# Algebraic Graph Theory and Applications

Hisham Anwer (hisham@aims.ac.za)
African Institute for Mathematical Sciences (AIMS)

Supervised by Stephan Wagner
University of Stellenbosch

May 26, 2007

# Abstract

Graphs are very simple models that are used for formulating models in physics, computer science, social sciences, and other areas, especially in problems that have a natural graph structure. Basically, a graph is a set of vertices, where some pairs of vertices are connected by an edge. In this project, we provide an overview on graph theory and discuss aspects of linear algebra on graphs. In particular, the following problem is studied: let each of the vertices of a given graph be equipped with an indicator light and a button. These lights are said to follow the $\sigma$-rule if pressing a button corresponding to a vertex $v$ changes the state of the lights associated with all the vertices in the neighbourhood of $v$ (i.e. all vertices connected to $v$), and they are said to follow the $\sigma^+$-rule if pressing a button corresponding to a vertex $v$ changes the state of the lights associated with all the vertices in the closed neighbourhood of $v$ ($v$ itself and all vertices connected to $v$). The lights all being initially OFF, the problem is to find a sequence of buttons that changes the states of all lights, the so-called All-Ones problem. We consider a generalised form of this problem on complete graphs under the $\sigma$-rule and on arbitrary graphs under the $\sigma^+$-rule. We provide an algebraic approach to the All-Ones problem and justify our solutions by experimental results obtained from computer algorithms.

# Contents

# 1. Introduction

In this project, we will briefly introduce the main concepts of graph theory and linear algebra on graphs. A graph is a set of vertices, where some pairs are connected by an edge. The vertices are commonly represented as points in the plane, the edges as lines. Graphs are commonly used to formulate models in many other areas of science; in particular, graph theory has remarkable applications in the study of networks.

As a nice, simple and interesting problem that shows the application of linear algebra to graph theory is the All-Ones problem, introduced in [Sut89] in its algebraic formulation. Suppose each of the vertices of a given graph is equipped with an indicator light and a button. These lights are said to follow the $\sigma$-rule if pressing a button corresponding to a vertex $v$ changes the state of the lights associated with all the vertices connected to $v$; analogously, the lights are said to follow the $\sigma^+$-rule if pressing a button corresponding to a vertex $v$ changes the state of the lights associated with all the vertices in the closed neighbourhood of $v$ (which contains $v$ and all vertices connected to $v$).

The All-Ones problem stated as follows: If all the lights are initially OFF, is it possible to press a sequence of buttons such that all the lights are ON at the end? A possible generalisation for the All-Ones problem is to find a sequence of buttons by which we can convert a given situation of lights $X$ to another situation $Y$. In [LZ05] the All-Ones problem is called the vertex-to-vertex problem, and other versions of the vertex-to-vertex problem are also considered, namely the vertex-to-edge, the edge-to-vertex, and the edge-to-edge problems. Furthermore, the minimum vertex-to-vertex problem for trees is discussed in [LZ05] and [CLWZ04], the term minimum indicating that we search for the minimum number of buttons which solves the All-Ones problem. In [ACS98] some further theoretical aspects of the All-Ones problem are discussed.

The essay is organised as follows: we provide all necessary graph-theoretic definitions, including special graph classes, operations on graphs, connectivity, and the matrix representation of graphs, which will be of most importance for us. In this chapter we mostly keep our notations consistent with [Die05].

Then we proceed by discussing the All-Ones problem on complete graphs. Using our algebraic terminology, we are able to develop a complete solution for the All-Ones problem on complete graphs in its general form. Then we consider the All-Ones problem on general graphs, formulate the problem in an algebraic manner again and provide a procedure by which the All-Ones problem can be solved on general graphs. We then introduce cellular automata and the linear rule $\sigma^+$ and reformulate the problem in terms of cellular automata. Then we discuss some algebraic properties and the reversibility of the $\sigma^+$-rule. Finally, we present an algebraic proof for the existence of an Odd Parity Cover (equivalently, a solution to the All-Ones problem under the $\sigma^+$-rule) for general graphs, as given in [Sut89].

In the last Chapter, we provide an algorithm based on solving systems of linear equations modulo 2, and investigate the solutions to the All-Ones problem in several cases. In the Appendix we provide some illustrative example on the solution provided in Chapter 3, and we also develop an algorithm which solves the All-Ones problem on complete graphs under the $\sigma$-rule.

# 2. Graph Theory

## 2.1 Introduction

A graph is a finite nonempty set $V$ together with an irreflexive[1], symmetric relation $R$ on $V$. Since $R$ is symmetric, for each ordered pair $(v, u) \in R$, the pair $(u, v)$ also belongs to $R$. We denote the set of symmetric pairs in $R$ by $E$. For example, a graph $G$ may be defined by the pair $(V, E)$, where the set of vertices is $V = \{v_1, v_2, v_3, v_4\}$, and the set of the edges is $E = \{v_1v_2, v_1v_3, v_2v_3, v_2v_4, v_3v_4\}$. In dealing with graphs, it is often convenient to represent them by means of diagrams[2]. In such a representation we indicate the vertices by points or small circles, and we represent the edges by line segments or curves joining the two appropriate points, see Figure 2.1. The lines or the curves are drawn so that they pass through no points other than the two points they join. A graph with vertex set $V$ is said to be a graph on $V$. The vertex set of a graph $G$ is referred to as $V(G)$, its edge set as $E(G)$.

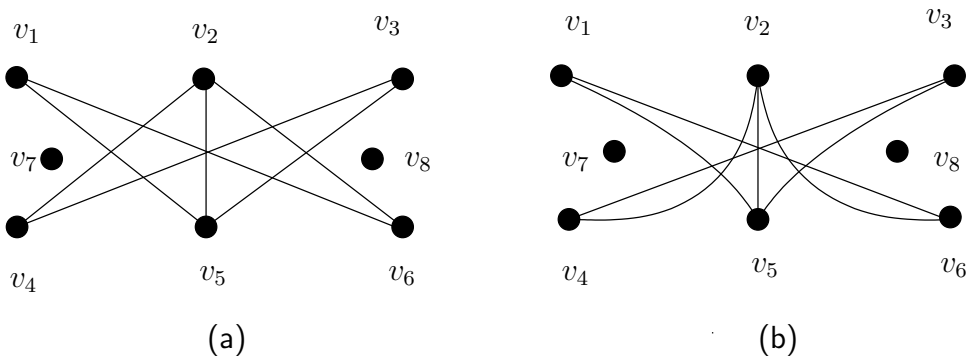



Figure 2.1: $(a)$ and $(b)$ are two equivalent diagrams of a graph $G$ on a set of vertices, $V = \{v_1, v_2, \ldots, v_8\}$ with edge set $E = \{v_1v_5, v_1v_6, v_2v_4, v_2v_5, v_2v_6, v_3v_4, v_3v_5\}$.

If $e = v_iv_j \in E(G)$, then we say that $e$ joins the vertices $v_i$ and $v_j$, or equivalently, we say $v_i$ and $v_j$ are each incident to $e$. Two vertices $v_i$ and $v_j$ are adjacent in a graph $G$ if $v_iv_j \in E(G)$. If $v_iv_j \notin E(G)$, then $v_i$ and $v_j$ are nonadjacent vertices. If $v_iv_j$ and $v_jv_k$ are distinct edges of $G$ then $v_iv_j$ and $v_jv_k$ are adjacent edges. In Figure 2.1, the vertex $v_2$ is adjacent to $v_4$, $v_5$ and $v_6$. The vertices $v_7$ and $v_8$ are not adjacent to any vertex in $G$.

For any vertex $v$ in a graph $G$, the neighbourhood of $v$ is defined as the set of all vertices in $G$ that are adjacent to $v$. The closed neighbourhood of a vertex $v$ in $G$ is defined as

$$N_v := \{u \in V |\ u \text{ adjacent to } v\} \cup \{v\} \tag{2.1}$$

[1] A relation $R$ on a set $S$ is said to be irreflexive if no element is related to itself

[2] A diagram of a graph completely describes the graph, and it is convenient to refer to the diagram of a graph $G$ as $G$ itself [Gar85].

## 2.2 Important Concepts

The number of vertices of a graph $G$ is called the order of $G$, the number of edges in $G$ is its size, namely

$$\mid G \mid = \mid V(G) \mid = \text{order}, \quad \parallel G \parallel = \mid E(G) \mid = \text{size}$$

We call $G$ a $(p, q)$-graph to indicate that $G$ has order $p$ and size $q$. A graph of order 0 or 1 is said to be a trivial graph. A graph is said to be finite if it has finite order, otherwise it is called infinite.

**Definition 2.2.1** *Let $G = (V, E)$ be a graph, and let $v$ be a vertex in $V$. The number $\mid E(v) \mid$ of edges of $G$ incident with $v$ is called the degree (or valency) of $v$ in $G$.*

Generally a vertex $v$ is called an even vertex in $G$ if it has even degree and it is an odd vertex if it has odd degree. A vertex $v$ of degree 0 is called an isolated vertex. The degree of $v$ is denoted by $d_G(v)$ or $d(v)$ if the graph is clear by context. The minimum degree in $G$ is denoted by $\delta(G) := \min\{d(v) \mid v \in V\}$ and the maximum degree is $\Delta(G) := \max\{d(v) \mid v \in V\}$. The average degree of $G$ is defined as

$$d(G) := \frac{1}{\mid V \mid} \sum_{v \in V} d(v) \tag{2.2}$$

It follows that $\delta \leq d(G) \leq \Delta(G)$. The average degree is a global quantification of the vertex degrees in $G$. In Figure 2.1 for instance, the vertex degrees in $G$ are $d(v_2) = d(v_5) = 3$, $d(v_1) = d(v_3) = d(v_4) = d(v_6) = 2$, and $d(v_7) = d(v_8) = 0$. The minimum degree in $G$ is $\delta(G) = 0$, the maximum degree is $\Delta(G) = 3$, and the average degree in $G$ is $d(G) = \frac{14}{8} = 1.75$. If every vertex in a graph $G$ has finite degree, the graph $G$ is said to be locally finite[3].

**Proposition 2.2.1** *For a $(p, q)$-graph $G$ the sum of the degrees of the vertices of $G$ equals twice the number of edges of $G$:*

$$\sum_{i=1}^{p} d(v_i) = 2q. \tag{2.3}$$

*Proof.* When summing the degrees of the vertices of a graph G we count each edge of G twice, once for each of the two vertices incident with the edge [Die05]. ⊡

**Corollary 2.2.1** *Every graph contains an even number of odd vertices*

*Proof.* This corollary is a direct consequence of Proposition 2.2.1. Any graph $G$ on $V$ has $\frac{1}{2}\sum_{v \in V} d(v)$ edges, so $\sum_{v \in V} d(v)$ is an even number, so for any graph $G$ the number of odd vertices has to be even [Gar85]. ⊡

[3] Locally finite graphs are not necessarily globally finite.

## 2.3 Null, Regular, Complete, and Grid Graphs

A graph which consists only of isolated vertices is called a null graph and denoted by $O^n$, where $n$ is the number of isolated vertices in the graph. In Figure 2.2(a) we present the null graphs $O^n$ for $n \leq 5$.

If every vertex of a graph $G$ has the same degree $k$, we say that $G$ is regular of degree $k$ or $G$ is $k-$regular. A graph $G$ is said to be a complete graph if every two of its vertices are adjacent. A complete graph of order $r$ is $(r-1)$-regular and denoted by $K^r$. In Figure 2.2(b) we present the complete graphs $K^r$ for $r \leq$5.

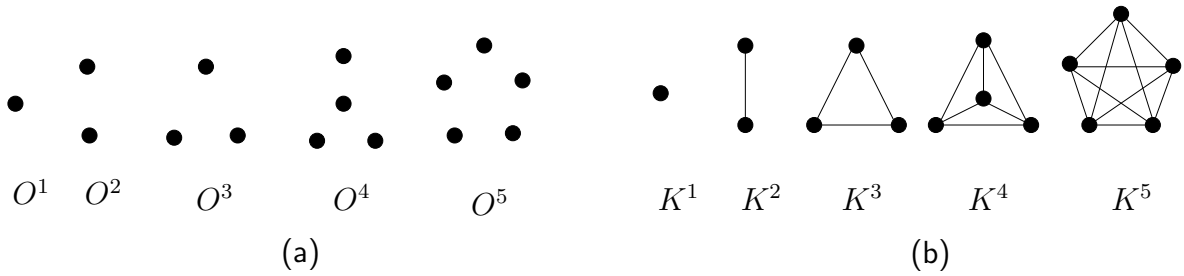



Figure 2.2: The Null graphs $O^n$ in part (a) and the complete graphs $K^r$ in part (b) are shown for $n, r \leq 5$.

