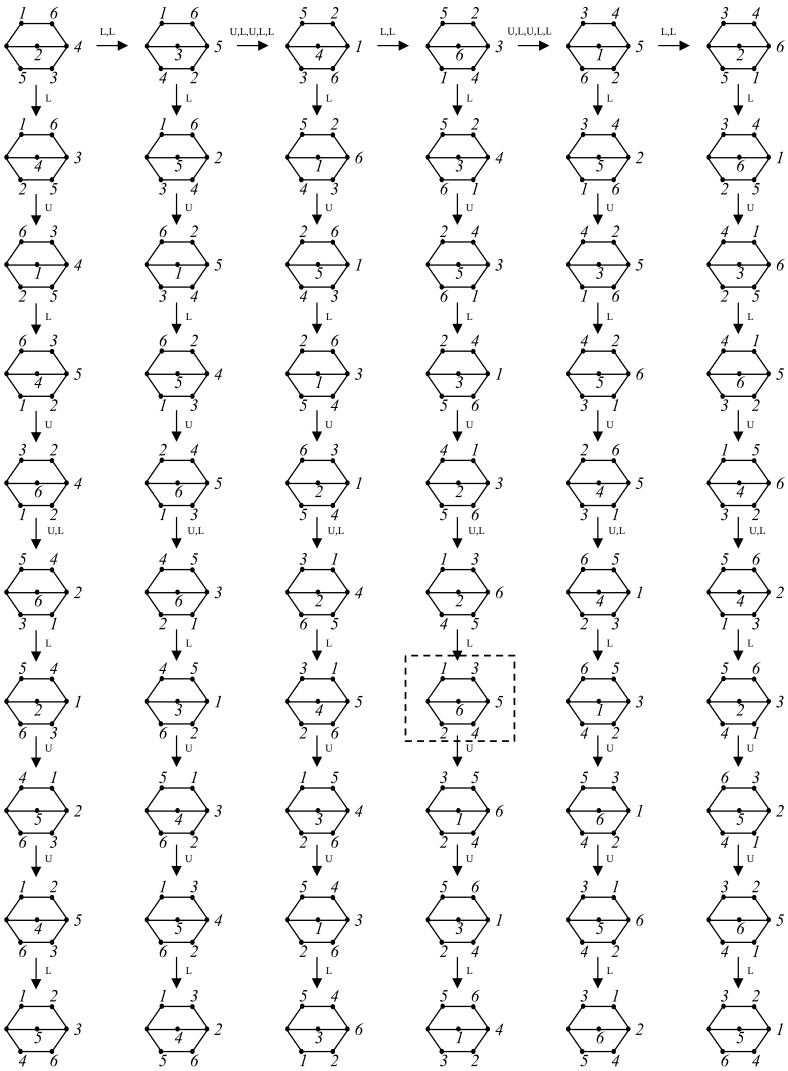


Figure A.21: Part 2 of Group C_4 in $T(\theta_0; (1,1,1,1,1,1))$

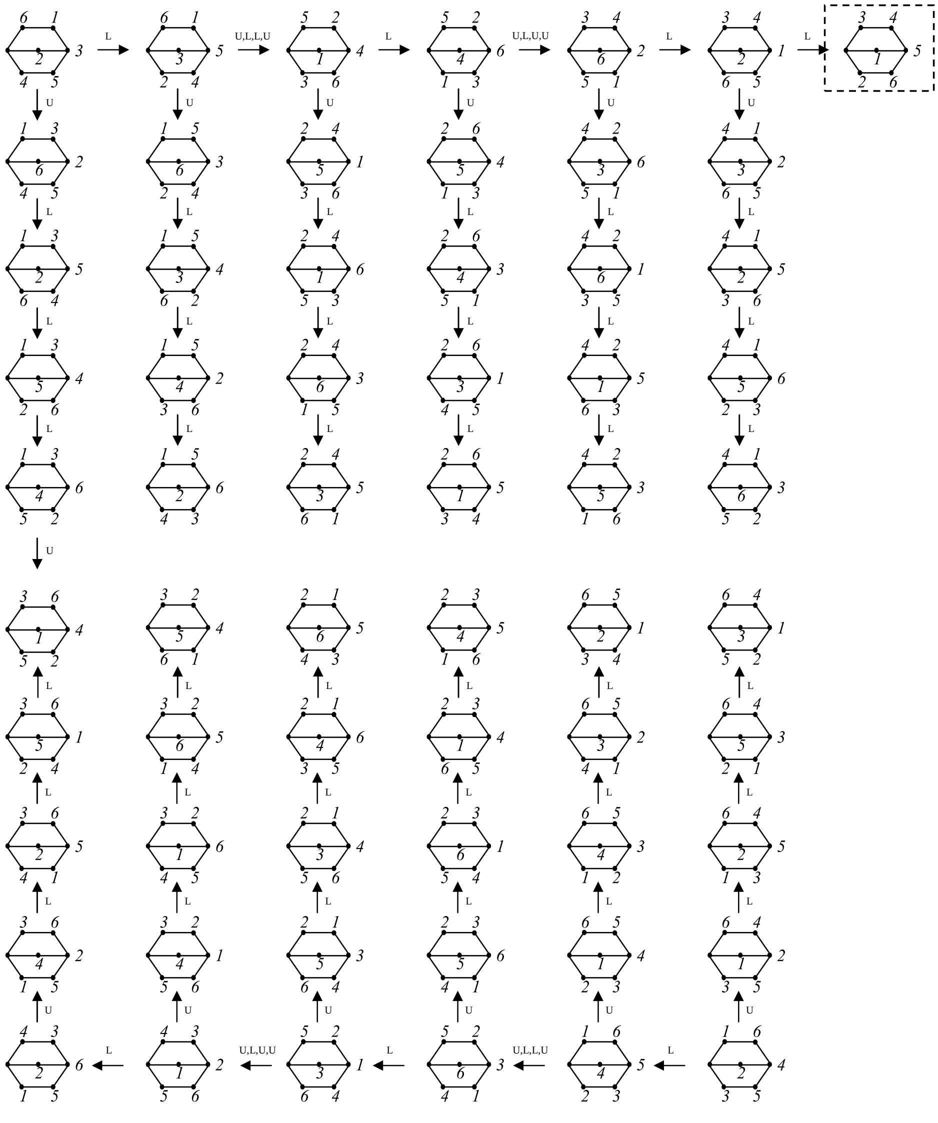