A graph with $n^2$ vertices with a neighbourhood defined as in an $n \times n$ grid, i.e. two vertices are adjacent if they are next to each other horizontally or vertically in the grid, is called a grid graph and denoted by $G^n$. For any $n$ we can divide the vertices in $G^n$ into three categories, the corner-vertices, four vertices that lie at the corners of the grid, each of which has exactly 2 edge-adjacent vertices, the outer-vertices, the $4(n-2)$ vertices that lie at the outer columns and rows excluding the corners, each of which has exactly 3 edge-adjacent vertices, and the inner-vertices which are those $(n-2)^2$ with exactly 4 edge-adjacent vertices.

Furthermore, a graph whose edges have direction is called a directed graph. In a directed graph, vertices have both in-degrees and out-degrees: the in-degree of a vertex is the number of edges leading to that vertex, and the out-degree of a vertex is the number of edges leading away from that vertex.

## 2.4 Operations and Isomorphisms on Graphs

### 2.4.1 Addition and Removal

For a graph $G = (V, E)$ we define addition and removal operations by which we can add or subtract sets of vertices or edges as follows. Let $V'$ be any set of vertices. $G - V'$ is a graph obtained by removing all the vertices in $V' \cap V$ and their incident edges. And for a subset

$E'$ of $V \times V$ we define adding and removing edges as follows: $G - E' = (V, E \setminus E')$, and $G + E' = (V, E \cup E')$ respectively.

### 2.4.2 Induced and Spanning Subgraphs

Let $G = (V, E)$ and $G' = (V', E')$ be two graphs. The union of $G$ and $G'$ is defined by $G \cup G' := (V \cup V', E \cup E')$, and the intersection of $G$ and $G'$ is defined by $G \cap G' := (V \cap V', E \cap E')$. We say that $G$ and $G'$ are disjoint if $G \cap G' = \emptyset$. $G'$ is called a subgraph of $G$ if $V' \subseteq V$ and $E' \subseteq E$, and in such a case $G$ is called a supergraph of $G'$. If $G' \subseteq G$ and $G' \neq G$, i.e., $V' \neq V$ or $E' \neq E$, $G'$ is a proper subgraph of $G$. If $G' \subseteq G$ and $G'$ contains all the edges $uv \in E$ with $u, v \in V'$, then $G'$ is an induced subgraph; we say $V'$ induces or spans $G'$, and write $G' := G[V']$. Finally, $G' \subseteq G$ is a spanning subgraph of $G$ if $V' = V$.

### 2.4.3 Isomorphisms

Let $G(V, E)$ and $G'(V', E')$ be two graphs. By an isomorphism from $G$ to $G'$ we mean a bijective map $\varphi : V \longrightarrow V'$ such that for all $u, v \in V$ $uv \in E$ iff $\varphi(u)\varphi(v) \in E'$. We say that $G$ and $G'$ are isomorphic if an isomorphism exists from $G$ to $G'$, see Figure 2.3. In particular, if $G = G'$ $\varphi$ is called automorphism.

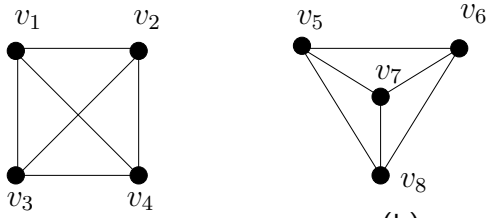



Figure 2.3: (a) and (b) give two isomorphic representations of the complete graph $K^4$, i.e. the two representations are equivalent.

## 2.5 Connected Graphs

### 2.5.1 Cycles and Paths

For a graph $G(V, E)$ let $v_i$ and $v_f \in V$. A $v_i - v_f$ walk in $G$ is an alternating sequence of vertices and edges of $G$, beginning with $v_i$ and ending with $v_f$, such that every edge joins the two vertices that come before and after it in the sequence, or equivalently, it is a non-empty graph $W = (V_w, E_w)$, where $V_w = \{v_i, v_1, \ldots, v_f\}$ and $E_w = \{v_i v_1, v_1 v_2 \ldots, v_{f-1} v_f\}$. $v_i$ and $v_f$ are called the ends.

A chain in a graph $G$ is a walk which does not repeat any edge. A circuit is a chain whose ends are equal i.e., $v_i = v_f$, and which contains at least three edges. A cycle is a circuit which does not repeat any vertex. A cycle with $n$ vertices is denoted by $C^n$. A path is a walk which does not repeat any vertex, see Figure 2.4. The number of edges of a path is its length. A path on $n$ vertices is denoted by $P^n$.[4] It is common to refer to a path by the natural sequence of its vertices.
The initial and terminal vertices are called the path ends, all the other vertices in $P$ are called inner vertices. Two or more paths are independent if none of them contains an inner vertex of

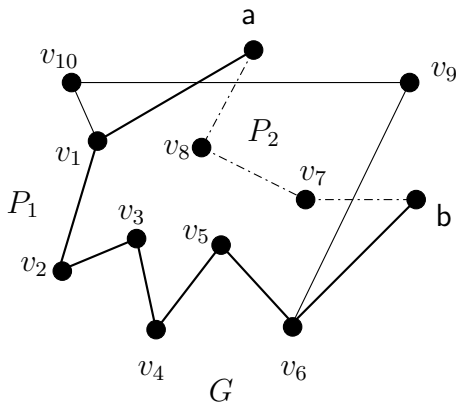



Figure 2.4: The thick lines $P_1$ and the dashed lines $P_2$ are two independent paths in $G$ linking the two vertices $a$ and $b$. $P_1 \cup P_2$ is a cycle in $G$.

the other. Clearly the union of any two independent paths with the same ends is a cycle in $G$. For instance in Figure 2.4 the path $P_1 = a\ v_1\ v_2\ v_3\ v_4\ v_5\ v_6\ b$ and the path $P_2 = a\ v_8\ v_7\ b$ are independent.

### 2.5.2 Connectivity

Two distinct vertices $u$ and $v$ in a graph $G$ are connected if a $u-v$ path exists in $G$. A non-empty graph $G$ is connected if every two vertices of $G$ are connected; otherwise $G$ is disconnected. If $V' \subseteq V(G)$ and $G[V']$ is connected we also say that $V'$ itself is connected in $G$.
A connected subgraph $G'$ of a graph $G$ is called a component of $G$ if $G'$ is not contained in any connected subgraph of $G$ having more vertices or edges than $G'$. A vertex $v$ in a connected graph $G$ is called a cut-vertex if $G - v$ is disconnected. An edge $e$ in a connected graph is called a bridge if $G - e$ is a disconnected graph. Figure 2.5 shows an example of a cut-vertex and a bridge in a graph $G$.

[4] In [Die05] $P^n$ is a path of length $n$.

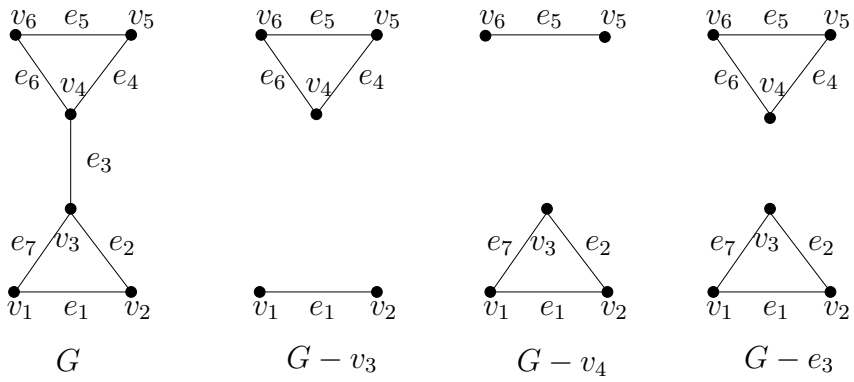



Figure 2.5: The edge $e_3$ is the only bridge in the connected graph $G$ and removing it leaves $G$ disconnected. $v_3$ and $v_4$ are two cut-vertices because $G - v_3$ and $G - v_4$ are disconnected graphs.

# 2.6 Matrix Representation of Graphs

## 2.6.1 Adjacency and Incidence Matrices

**Definition 2.6.1** *The adjacency matrix of a graph $G$ with $n$ vertices $v_1, v_2, \ldots, v_n$ is the $n \times n$ matrix $A{=}A(G)$ whose entries $a_{ij}$ are given by*

$$a_{ij} = \begin{cases} 1 & \text{if } v_i \text{ and } v_j \text{ are adjacent;} \\ 0 & \text{otherwise.} \end{cases} \tag{2.4}$$

It follows directly from the definition that $A$ is a real symmetric matrix, and that the trace of $A$ is zero. Since the rows and the columns of $A$ correspond to an arbitrary labelling of the vertices of $G$, it is clear that properties of the adjacency matrix which are invariant under permutation of rows and columns, such as the spectral properties of $A$, are of most importance.

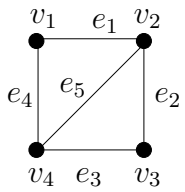
In fact the adjacency matrix of a graph $G$ contains all the information about the vertices in $G$. We also define a matrix which identifies all the edges and their incident vertices in $G$. This matrix is called the incidence matrix of $G$.

**Definition 2.6.2** *The incidence matrix of a graph $G$ with $n$ vertices and $m$ edges is the $n \times m$ matrix $B{=}B(G)$ whose entries $b_{ij}$ are given by*

$$b_{ij} = \begin{cases} 1 & \text{if } v_i \text{ is incident with } e_j; \\ 0 & \text{otherwise.} \end{cases} \tag{2.5}$$

In general the incidence matrix is not symmetric and its entries depend on how we label both vertices and edges.

**Theorem 2.6.1** *Consider a graph $G$ of order $n$ with an adjacency matrix $A = (a_{ij})$ and an incidence matrix $B = (b_{ij})$. Suppose that $D$ is an $n \times n$ diagonal matrix with $D_{ii}{=}d(v_i)$, the*




$$A = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix} \qquad B = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \end{pmatrix}$$

Figure 2.6: A (4,5) graph $G$ with its adjacency matrix $A$ and its incidence matrix $B$ are given.

*valency of the vertex $v_i$ in $G$. Then*

$$BB^T = A + D \tag{2.6}$$

*Proof.* We have

$$(BB^T)_{ij} = \sum_{s=1}^{n} b_{is}\ b_{js}.$$

Hence, for $i = j$ the entry of $(BB^T)$ represent the valency of the vertex $v_i$ i.e., $(BB^T)_{ii} = d(v_i)$. For $i \neq j$ the entry of $(BB^T)$ is 1 iff $v_i$ and $v_j$ are adjacent and 0 otherwise. ⊡

In particular, if $G$ is a k-regular graph, equation 2.6 takes the following form.

$$BB^T = A + k\ \mathbf{I}.$$

where **I** is the identity matrix.

### 2.6.2 Spectrum and Cospectral Graphs

Let $\lambda$ be an eigenvalue of $A$. Then, since $A$ is real and symmetric, it follows that $\lambda$ is real, and the multiplicities of $\lambda$ as a root of the equation $\det(\lambda I - A) = 0$ is equal to the dimension of the space of eigenvectors corresponding to $\lambda$.

**Definition 2.6.3** *The spectrum of a graph $G$ is the set of eigenvalues of $A(G)$, together with their multiplicities as eigenvalues of $A(G)$. If the distinct eigenvalues of $A(G)$ are $\lambda_0 > \lambda_1 > \ldots > \lambda_{s-1}$, and their multiplicity are $m(\lambda_0), m(\lambda_1), \ldots, m(\lambda_{s-1})$, then we shall write*

$$\text{Spec } (G) = \begin{pmatrix} \lambda_0 & \lambda_1 & \ldots & \lambda_{s-1} \\ m(\lambda_0) & m(\lambda_1) & \ldots & m(\lambda_{s-1}) \end{pmatrix} \tag{2.7}$$

It is common to refer to the eigenvalues of $A(G)$ as the eigenvalues of $G$. The characteristic polynomial of a graph $G$, denoted as $\chi(G;\lambda)$, is the polynomial $\det(A$ -$\lambda I)$, where $A$ is the adjacency matrix and $I$ is the identity matrix [Big93]. The following example gives the spectrum of the complete graphs of order 2, 3, and 4.

**Example 2.6.1** *Let $A_2$, $A_3$, and $A_4$ represent the adjacency matrices of the complete graphs $K^2$, $K^3$, and $K^4$ shown in Figure 2.2 respectively.*

$$A_2 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad A_3 = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix} \quad A_4 = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

*It is not difficult to show that the corresponding characteristic polynomials are $\chi(K^2;\lambda) = (\lambda - 1)\ (\lambda + 1)$, $\chi(K^3;\lambda) =$ - $(\lambda - 2)\ (\lambda + 1)^2$, and $\chi(K^4;\lambda) = (\lambda - 3)\ (\lambda + 1)^3$. and the corresponding spectrums are*

$$\textit{Spec}\ (K^2) = \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix} \quad \textit{Spec}\ (K^3) = \begin{pmatrix} 2 & -1 \\ 1 & 2 \end{pmatrix} \quad \textit{Spec}\ (K^4) = \begin{pmatrix} 3 & -1 \\ 1 & 3 \end{pmatrix}$$

Two non-isomorphic graphs are said to be cospectral if they have the same eigenvalues with the same multiplicities. An example of two cospectral graphs is given in Figure 2.7.

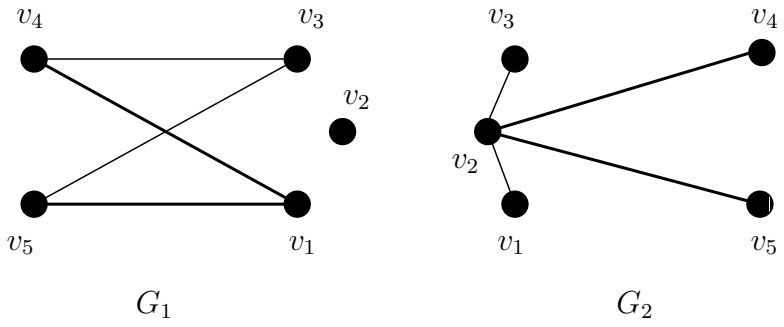



Figure 2.7: $G_1$ and $G_2$ are two non-isomorphic graphs that share the same spectrum.

The adjacency matrices for $G_1$ and $G_2$ are denoted by $A_1$ and $A_2$ respectively.

$$A_1 = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \end{pmatrix} \quad A_2 = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

It is easy to show that the characteristic polynomial of both $G_1$ and $G_2$ is $-\lambda^5 + 4\lambda^3$. It follows that $G_1$ and $G_2$ have the same spectrum, and since the two graphs are clearly non-isomorphic, $G_1$ and $G_2$ are two cospectral graphs. ‘

# 3. $\sigma$-rule on Complete Graphs

Let $K^n$ be a complete graph of order $n$ on a set of vertices $V$. Suppose each of the vertices of $K^n$ is equipped with an indicator light and a button and these lights follow the following switching rule: if the button corresponding to a vertex $v$ is pressed, the light associated with all the vertices in the neighbourhood of $v$ will change their states. This rule will be called $\sigma$-rule. The All-Ones problem on complete graphs can be stated as follows: If all the lights are intially $OFF$ is it possible to press a sequence of buttons such that all the lights are $ON$ at the end. More generally, let $X$ and $Y$ be two different situations of lights in a complete graph $K^n$. A possible generalisation of the All-Ones problem on complete graphs is to find a sequence of buttons by which we can convert $X$ to $Y$. This chapter provides a complete solution for the All-Ones problem on complete graphs in its general form.

## 3.1 Introduction

A possible algebraic representation of the All-Ones problem on a complete graph is to represent a situation on a complete graph $K^n$ of order $n$[1] by a vector $X = (a_1, a_2, a_3, \ldots, a_n)$, where $a_i \in \{0, 1\}$. More explicitly, $a_i = 0$ if the light at the vertex $i$ is $OFF$, and $a_i = 1$ if the light at the vertex $i$ is $ON$. So all possible situations can be represented as a set $\mathcal{X}_n = \{(a_1, a_2, a_3, \ldots, a_n) : a_i = 0, 1\}$, and in this way the operation of a switch is to convert a given situation to another. Working with integers modulo 2 helps in defining a switch operation. According to the switching rule in this problem, for a given situation $X$ pressing a button $i$ is equivalent to changing the states of all the lights in its neighbourhood, and algebraically this is equivalent to adding 1 modulo 2 to every other component in the vector state $X$. Let us define the basis $e_1 = (1, 0, 0, 0, \ldots, 0), e_2 = (0, 1, 0, 0, \ldots, 0), \ldots, e_n = (0, 0, \ldots, 1)$, and a vector $E = e_1 + e_2 + \ldots + e_n$, i.e. $E = (1, 1, 1, \ldots, 1)$. In fact $E + e_i$ represents the row $i$ in the adjacency matrix of $K^n$. Let us also define $P_i$ as an operator that changes that states of all the lights in $K^n$ except the light associated with the vertex $v_i$. It is clear that pressing the $i^{th}$ button is equivalent to adding the $i_{th}$ row $e_i + E$ of the adjacency matrix of $K^n$ modulo 2.
For example pressing the first button in a given situation $X$ is represented as follows

$$\begin{aligned} X = (a_1, a_2, \ldots, a_n) &\xmapsto{P_1} (a_1, a_2, a_3, \ldots, a_n) + (e_1 + E) \qquad \mod 2 \\ &= (a_1, a_2 + 1, a_3 + 1, \ldots, a_n + 1) \qquad \mod 2 \end{aligned} \tag{3.1}$$

In our matrix representation, finding a sequence of buttons by which a situation $X$ can be converted to a situation $Y$ is equivalent to solving the linear equation

$$A.K = Y - X \qquad \mod 2, \tag{3.2}$$

for the indeterminate vector $K = (k_1, \ldots, k_n)$ where $k_i = 1$ if we press the $i^{th}$ button and $k_i = 0$ otherwise, and $A$ is the adjacency matrix of $K^n$. Given the fact that the addition of vectors is

[1] We say that a graph $K^n$ is an even graph if $K^n$ has even order, and $K^n$ is an odd graph if $K^n$ has an odd order.

a commutative operation, the order of pressing the buttons does not affect the final result. Also one of the consequences of working modulo 2 is that if we press a button an even number of times it is exactly equivalent to not pressing the button at all, and pressing a button an odd number of times is exactly equivalent to pressing the button once. So in order to convert any state $X$ to any other state $Y$ it is sufficient to use each button at most once [Bea06] [Sut89]. We will say that a given situation $X$ is even if the total number of $ON$ lights is $N = 2m,\ m \in \mathbb{Z}$, and a given situation $Y$ is odd if the total number of $ON$ lights is $N = 2m + 1,\ m \in \mathbb{Z}$.

## 3.2 Even Complete Graphs

### 3.2.1 Motivation

For a situation $X$ in an even $K^n$ graph, we can change the state of a specific light by pressing the buttons that are associated with all the other lights. This is true because when we press an odd number buttons in an even situation $X$ that will change the states of the lights which we did not touch its associated buttons. Since this is true for any light we can convert any situation $X$ into any other situation $Y$.[2] In the following section, we will prove this in detail.

### 3.2.2 Solution and Uniqueness

**Proposition 3.2.1** *Let $X$ and $Y$ be two situations in an even $K^n$ graph. In order to convert $X$ to $Y$ we only need to press the set of buttons whose associated lights have different states in $X$ and $Y$ if $X$ and $Y$ have the same parity. Otherwise, we only need to switch the set of buttons whose associated lights have the same states in $X$ and $Y$.*

*Proof.* Let $X$ and $Y$ be two different situations in an even $K^n$ graph.
**1. Same Parity**:

Let a solution be given by a vector $K = (k_1, k_2, \ldots, k_n)$, where $k_i = 1$ if we press the $i^{th}$ button and $k_i = 0$ otherwise. When we press a button $i$ the states of all the lights will change, but not the light that is associated with the button $i$ itself. So pressing any button will change the parity of the situation, because an odd number of lights will change their states, and that changes the parity of the situation. Accordingly to convert a situation $X$ to another situation $Y$ that has the same parity we need to press an even number of buttons i.e.

$$\sum_{i=1}^{n} k_i \equiv 0 \qquad \mod 2, \tag{3.3}$$

and since $(k_1, \ldots, k_n)$ is a solution to the problem, the situation $Y$ can be written as follows:

$$Y = X + k_1(e_1 + E) + k_2(e_2 + E) + \ldots + k_n(e_n + E), \tag{3.4}$$

[2] In fact this is correct only for even complete graphs. See Example A.2.1

where $k_i \in \{0, 1\}$. Hence

$$Y = X + k_1e_1 + k_2e_2 + \ldots + k_ne_n + \sum_{i=1}^{n} k_iE. \tag{3.5}$$

Using equation (3.3) then

$$K = Y - X \mod 2. \tag{3.6}$$

It follows that

$$k_i = \begin{cases} 0 & \text{for } a_i = b_i. \\ 1 & \text{for } a_i \neq b_i \end{cases} \tag{3.7}$$

Hence to convert a situation $X$ to another situation $Y$ that has the same parity we need to press the buttons whose associated lights have different states in $X$ and $Y$, and the solution is unique.[3]

**2. Different Parity**:

By the same argument as in the first case, to convert an even situation to an odd situation (or odd to even) we need to press an odd number of buttons, i.e. for a solution written as a vector $K = (k_1, k_2, \ldots, k_n)$, where $k_i = 1$ if we press the $i^{th}$ button and $k_i = 0$ otherwise,

$$\sum_{i=1}^{n} k_i \equiv 1 \mod 2. \tag{3.8}$$

And since $K$ is a solution, the situation $Y$ can be given as follows.

$$Y = X + k_1(e_1 + E) + k_2(e_2 + E) + \ldots + k_n(e_n + E) \tag{3.9}$$

or

$$Y = X + k_1e_1 + k_2e_2 + \ldots + k_ne_n + \sum_{i=1}^{n} k_iE \tag{3.10}$$

Using equation (3.8) then

$$K + E = Y - X \mod 2 \tag{3.11}$$

modulo 2 the vector $E$ acts as a not operator i.e it inverts every 0 to 1 and every 1 to 0. Hence

$$k_i = \begin{cases} 1 & \text{for } a_i = b_i. \\ 0 & \text{for } a_i \neq b_i \end{cases} \tag{3.12}$$

Therefore to convert a situation $X$ to another situation $Y$ that has different parity we need to press the buttons whose associated lights have the same states in $Y$ and $X$. ⊡

**Corollary 3.2.1** *For even $K^n$ graphs the All-Ones problem has a unique solution.*

*Proof.* Corollary 3.2.1 is a direct consequence of Proposition 3.2.1. The all-$OFF$ situation is an even situation, and for an even $K^n$ graph, the all-$ON$ situation is an even situation. So for even $K^n$ graphs the All-Ones problem has a solution, and the solution is unique. ⊡

[3] See Example A.2.2 and Example A.2.3 for more illustration

# 3.3 Odd Complete Graphs

## 3.3.1 Motivation

For an odd $K^n$ graph the All-Ones problem has no solution because for any given situation $X$ pressing any combination of buttons will not change the parity of the situation $X$. But in fact we can convert any situation $X$ to any other situation $Y$ provided that $X$ and $Y$ have the same parity, and there are exactly two solutions.

## 3.3.2 Parity Conservation

**Proposition 3.3.1** *For a situation $X$ in an odd $K^n$ graph the parity of $X$ is conserved under pressing any combination of buttons.*

*Proof.* In an odd $K^n$ graph we have $n$ lights. If a button is pressed, $n-1$ lights change their states. Since $n-1$ has to be even, this preserves the parity of the situation. ⊡

**Proposition 3.3.2** *For an odd $K^n$ graph to convert a situation $X$ to another situation $Y$ that has the same parity we have exactly two solutions: either to press the set of buttons whose associated lights have the same states in $X$ and $Y$ or equivalently to press the set of buttons whose associated lights have different states in $X$ and $Y$.*

*Proof.* Let $X$ and $Y$ be two different situations in an odd $K^n$ graph. Since pressing an even number of buttons in an odd $K^n$ graph will only change the states of the lights that are associated with the buttons we pressed, and since $X$ and $Y$ have the same parity, the set of lights which have different states in $X$ and $Y$ has even cardinality, hence the set of buttons whose associated lights have different states in $X$ and $Y$ is a solution.