FIGURE A.22: Part 1 of Group C_5 in $T(\theta_0; (1,1,1,1,1,1))$

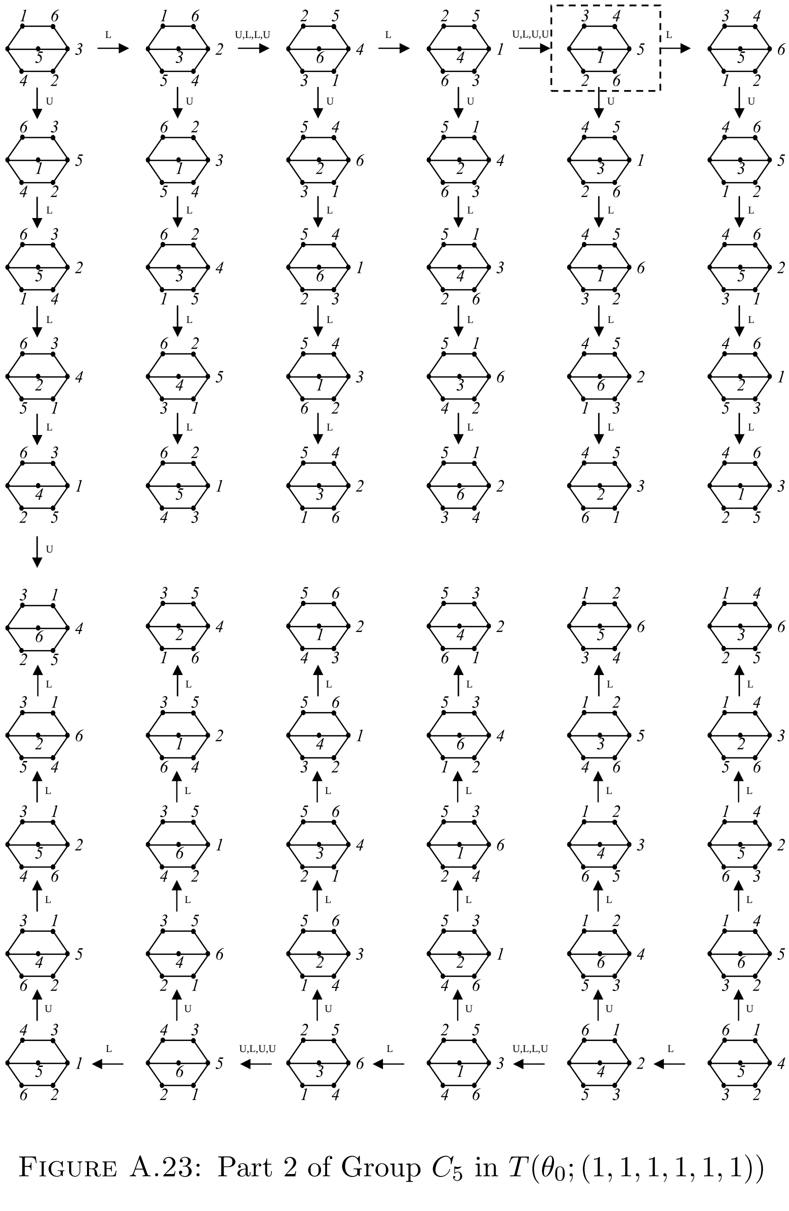