Generally, let a solution be given by a vector $K = (k_1, k_2, \ldots, k_n)$, where $k_i = 1$ if we press the $i^{th}$ button and $k_i = 0$ otherwise. Suppose that a solution $K$ satisfies

$$\sum_{i=1}^{n} k_i \equiv 0 \mod 2. \tag{3.13}$$

Since $K$ is a solution $Y$ can be written as

$$Y = X + k_1(e_1 + E) + k_2(e_2 + E) + \ldots + k_n(e_n + E)$$

$$= X + k_1e_1 + k_2e_2 + \ldots + k_ne_n + \sum_{i=1}^{n} k_iE \tag{3.14}$$

Using equation (3.13)

$$K = Y - X \mod 2, \tag{3.15}$$

Hence for an odd $K^n$ graph, to convert a situation $X$ to another situation $Y$ that has the same parity we can switch the buttons whose associated lights have different states in $X$ and $Y$, this is the only solution that satisfies equation (3.13).

On the other hand, since pressing an odd number of buttons in an odd $K^n$ graph will change only the states of the lights which are associated with the buttons we have not pressed, and the set of lights which has the same states in $X$ and $Y$ is always a set of odd cardinality, hence the set of buttons whose associated lights have the same states in $X$ and $Y$ is a solution.
To prove uniqueness, let us suppose that the solution $K$ satisfies

$$\sum_{i=1}^{n} k_i \equiv 1 \qquad \mod 2. \tag{3.16}$$

So we can write the situation $Y$ as follows

$$Y = X + k_1(e_1 + E) + k_2(e_2 + E) + \ldots + k_n(e_n + E)$$

$$Y = X + k_1e_1 + k_2e_2 + \ldots + k_ne_n + \sum_{i=1}^{n} k_iE \tag{3.17}$$

Using equation (3.16),

$$K + E = Y - X \mod 2 \tag{3.18}$$

Therefore for an odd $K^n$ graph, to convert a situation $X$ to another situation $Y$ that has the same parity we can also switch the buttons whose associated lights have the same states in $X$ and $Y$, and this is the only solution that satisfies equation (3.16). ⊡

**Corollary 3.3.1** *For odd $K^n$ graphs the All-Ones problem has no solution.*

*Proof.* The same argument also applies for the odd case. The all-$OFF$ situation is an even situation, and for odd $K^n$ graphs, the all-$ON$ situation is an odd situation. In this case, according to Proposition 3.3.1 for odd $K^n$ graphs the All-Ones problem has no solution. ⊡

**Theorem 3.3.1** *The adjacency matrix of $K^n$ graphs has full rank iff $n$ is even.*

*Proof.* According to Proposition 3.2.1, in the case of even $K^n$ graphs we have exactly one solution satisfies equation (3.2) which means that the adjacency matrix of even complete graphs is invertible or equivalently has full rank. However, in the case of odd $K^n$ graphs, according to Proposition 3.3.1 and Proposition 3.3.2 equation (3.2) either has two solutions or no solution exists which means that the adjacency matrix of odd complete graphs is singular or equivalently does not have full rank. Hence the adjacency matrix of $K^n$ graphs has full rank if and only if $n$ is even. ⊡

# 4. $\sigma^+$-rule on General Graphs

A more general form of the All-Ones problem on complete graphs is to consider the All-Ones problem on general graphs[1]. In this chapter we will approach this problem using the $\sigma^+$-rule; that is, for a graph $G$ with $n$ vertices, if a button of a vertex $v$ is pressed, the state of all the lights in the closed neighbourhood of $v$ will change.

## 4.1 Introduction

Stating the $\sigma^+$-rule in a different way, we come to the conclusion that a solution set $X$ must have the following properties

- Any vertex $v \in X$ should have an even number of vertices $\in X$ in its neighbourhood.
- Any vertex $v \notin X$ should have an odd number of vertices $\in X$ in its neighbourhood.

These two properties are sufficient for a set $X$ to be a solution to the All-Ones problem with $\sigma^+$-rule [Sut89], i.e. a set $X$ is a solution to the All-Ones problem with $\sigma^+$-rule for a general graph $G$ if and only if for every vertex $v$ the number of vertices in $X$ in the closed neighbourhood of $v$ is odd.

## 4.2 Algebraic Terminology

In principle, we can perform a brute-force[2] search over all the possible configuration of a graph $G$ to determine a solution or to prove it does not exist. But in fact this method is not too useful in reality, because for example the search space of an $10 \times 10$ grid is $2^{100} \simeq 1.27 \times 10^{30}$. And if we can perform $10^6$ per a second it means we need approximately $4.08 \times 10^{16}$ years to find out the solutions, so we need another tool to solve this problem.

Let us consider $G^n$ graph with a set of vertices $V = \{v_{11}, v_{12}, \ldots, v_{nn}\}$. In an algebraic terminology we can consider each configuration of $G^n$ as an $n \times n$ matrix $C$ with entries in $\mathbb{F}_2$, and pressing a button of a vertex $v_{ij}$ is equivalent to the matrix addition $C + T$, where $T$ is the matrix in which the only entries equal to 1 are those corresponding to the vertices in the closed neighbourhood of $v_{ij}$. There are three different types of matrices $T$ depending on the button we press whether it represents a corner-vertex, an outer-vertex, or an inner-vertex. We know that the order in which the buttons are pressed does not matter and pressing a button twice is the same as not pressing at all. Let $A$ be the adjacency matrix of $G^n$, so we define the closed neighbourhood matrix $M = A + I$, where $I$ is the identity matrix. Algebraically the All-Ones problem with $\sigma^+$-rule can be represented by a linear system

$$\mathbf{0} + M.K = \mathbf{1}, \tag{4.1}$$

[1] Other interesting problems related to the All-Ones problem on graphs is given in [LZ05].

[2] A mathematical algorithm that exhausts all the elements in the search space.

$$(A + I).K = \mathbf{1}, \tag{4.2}$$

where $\mathbf{0}$ is the vector with all components equal to 0, and $\mathbf{1}$ is the vector with all components equal to 1, and $K$ is a column vector with indeterminate entries. So solving the All-Ones problem with $\sigma^+$-rule[3] would be to solve the linear system in the indeterminate $K$ over the field $\mathbb{F}_2$.

In fact this method involving $n^2$ equations is still not applicable for large values of $n$. In principle we can find the solution for any grid graph by writing an algorithm based on the methods given in this section, but we still need to prove mathematically that a solution of All-Ones problem exists for any grid graph. Let us first reformulate the problem in terms of Cellular Automata.

## 4.3 Cellular Automata and Algebraic Properties of the $\sigma^+$-rule

### 4.3.1 Notations

**Definition 4.3.1** *A cellular automaton is a discrete dynamical system that consists of an arrangement of basic components called cells together with a transition rule.*

Generally every cell possesses a finite number of possible states. For the purpose of our study we will consider a cellular automaton with cells of only two states, namely 0 and 1, corresponding to $OFF$ and $ON$ states for each vertex $v$ in a graph $G$, and the local transition rule which defines the state of any cell $i$ at time $t+1$ depending on the state of the cells in the closed neighbourhood of $i$. Now let us consider a locally finite graph $G = (V, E)$, a graph in which every vertex $v \in G$ has a finite number of vertices in its neighbourhood. The number of the neighbours of a vertex $v$ for a grid graph is completely defined for each vertex $v$ according to the type of the vertex, i.e corner-vertex, outer-vertex or inner vertex. In order to identify the state of each vertex in $G$ consider the following definition

**Definition 4.3.2** *A pattern $X$ which represents the states of all the vertices in a graph $G$ can be defined as a function*

$$X : V \longrightarrow \{0, 1\} \tag{4.3}$$

This function takes a vertex from the domain of all vertices $V$ and assigns a value from the range which is either 0 or 1. $X$ can be identified with a subset of the vertex set $V$ in which a vertex $v \in X$ if and only if $X(v) = 1$.

In terms of an automaton, the $\sigma^+$-rule can be stated as[4]

$$\sigma^+(X)(v) = \sum_{u \in N_v} X(u) \mod 2 \tag{4.4}$$

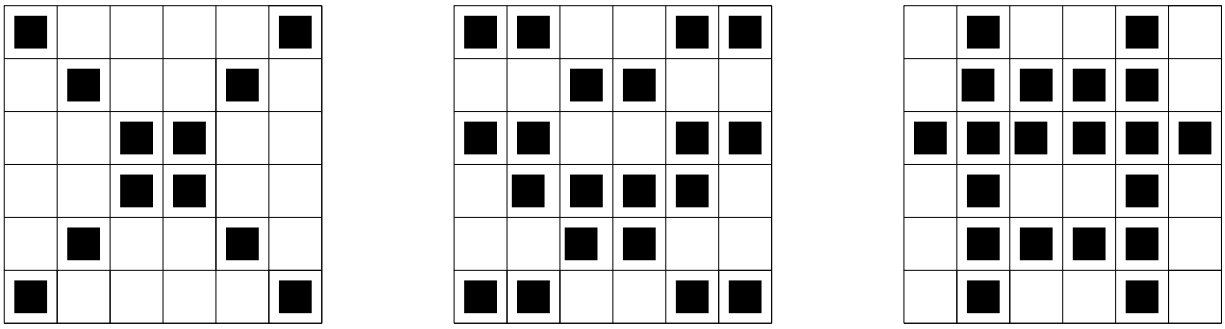
[3] In the case of the $\sigma$-rule we were considering the adjacency matrix $A$ instead of the closed neighbourhood matrix $M$, see Section 3.1

[4] In the case of the $\sigma$-rule the summation would be over the neighbourhood instead of the closed neighourhood of the vertex $v$

This means for a given pattern $X$ the state of a cell $v$ at $t+1$ is effectively determined by the state of all the vertices in its closed neighbourhood at time $t$, namely it is 0 if the closed neighbourhood $N_v$ has an even number of cells in state 1, and it is 1 if $N_v$ has an odd number of cells in state 1. A graph together with the $\sigma^+$-rule is called a $\sigma^+$-automaton [Sut89].

### 4.3.2 Pattern Space

Let us now define a pattern space, that is; a vector space whose elements are all the possible patterns in $C_G$ over the field $\mathbb{F}_2$. The set of all singletons $\{v\}$ $(v \in V)$ gives a basis in the pattern space. So if $X$, $Y \in C_G$ then $X+Y \in C_G$, and for every $\lambda \in \mathbb{F}_2$, $\lambda X \in C_G$, see Figure 4.1. The additive inverse of a pattern $X$ in the pattern space $C_G$ is the pattern $X$ itself.



$X$ $Y$ $X+Y$

Figure 4.1: Addition of two elements $X$ and $Y$ in the pattern space $C_G$ gives another element $Z$ in the pattern space $C_G$

For any patterns $X, Y \in C_G$ such that

$$\sigma^+(Y) = X, \tag{4.5}$$

the pattern $Y$ is called predecessor of $X$ under the rule $\sigma^+$. We will call a pattern which does not have any predecessors a Garden-of-Eden. Solving the All-Ones problem on a graph $G$ is equivalent to solving the linear equation

$$\sigma^+(X) = \mathbf{1}, \tag{4.6}$$

and finding out all the possible predecessors of this equation if there are any. And the question to be answered is whether $\mathbf{1}$ a Garden-of-Eden for a given $\sigma^+$-automaton. Let $x$ be a column vextor associated to a pattern $X \in C_G$ such that $x_i = X(v_i)$, and $y$ is the vector corresponding to $\sigma^+(X)$. So according to the discussion in section 4.2, application of the $\sigma^+$-rule on a pattern $X$ is equivalent to a matrix multiplication

$$y = M.x = (A+I).x \tag{4.7}$$

where $M$ is the closed neighbourhood matrix.[5] The rules $\sigma^+$ and $\sigma$ are linear rules (see Figure 4.2), so for $X$, $Y \in C_G$, we have

$$\sigma^+(X+Y) = \sigma^+(X) + \sigma^+(Y). \tag{4.8}$$

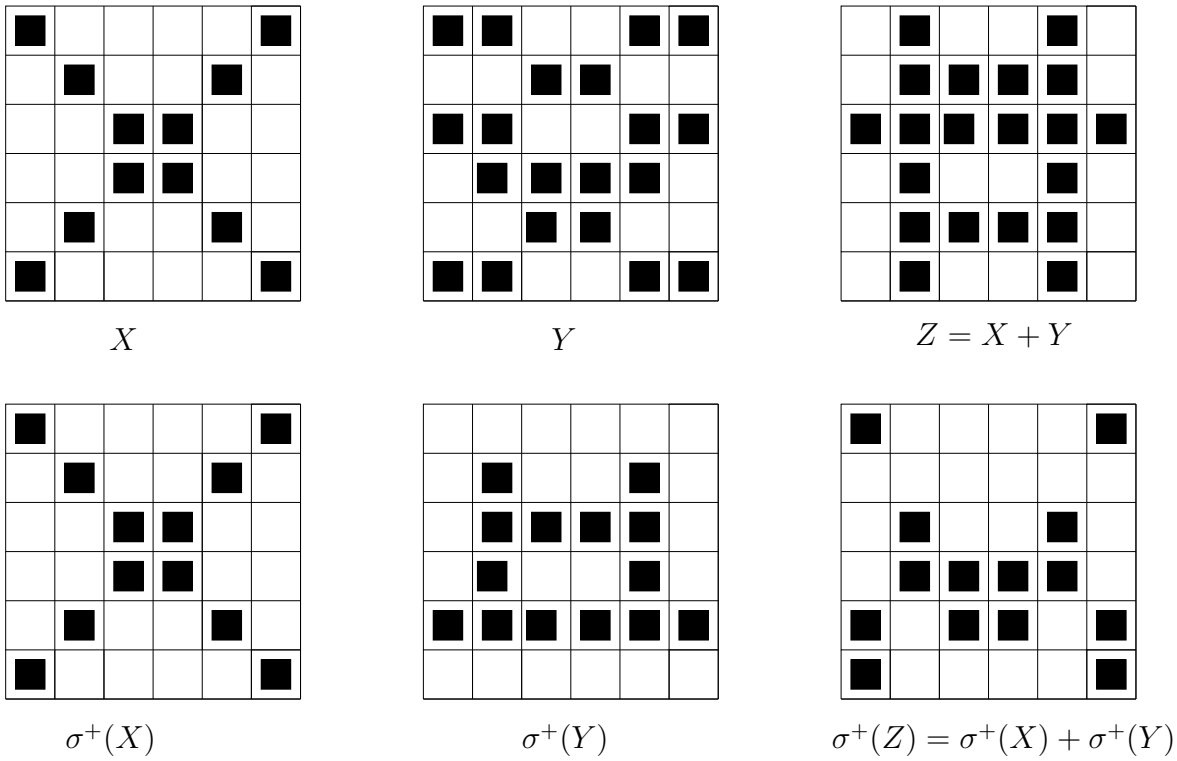



Figure 4.2: An illustration of the linearity of the $\sigma^+$-rule.

### 4.3.3 Nullspace and Reversibility of $\sigma^+$-automata

The nullspace or the kernel of the linear rule $\sigma^+$ on a finite graph $G$ can be defined as follows

**Definition 4.3.3** *For a finite graph* [6] $G$ *the kernel or the nullspace of the rule* $\sigma^+$ *on* $G$ *is*

$$K_{G,\sigma^+} = \{X \in C_G : \sigma^+(X) = 0\} \tag{4.9}$$

$K_{G,\sigma^+}$ is a subspace of $C_G$ and it satisfies the following properties. For any $X, Y \in K_{G,\sigma^+}$, so that $\sigma^+(X)= \sigma^+(Y)=\mathbf{0}$, and any $\lambda_1$, $\lambda_2 \in \mathbb{F}_2$

$$\sigma^+(\lambda_1 X + \lambda_2 Y) = \lambda_1\sigma^+(X) + \lambda_2\sigma^+(Y) = \mathbf{0}, \tag{4.10}$$

[5] For the $\sigma$-rule the closed neighbourhood matrix $M$ would be replaced by the adjacency matrix $A$.
[6] See section 2.2

which implies that $\lambda_1 X + \lambda_2 Y \in K_{G,\sigma^+}$ [OS06]. The dimension of the kernel $K_{G,\sigma^+}$ which represents the number of basis elements of the nullspace of $\sigma^+$ on $G$ will be denoted as $d(K_{G,\sigma^+})$. $d(K_{G,\sigma^+})$ is also called the parity dimension of $G$ [ACS98].

Furthermore, the range or the image of the rule $\sigma^+$ over the pattern space $C_G$ can be defined by

$$R\,(\sigma^+) = \{\sigma^+(X) : X \in C_G\}. \tag{4.11}$$

For a given $\sigma^+$-automaton if the linear rule $\sigma^+$ is reversible or equivalently bijective, equation (4.2) will have a unique solution and it is possible to find a predecessor pattern $Y$ of the pattern $X$. In spite of the fact that the rules $\sigma$ and $\sigma^+$ are locally irreversible [7], they may be well globally reversible [Sut89]. $\sigma^+$ is defined as a linear map from $C_G$ to $C_G$ itself, so $\sigma^+$ is reversible if and only if each patten in its range has a predecessor.

## 4.4 Paths, Cycles, and Grid graphs

### 4.4.1 Predecessors and Transition Diagrams

Let $X$ and $Y$ be two different predecessors of the pattern $\mathbf{1}$, then $X + Y$ is in $K_{G,\sigma^+}$.

$$\begin{aligned} \sigma^+(X+Y) &= \sigma^+(X) + \sigma^+(Y) \\ &= \mathbf{1} + \mathbf{1} = \mathbf{0} \end{aligned} \tag{4.12}$$

For any $Z \in K_{G,\sigma^+}$ if $X$ is a predecessor of $\mathbf{1}$, then $X + Z$ is also a predecessor of $\mathbf{1}$.

$$\begin{aligned} \sigma^+(X+Z) &= \sigma^+(X) + \sigma^+(Z) \\ &= \mathbf{1} + \mathbf{0} = \mathbf{1} \end{aligned} \tag{4.13}$$

So for a given graph $G$ the number of predecessors of a pattern $X$ is either 0 or $2^{d(K_{G,\sigma^+})}$.
The transition diagram $T_{G,\sigma^+}$ of the $\sigma^+$-rule on a finite graph $G$ which illustrates the predecessor relation of the $\sigma^+$-rule on $G$ is a directed graph in which the patterns in $G$ represent the vertices in $T_{G,\sigma^+}$, and a directed edge from $X$ to $Y$ exists in $T_{G,\sigma^+}$ if and only if $\sigma^+(X) = Y$. The in-degree of every vertex in $T_{G,\sigma^+}$ is either 0 or $2^{d(K_{G,\sigma^+})}$, and the out-degree of any vertex in $T_{G,\sigma^+}$ is exactly 1. Figure 4.3 gives the transition diagram $T_{P_5,\sigma^+}$.

### 4.4.2 Odd-Parity Covers for $P^n$- and $C^n$-graphs

Any vertex $v$ in a $P^n$ graph has exactly three elements in the closed neighbourhood $N_v$ except the end vertices 1 and $n$ - they have two neighbours. Given those properties of $P^n$ graphs working modulo 3 is helpful solving the All-Ones problem for $P^n$ graphs. In fact an odd-parity cover always exists for $P^n$ graphs for any arbitrary $n$.

[7] Two different patterns or more can lead to the same state at a vertex $v$ at $t+1$.

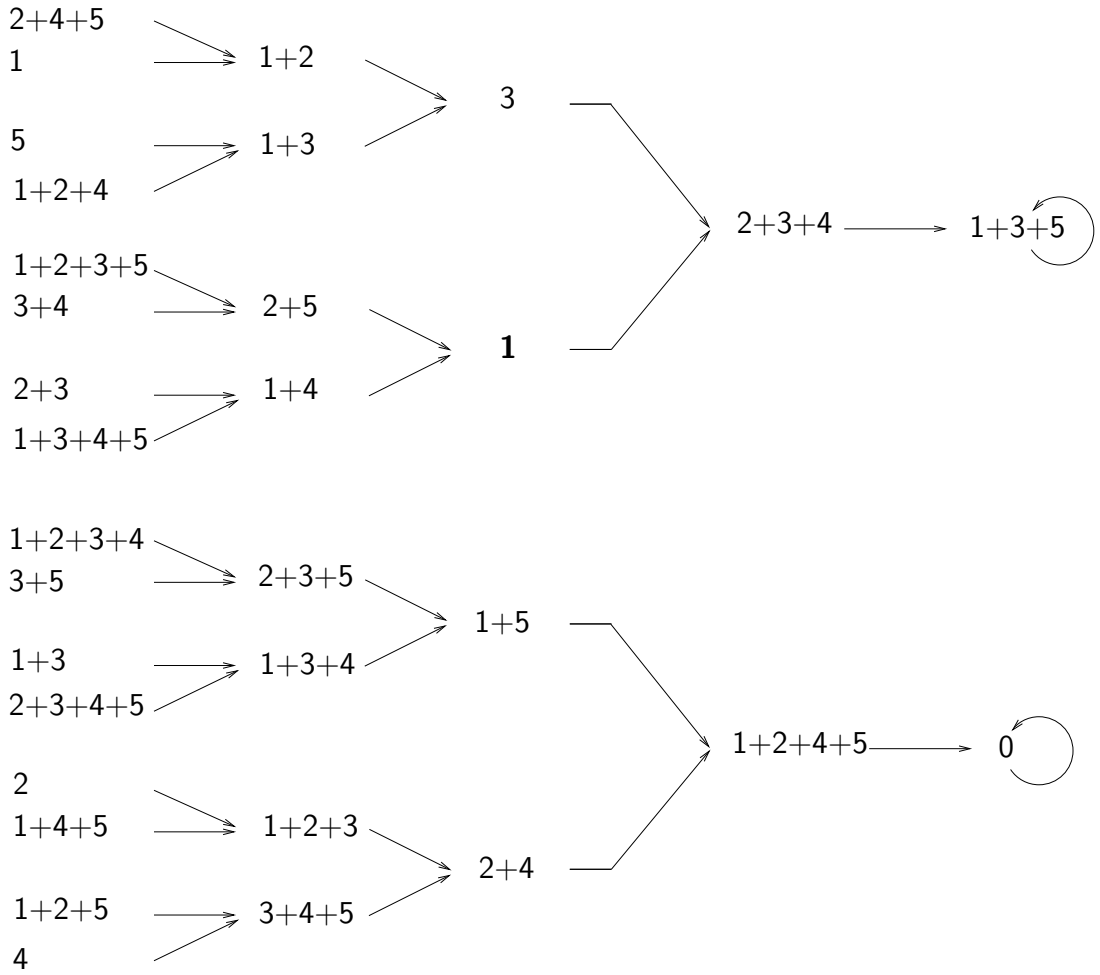



Figure 4.3: The transition Diagram $T_{P^5,\sigma^+}$ demonstrates the predecessor relation for the $P^5$ graph.

Table 4.1: The Odd Parity Covers for $P^n$ graphs.

| Case | $n$ | Number of solutions | Odd Parity Cover |
|---|---|---|---|
| 1 | $n \equiv 0 \mod 3$ | 1 | $2 + 5 + \ldots + (n-4) + (n-1)$ |
| 2 | $n \equiv 1 \mod 3$ | 1 | $1 + 4 + 7 + \ldots + (n-3) + n$ |
| 3 | $n \equiv 2 \mod 3$ | 2 | $1 + 4 + 7 + \ldots + (n-4) + (n-1)$ |
| | | | and $2 + 5 + 8 + \ldots + (n-3) + n$ |

In Table 4.1 the number of solutions are given for arbitrary $n$ and the odd-parity covers are provided as well, and Figure 4.4 gives a graphical representation of the solutions to the All-Ones problem for $P^n$ graphs.