Figure A.23: Part 2 of Group C_5 in $T(\theta_0; (1,1,1,1,1,1))$

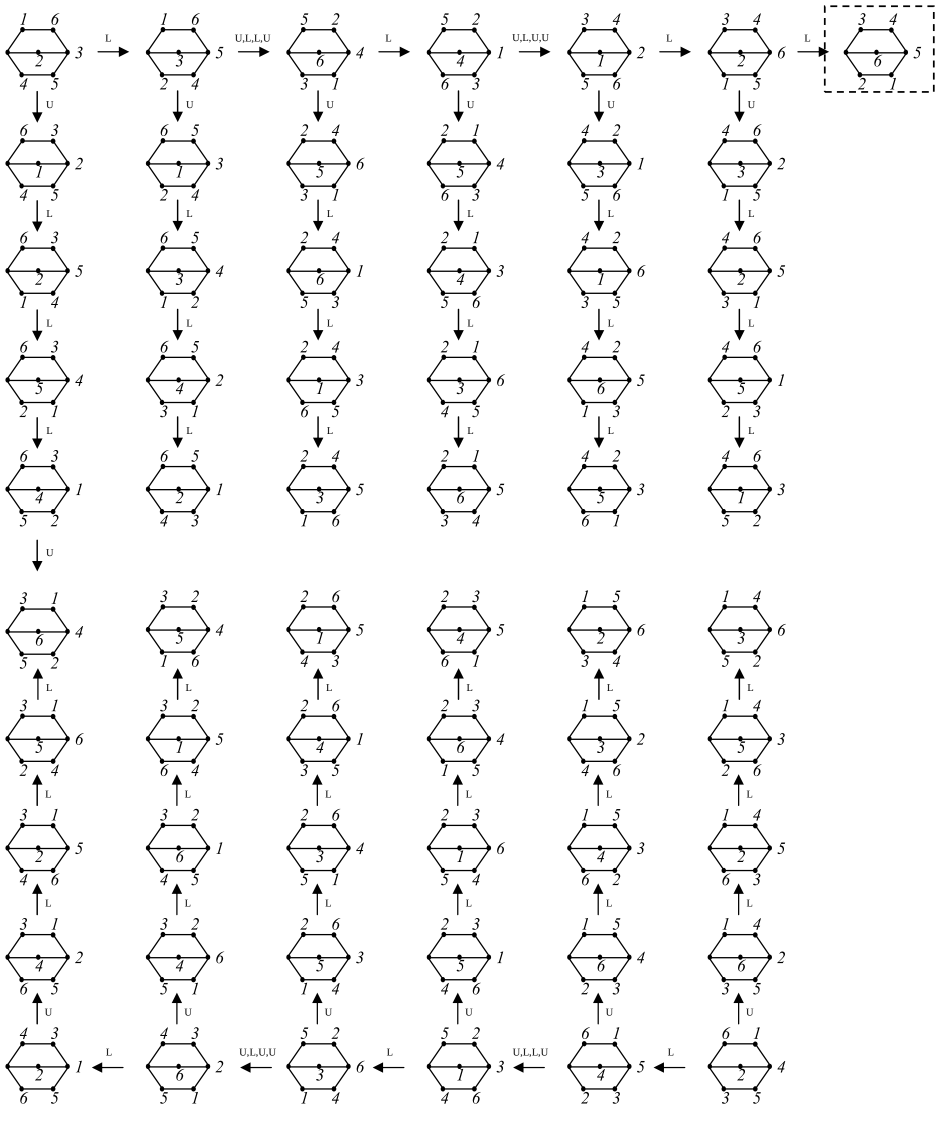


Figure A.24: Part 1 of Group C_6 in $T(\theta_0;(1,1,1,1,1,1))$

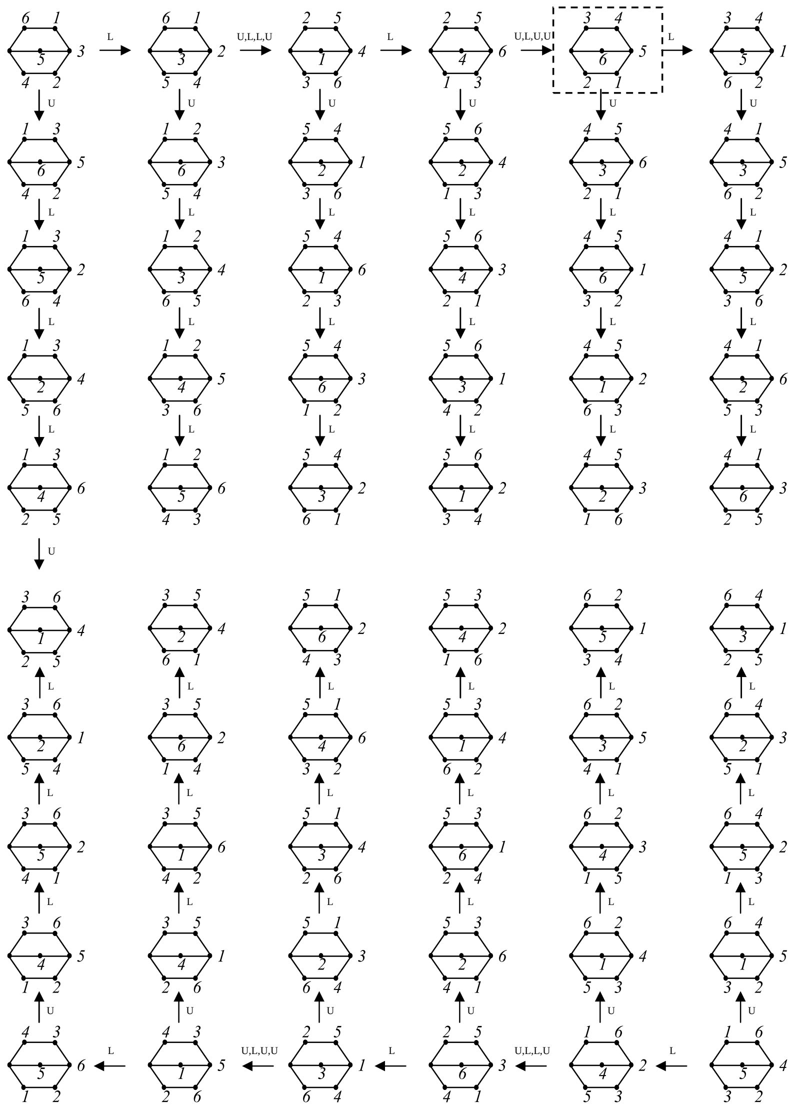


FIGURE A.25: Part 2 of Group C_6 in $T(\theta_0; (1,1,1,1,1,1))$