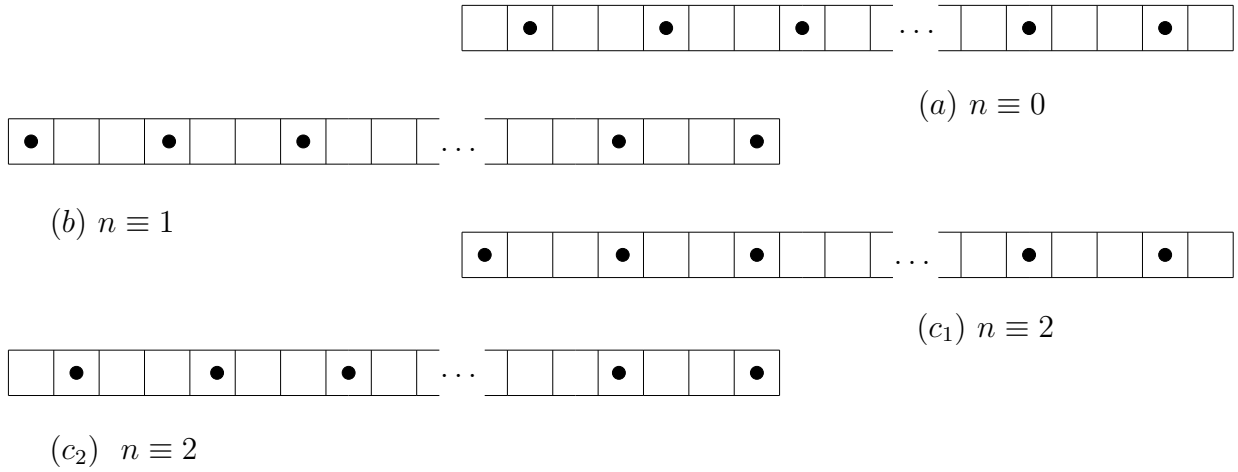



Figure 4.4: (a) and (b) represent the the cases in which the $\sigma^+$-rule is reversible for the graph $P^n$. $(c_1)$ and $(c_2)$ represent the case in which the $\sigma^+$-rule is irreversible for the graph $P^n$.

$C^n$ graphs also have the property that any vertex $v$ in $C^n$ has exactly three neighbour vertices in the closed neighbourhood $N_v$ of $v$, thus working modulo 3 again proves very helpful in solving the All-Ones problem for cycles. Table 4.2 gives the odd-parity covers for $C^n$ graphs for arbitrary $n$. The pattern $\mathbf{1}$ for $C^n$ graphs is a fixed pattern under the $\sigma^+$-rule, i.e. $\sigma^+(\mathbf{1}) = \mathbf{1}$ for arbitrary $n$.

Table 4.2: The odd-parity covers for $C^n$ graphs.

| Case | $n$ | Number of solutions | Odd parity cover |
|---|---|---|---|
| 1 | $n \not\equiv 0 \mod 3$ | 1 | $\mathbf{1}$ |
| 2 | $n \equiv 0 \mod 3$ | 4 | $\mathbf{1}$, $1+4+\ldots+(n-2)$, $2+5+(n-1)$, $3+6+\ldots+n$ |

A graphical representation of the solutions of the All-Ones problem for $C^n$ graphs is given in Figure 4.5.

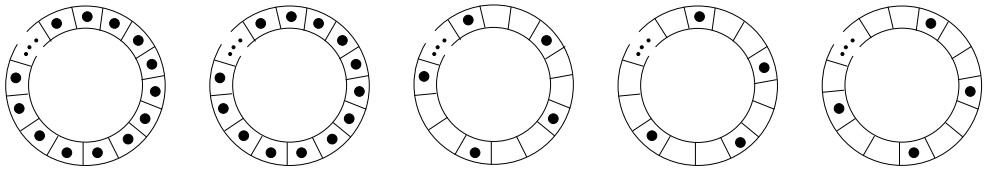



Figure 4.5: $(a)$ shows the predecessor of $\mathbf{1}$ for cycles in which $\sigma^+$ is reversible. $b_1, b_2, b_3$, and $b_4$ show the predecessors of $\mathbf{1}$ for cycles in which $\sigma^+$ is irreversible .

### 4.4.3 Grid Graphs and Parity Dimensions

In this section, we provide solutions for the All-Ones problem on $G^n$ graphs of some selected sizes.[8] In table 4.4.3, the parity dimension of $G^n$ graphs ($n \leq 99$) and the number of corresponding predecessors of the pattern **1** is given in all irreversible cases. The minimum All-Ones problem is the one in which we search for the minimum number of buttons by which the All-Ones problem can be solved. The minimum All-Ones problem for trees is discussed in [CLWZ04] and [LZ05]. Table 4.4.3 shows the minimum number of buttons by which the All-Ones problem on $G^n$ graphs can be solved for some values of $n$ in Min column.

Table 4.3: The parity dimension $d(K_{G,\sigma^+})$ and the number of predecessors $G^n$ graphs (irreversible cases only).

| $n$ | $d(K_{G,\sigma^+})$ | Number of predecessors | Min | $n$ | $d(K_{G,\sigma^+})$ | Number of predecessors | Min |
|---|---|---|---|---|---|---|---|
| 4 | 4 | 16 | 4 | 53 | 2 | 4 | 1487 |
| 5 | 2 | 4 | 15 | 54 | 4 | 16 | 1400 |
| 9 | 8 | 256 | 25 | 59 | 22 | 419430 | – |
| 11 | 6 | 64 | 55 | 61 | 40 | 1099511627776 | – |
| 14 | 4 | 16 | 56 | 62 | 24 | 16777216 | – |
| 16 | 8 | 256 | 104 | 64 | 28 | 268435456 | – |
| 17 | 2 | 4 | 147 | 65 | 42 | 4398046511104 | – |
| 19 | 16 | 65536 | 141 | 67 | 32 | 4294967296 | – |
| 23 | 14 | 16384 | 231 | 69 | 8 | 256 | – |
| 24 | 4 | 16 | 270 | 71 | 14 | 16384 | – |
| 29 | 10 | 1024 | 345 | 74 | 4 | 16 | – |
| 30 | 20 | 1048576 | – | 77 | 2 | 4 | – |
| 32 | 20 | 1048576 | – | 79 | 64 | 18446744073709551616 | – |
| 33 | 16 | 65536 | 469 | 83 | 6 | 64 | – |
| 34 | 4 | 16 | 520 | 84 | 12 | 4096 | – |
| 35 | 6 | 64 | 563 | 89 | 10 | 1024 | – |
| 39 | 32 | 4294967296 | – | 92 | 20 | 1048576 | – |
| 41 | 2 | 4 | 891 | 94 | 4 | 16 | – |
| 44 | 4 | 16 | 772 | 95 | 62 | 4611686018427387904 | – |
| 47 | 30 | 1073741824 | – | 98 | 20 | 1048576 | – |
| 49 | 8 | 256 | 1165 | 99 | 16 | 65536 | – |
| 50 | 8 | 256 | 1220 | | | | |

In Figure 4.6 and Figure 4.7, we present the solutions to the All-Ones problem for grid graphs of sizes $\leq 8$. Graphically, Figure 4.6 contains only the solutions for the cases in which $\sigma^+$ has zero nullity. On the other hand we included in Figure 4.7 the solutions in which the kernel of $\sigma^+$ has a dimension different from zero, and in such a case we have more than one solution to the

[8] In Chapter 5 we provide an algorithm which gives the solution to the All-Ones problem for $G^n$ graphs of any size $n$.

All-Ones problem, regarding solutions which differ only by rotation and/or reflection as distinct ones. Sequences related to the All-Ones problem on $G^n$ graphs are provided in [Wei02].[9]

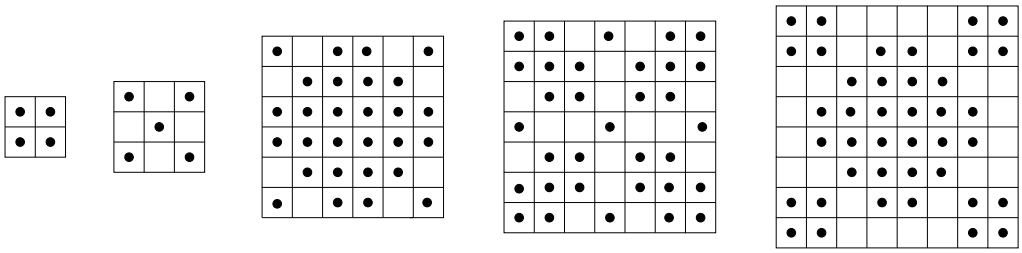


Figure 4.6: Solutions for the All-Ones problem on $G^n$ graphs for $n = 2, 3, 6, 7,$ and $8$.

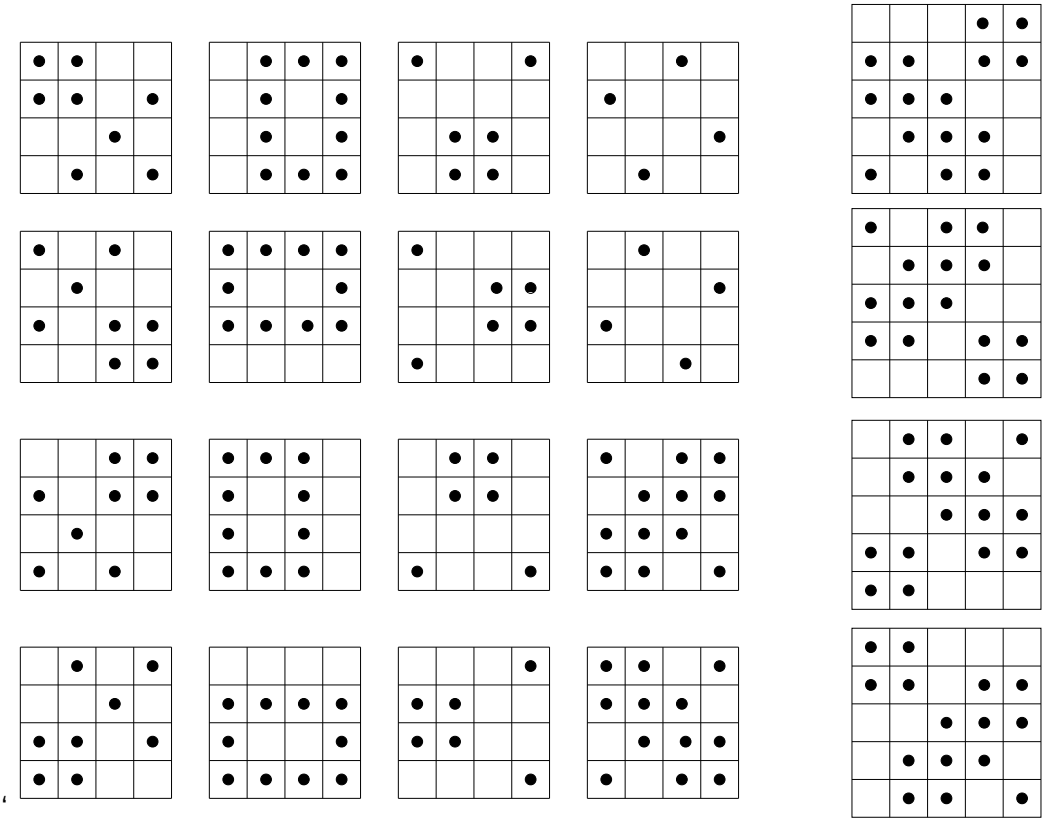


Figure 4.7: Solutions for the All-Ones problem on $G^n$ graphs for $n = 4$ and $5$.

The symmetries appearing in the solutions in both Figure 4.6 and Figure 4.7 make it necessary to mention that the solution for the All-Ones problem that corresponding to reversible $\sigma^+$-automata have to be symmetric under any reflection, rotation or inversion operations. Otherwise the solution will not be unique. In Figure 4.7, it is remarkable that the minimum number of buttons by which the All-Ones problem on $G^4$ graph can be solved is 4, and for $G^5$ is 15.

[9] In [Wei02] the sequence for the minimum number of buttons by which the All-Ones problem on $G^n$ graphs can be solved is provided for sizes up to 25.

## 4.5 The Pattern 1 is not a Garden-of-Eden

Now let us prove algebraically that a solution to the All-Ones problem on a general graph always exists. First let us consider the following theorem.

**Theorem 4.5.1** *[Sut89] Let $G$ be an arbitrary finite graph [10]. A pattern $X$ has a predecessor in $G$ under $\sigma^+$-rule if and only if $X$ is perpendicular to the kernel of $\sigma^+$.*

*Proof.* As stated in section 2.6.1 the adjacency matrix of any graph $G$ is symmetric and so is $A + I$, thus $\sigma^+$ defined as in equation (4.7) is selfadjoint[11], so for any $X, Y \in C_G$

$$< \sigma^+(X), Y >=< X, \sigma^+(Y) > . \tag{4.14}$$

Define $R$ to be the range of $\sigma^+$, and $R^\perp$ to be the set of patterns perpendicular to $R$. So

$$\begin{aligned} Y \in R^\perp \quad & \text{iff for all } X \text{ in } C_G < \sigma^+(X), Y >= 0 \\ & \text{iff for all } X \text{ in } C_G \ < X, \sigma^+(Y) >= 0 \end{aligned} \tag{4.15}$$

But since $< X, Z > = 0$ for all $Z$ in $C_G$ iff $X = 0$, therefore

$$Y \in R^\perp \text{ iff } \sigma^+(Y) = 0 \tag{4.16}$$

The above equation is equivalent to saying that $R^\perp = K_{G,\sigma^+}$ or equivalently

$$R = K^\perp_{G,\sigma^+} \tag{4.17}$$

So $X$ is in the range of $\sigma^+$(or has a predecessor) iff it is perpendicular to $\sigma^+$. ⊡

Theorem 4.5.1 gives the potential for the All-Ones problem to be solved for an arbitrary finite graph. In fact, to prove that the All-Ones problem has a solution all what we need is to prove that the pattern **1** is perpendicular to the kernel of $\sigma^+$.

**Theorem 4.5.2** *[Sut89] The All-Ones problem has a solution in an arbitrary finite graph.*

*Proof.* Any pattern $X$ in the kernel of $\sigma^+$ must satisfy the condition that every vertex $v$ in $X$ is an odd vertex in $G[X]$ . Moreover according to Corollary 2.2.1 for any finite graph $G$ the number of odd vertices is even. Hence any pattern $X$ in the kernel of $\sigma^+$ must have even cardinality. Since $< X, \mathbf{1} >$ represents the cardinality of $X$, for any $X$ in the kernel of $\sigma^+$ $< X, \mathbf{1} >= 0 \mod 2$, which means that **1** is perpendicular to $K_{G,\sigma^+}$, and according to Theorem 4.5.1 **1** has a predecessor under $\sigma^+$ rule for an arbitrary finite graph. ⊡

So the pattern **1** fails to be a Garden-of-Eden under the $\sigma^+$-rule in any finite graph $G$, and the All-Ones problem always has a solution for any finite graph.

[10] This theorem also applies to the $\sigma$-rule

[11] That makes the eigenvalues of $\sigma^+$ over $G$ necessarily real.

# 5. Algorithm and Odd-Parity Covers for Grid Graphs

## 5.1 Introduction

In this chapter we provide an algorithm written in Python which solves the All-Ones problem for $G^n$ graphs of any size, and gives all the solutions[1].[2] Apparently the program can handle successfully closed neighbourhood matrices up to $10404 \times 10404$ after which normal computers run out of memory. As discussed in Chapter 4 the All-Ones problem can be represented algebraically as a linear system $M.K = \mathbf{1}$, where $M$ is the closed neighbourhood matrices of $G^n$ graphs, $K$ is the solution vector to be determined, and $\mathbf{1}$ is a vector with all entries are 1.

## 5.2 Gaussian Elimination and Parity Dimension

### 5.2.1 Closed Neighbourhood Matrix

The first function in the algorithm stands for building up the closed neighbourhood matrix of a $G^n$ graph of a given size $n$. The matrix $M_2$ and $M_3$ given below represent the closed neighbourhood matrix of grid graphs of sizes $2$ and $3$.

$$M_2 = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix} \qquad M_3 = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

The function has the following form

```
(1) This function creates the closed neighbourhood matrix and delivers it
    to Forward_Elimination() function.
from scipy import *
def Adjacency_Matrix():
    print "\n"
    k = input('Please enter the size n of the grid graph: ')
```

[1] In Table 4.4.3 we provided the number of solutions for grid graphs which possess more than one solution for the All-Ones problem.

[2] In Appendix A, we provide a computer algorithm which solves the All-Ones problem for $K^n$ graphs for any size $n$

```
n = k**2
print "Your graph is",k,"x",k,"grid graph,
       and has",n," Vertices"
b=[]    # For 1 vector
P=[]    # For 0 vector
X=[]    # For unknowns
M=[]    # For the adjacency matrix
# create zero n x n matrix
for i in range(n):
    M.append([0]*n)
# Create the vector 1.
for r in range(n):
    b.append([1])
for s in range(n):
    P.append([0])
for row in range(n):
    M[row][row]=1
    if (row+1)% k !=0:
        M[row][row+1]=1
    if row % k !=0:
        M[row][row-1]=1
    if row > k-1:
        M[row][row-k]=1
    if row < n-k:
        M[row][row+k]=1
print "The Adjacency matrix = ","\n",M
Forward_Elimination(M,b,n,k,P)
```

## 5.2.2 Pivoting Operation

Then the second function takes the adjacency matrix and performs the first step in the Gaussian elimination, the Forward elimination. We used the partial-pivoting operation to avoid zero division.The given matrices below, $A_3$ and $A_4$ are the augmented matrices in their reduced form for $G^n$ of sizes 3 and 4 respectively as produced by the function. The pivot elements are indicated by a box around it. And it is clear that for the $G^3$ graph the adjacency matrix has full rank and its kernel has an empty basis, consequently we have exactly one solution. On the other hand for $G^4$ graph the adjacency matrix is not invertible and the null space is of dimension 4, hence we have 16 solutions in this case.

$$A_3 = \begin{pmatrix} \boxed{1} & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & \boxed{1} & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & \boxed{1} & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \boxed{1} & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & \boxed{1} & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & \boxed{1} & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \boxed{1} & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \boxed{1} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \boxed{1} & 1 \end{pmatrix} \quad A_4 = \begin{pmatrix} \boxed{1} & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & \boxed{1} & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & \boxed{1} & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \boxed{1} & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & \boxed{1} & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & \boxed{1} & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \boxed{1} & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \boxed{1} & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \boxed{1} & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \boxed{1} & 1 & 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \boxed{1} & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \boxed{1} & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

```
(2) This function creates the augmented matrix in its reduced form, and
   gives the dimensions of the kernel.
def Forward_Elimination(M,b,n,k,P,det):
    d = 0
    M_Aug=concatenate((M,b),1)
    M_Aug=array(M_Aug)
    print "The augmented matrix M_Aug = ","\n","M_Aug"
    size = shape(M_Aug)
    for i in range(n):
        if M_Aug[i,i] != 1: ## Looking for a pivot
            for r in range(i+1,n):
                if M_Aug[r][i]==1:
                    M_Aug[r],M_Aug[i]=M_Aug[i],M_Aug[r]+0
                    break
        for j in range(i+1,n):
            if M_Aug[j][i]==1:
                for r in range(n+1):
                    M_Aug[j][r]=mod(M_Aug[j][r]+M_Aug[i][r],2)
    print "The augmented matrix in its reduced form= ","\n",M_Aug
    M_Aug=array(M_Aug)
    for u in range(n):
        if sum(M_Aug[u])==0:
            d = d+1
    print "The dimensions of the Kernel d_n= ",d
    Backward_substitution(M_Aug,n,k,P,d)
```

### 5.2.3 Null Space Engine

Next we proceed with the last step in the Gaussian elimination technique; that is, the backward substitution. Here we distinguish two different cases. If the parity dimension $d(K_{G,\sigma^+})$ is zero, we solve directly for the unique predecessor of the pattern $\mathbf{1}$. Otherwise we calculate the number of solutions depending on the nullity of the adjacency matrix which is to be calculated in the Null Space() function, and then proceed by producing the solutions.

```
(3) This part performs the back substitution step in Gaussian elimination
    and provides all the solutions of the system.
def Backward_substitution(M_Aug,n,k,P,d):
    if d ==0:  ## Reversible case
        print "The solution is Unique"
        for s in range(n):
            add=0
            for r in range(n-s,n):
                add=add+ (M_Aug[n-1-s][r]*P[r][0])
            x = int(mod((M_Aug[n-1-s][n]-add),2))
            P[n-1-s][0]=x
        P=reshape(P,(k,k))
        print "Gauss_Pivot: for" ,k,"X",k,"grid graph a solution =",'\n',P
        Odd_Parity_Cover(P,k)
    else: ## Irreversible case
        Q = Null_Space(d)
        for w in Q:
            P=([0]*(n-d))+w
            P=reshape(P,(n,1))
            for s1 in range(n-d):
                s=s1+d
                add=0
                for r in range(n-s,n):
                    add=add+ (M_Aug[n-1-s][r]*P[r][0])
```

```
                x = int(mod((M_Aug[n-1-s][n]-add),2))
                P[n-1-s][0]=x
            P=reshape(P,(k,k))
            print "Gauss_Pivot: for" ,k,"X",k,"grid graph a solution =",'\n',P
            Odd_Parity_Cover(P,k)
        print "You have got",2**d,"Solutions"
def Null_Space(d):
    Null=[]
    if d ==0:
        return []
    elif d == 1:
        return [[0],[1]]
    else:
        temp = Null_Space(d-1)
        for c in temp:
            Null.append([0]+c)
            Null.append([1]+c)
        return Null
```

### 5.2.4 Pattern Analyser

Checking whether a given pattern is a predecessor of the pattern **1** is the object of the last function of the algorithm. This function tests each solution produced in the previous step and tells whether it is a solution to the All-Ones problem of given size $n$.

```
(4) This function checks if a given pattern is an odd-parity cover.
def Odd_Parity_Cover(P,k):
     sol=1
     even=[]
     for row in range(k):
        for col in range(k):
            i = 0
            if P[row][col]==1:
                i = i+1
            if col != (k-1):        # Stands for or the last column
                i =i+ P[row][col+1]
            if col != 0:            # Stands for the first column
                i = i+ P[row][col-1]
            if row > 0:             # Stands for first row
                i = i+ P[row-1][col]
            if row < k-1:           # Stands for the last row
                i= i+ P[row+1][col]
            if int(i) % 2 == 0:
                sol=0
                ver=(col+1)+(k*row)
                even.append(ver)
     if sol ==0:
         even=sort(even)
         print "oops. The vector X is not an odd Parity cover,
                Vertices (",even,") have even number of neighbours from X"
     elif sol ==1 :
         print "X is an odd parity cover.
Choice()
```

Figure 5.1 represents the solution to the All-Ones problem for the $G^{26}$ graph. $G^{26}$ graph has a unique solution to the All-Ones problem and is therefore not listed in Table 4.4.3.

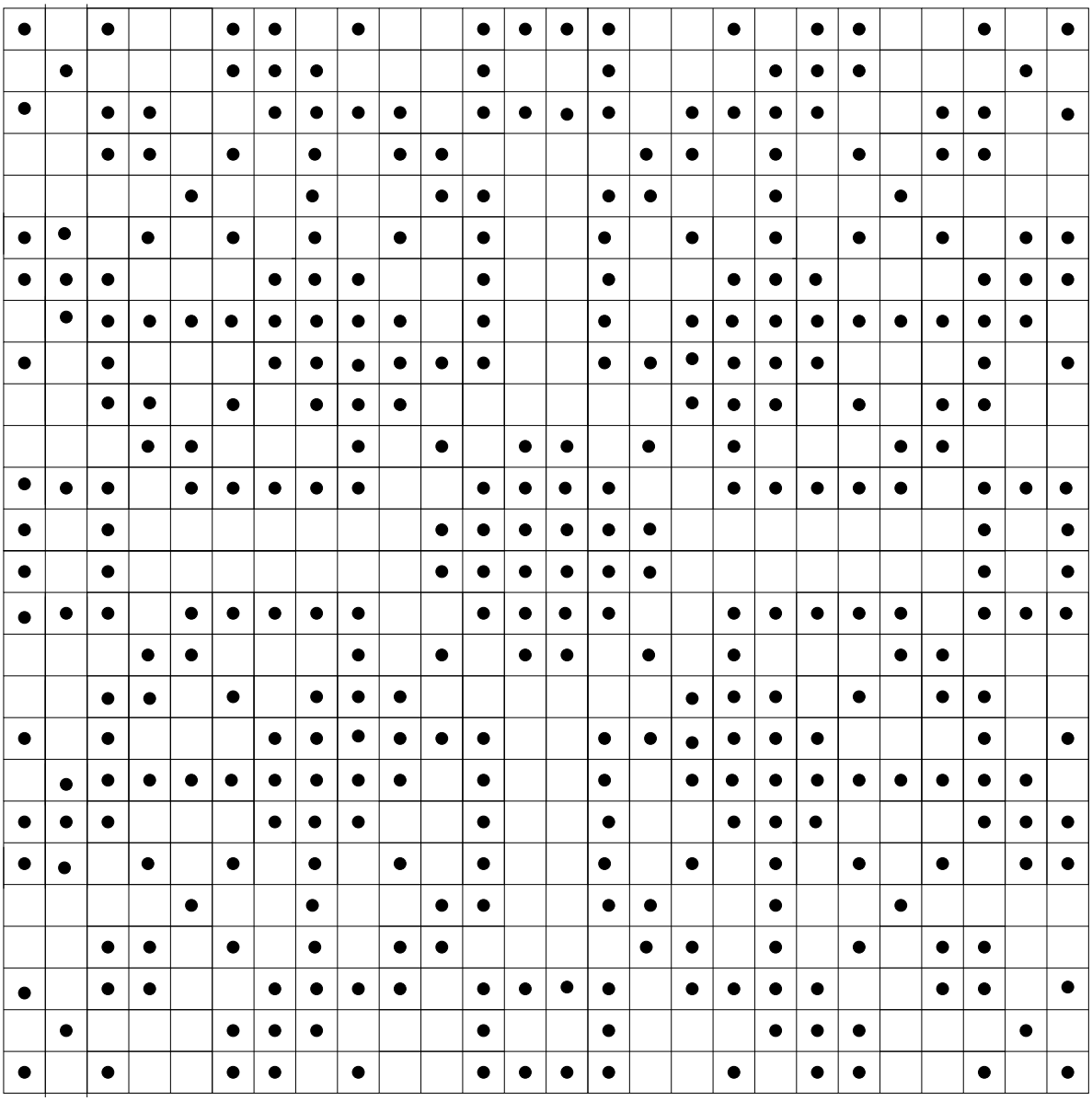


Figure 5.1: The Odd Parity Cover for $G^{26}$ graph.

It is remarkable that the solution for the $G^{26}$ graph contains in its middle part the solution for the $G^6$ graph, also the solution for $G^4$ graph appears in the middle of each side, and the solution for the $G^3$ graph appears at each corner of Figure 5.1.

# 6. Conclusion

We provided a short introduction to the wide area of graph theory, and showed how graphs can serve as mathematical models. We also demonstrated how linear algebra can be used as a supplementary tool in the treatment of graph-theoretical problems. Indeed, we were able to provide a complete solution to the All-Ones problem on complete graphs in its general form, and we could prove that the All-Ones problem on general graphs is always solvable under the $\sigma^+$-rule.

There are many interesting problems related to the All-Ones problem which we considered in this work that are still open; for instance, the vertex-to-vertex, the vertex-to-edge, the edge-to-vertex, and the edge-to-edge problems that introduced in [LZ05] can be considered for different switching rules. We can also define these problems on coloured graphs; for instance, for the case of the vertex-to-vertex problem, we associate each vertex in a graph $G$ with a button and one of three colours red, blue, or green. And we define a switching rule as follows: switching a button that is associated with a vertex $v$ in $G$ changes the colour of all the vertices in the neighbourhood of $v$ in $G$ from red to blue, from blue to green, or from green to red. Now the problem can be stated as follows: If all the vertices are initially red is it possible to press a sequence of buttons such that all the vertices are green at the end? More generally, we can consider two different configurations of colours $X$ and $Y$ in look for a sequence of buttons.

# Appendix A. Algorithm for Complete Graphs and Illustrations

## A.1 Algorithm and Complete Graphs

In this section we provide an algorithm written in Python that solves the All-Ones problem on complete graphs in its general form.

### A.1.1 Parity Identifier

The first task is to provide a procedure for solving the All-Ones problem on $K^n$ graphs in its general form. The function "Parity Identifier()" applies the procedures provided in chapter 3 and provides the solution set which converts a given situation $X$ to another situation $Y$ if any exists.

```
(1)This Function solves the All-Ones problem on complete graphs in its general form.
from scipy import *
def Parity_Identifier():
    Sol1=[]
    Sol2=[]
    test=[0,1]
    Counter1=[]
    Counter2=[]
    print "This program is made to solve the All-Ones problem on
           complete graphs in its general form",'\n'
    X = input("please input a situation X, between []: ")
    Y = input("please input a situation Y, between []: ")
    n = len(X)
    if len(X) != len(Y):
        print "Sorry X and Y should have the same length."
        Complete_Graph()
    if X == Y:
        print "The two patterns are the same"
        return
    for i in range(n):
        if X[i] not in test:
            print "The entries of X must be either 0 or 1."
            Complete_Graph()
        if Y[i] not in test:
            print "The entries of Y must be either 0 or 1."
            Complete_Graph()
    if n % 2 == 0:
        if (sum(X)%2) != (sum(Y)%2):
            for i in range(len(X)):
                if X[i]==Y[i]:
                    Sol1.append(i+1)
                    Counter1.append(1)
                else:
                    Counter1.append(0)
            print "The solution set is the buttons: ",Sol1
        if (sum(X)%2 == sum(Y)%2):
            for j in range(len(X)):
                if X[j]!=Y[j]:
                    Sol1.append(j+1)
                    Counter1.append(1)
```

```
            else:
                Counter1.append(0)
        print "The solution set is the buttons: ",Sol1
if n % 2 != 0:
    if (sum(X)%2) != (sum(Y)%2):
        print "It is imppossible to convert the situation X to Y!"
        return
    if (sum(X)%2 == sum(Y)%2):
        print 'you have two solutions','\n'
        for j in range(len(X)):
            if X[j]!=Y[j]:
                Sol1.append(j+1)
                Counter1.append(1)
            else:
                Counter1.append(0)
        print "The first solution set is: ",Sol1
        for k in range(len(X)):
            if X[k]==Y[k]:
                Sol2.append(k+1)
                Counter2.append(1)
            else:
                Counter2.append(0)
        print "The second solution set is: ",Sol2
Test_Engine(X,Y,n,Sol1,Sol2,Counter1,Counter2)
```

## A.1.2 Neighbourhood Matrix and Test Engine

In this part, we solve the linear system which represents the All-Ones problem on complete graphs as given in section 3.1 to justify the procedure used in the "Parity Identifier()" function to solve the problem.

$N_8$ and $N_{15}$ represent the neighbourhood matrix of the $K^8$ and the $K^{15}$ graphs respectively.

$$N_8 = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \end{pmatrix} \quad N_{15} = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \end{pmatrix}$$

The function Test Engine() checks whether a given set of buttons converts a situation $X$ to another situation $Y$.

```
(2)This function checks whether a given set of buttons converts a situation X to a situation Y.

def Test_Engine(X,Y,n,Sol1,Sol2,Counter1,Counter2):
    M=[]
    for i in range(n):
        M.append([1]*n)
    for d in range(n):
        M[d][d]=0
    M=reshape(M,(n,n))
    print M,"\n"
    X=reshape(array(X),(n,1))
    Y=reshape(array(Y),(n,1))
    L1=reshape(array(Counter1),(n,1))
    Y1=mod((X+(matrixmultiply(M,L1))),2)
    if Y==Y1:
        if n % 2 ==1:
            print "The first set of buttons provided will definitely convert X to Y."
        else:
            print "The set of buttons provided will definitely convert X to Y."
    else:
        print "Please run the program again, There is unknown error occurred."
    if n % 2 ==1:
        L2 = reshape(array(Counter2),(n,1))
        Y2=mod((X+(matrixmultiply(M,L1))),2)
        if Y==Y2:
            print "The second set of buttons will also convert X to Y"
        else:
            print "Please run the program again, There is unknown error occurred."
```

## A.2 Illustrative Examples on Chapter 3

This section provides some illustrative examples to the All-Ones problem on Complete graph in its general form.

**Example A.2.1** *For a given situation $X$ in an even $K^n$ graph, to change the state of only one light we need to press all the buttons except the button that is associated with the light we do want to change. So to convert $X$ = (1 0 1 0 1 1) to $X'$ = (0 0 1 0 1 1) we need to do the following.*

$$X = (101011) \xmapsto{P_2} (000100) \xmapsto{P_3} (110011) \xmapsto{P_4} (001000) \xmapsto{P_5} (110101) \xmapsto{P_6} (001011) = X' \tag{A.1}$$

**Example A.2.2** *In the case of the complete graph $K^n$,let the situation $X = (101101)$ be given. Is it possible to convert $X$ into another situation $Y = (010111)$? The situations $X$ and $Y$ both have even parity, and according to Proposition 3.2.1 it is possible to convert $X$ to $Y$, and the solution set is $\{P_1, P_2, P_3, P_5\}$.*

$$X = (101101) \xmapsto{P_1} (110010) \xmapsto{P_2} (011101) \xmapsto{P_3} (101010) \xmapsto{P_5} (010111) = Y \tag{A.2}$$

**Example A.2.3** *Let $X = (11100101)$ be a situation in the $K^8$ graph. Is it possible to convert $X$ to another situation $Y = (10110011)$. The situations $X$ and $Y$ both have odd parity, so as*

*shown in Proposition 3.2.1 it is possible to convert the situation $X$ to $Y$ and the solution is the set of buttons whose associated lights have different states in $X$ and $Y$ i.e $\{P_2, P_4, P_6, P_7\}$ .*

$$X = (11100101) \overset{P_2}{\longmapsto} (01011010) \overset{P_4}{\longmapsto} (10110101) \overset{P_6}{\longmapsto} (01001110) \overset{P_7}{\longmapsto} (10110011) = Y \tag{A.3}$$

**Example A.2.4** *For a situation $X = (111101111)$ in the $K^9$ graph is it possible to convert $X$ to another situation $Y = (101000101)$. According to Proposition 3.3.2 this is possible and we have two solutions. The first solution is the set of buttons whose associated lights have different states in $X$ and $Y$, i.e. $\{P_2, P_4, P_6, P_8\}$.*

$$X = (111101111) \overset{P_2}{\longmapsto} (010010000) \overset{P_4}{\longmapsto} (101001111) \overset{P_6}{\longmapsto} (010111000) \overset{P_8}{\longmapsto} (101000101) = Y \tag{A.4}$$

*The set of lights which have the same states in $X$ and $Y$ represents the second solution i.e. $\{P_1, P_3, P_5, P_7, P_9\}$*

$$\begin{aligned} X \quad = \quad & (111101111) \overset{P_1}{\longmapsto} (100010000) \overset{P_3}{\longmapsto} (010101111) \overset{P_5}{\longmapsto} (101000000) \overset{P_7}{\longmapsto} (010111011) \\ & \overset{P_9}{\longmapsto} (101000101) = Y. \end{aligned} \tag{A.5}$$

# Bibliography


[ACS98] Ashok T. Amin, Lane H. Clark, and Peter J. Slater. Parity dimension for graphs. *Discrete Math.*, 187(1-3):1–17, 1998.

[Bea06] A. F. Beardon. *From Problem Solving to Research*. 2006. Unpublished manuscript.

[Ber62] Claude Berge. *The Theory of Graphs and Its Applications*. Mathuen and John Wiley, 1962.

[Big93] Norman Biggs. *Algebraic Graph Theory*. Cambridge University Press, 2nd edition, 1993.

[Bol98] Béla Bollobás. *Modern graph theory*, volume 184. Springer-Verlag, Berlin, 1998.

[CLWZ04] William Y. C. Chen, Xueliang Li, Chao Wang, and Xiaoyan Zhang. The minimum all-ones problem for trees. *SIAM Journal on Computing*, 33(2):379–392, 2004.

[Die05] Reinhard Diestel. *Graph Theory*, volume 137. Springer-Verlag Heidelberg, Cambridge, 3rd edition, 2005.

[Gar85] C. Gary. *Introductory Graph Theory*. Dover Publications, Inc, 1985.

[GR01] C. Godsil and G. Royle. *Algebraic Graph Theory*, volume 207. Springer, 2001.

[LZ05] Xueliang Li and Xiaoyan Zhang. New versions of the all-ones problem. *Available from "arXiv:math/0512653"*, 2005.

[Ore63] Oystein Ore. *Graphs and their Uses*. Random House, New York, 1963.

[OS06] Peter J. Olver and Chehrzad Shakiban. *Applied Linear Algebra*. Prentice Hall, Upper Saddle River, NJ, 2006.

[Slo02] N.J.A. Sloane. The on-line encyclopedia of integer sequences. *Published electronically at: http://www.research.att.com/∼njas/sequences/*, 2002.

[Sut89] K. Sutner. Linear cellular automata and the garden-of-eden. *Math. Intelligencer*, 11:49–53, 1989.

[Tem81] H. N. Temperley. *Graph Theory and Applications*. Ellis Horwood, Chichester, 1981.

[Tut84] W. T. Tutte. *Graph Theory*. Cambridge University Press, 1984.

[Wal84] Hansjoachim Walther. *Ten Applications of Graph Theory*. D. Reidel Publishing Company, Dodrecht, 1984.