國立暨南國際大學資訊工程學系

碩士論文

指導教授：李家同教授

符號邏輯於基因調控系統之應用

Applications of Symbolic Logic to Gene Regulation Systems

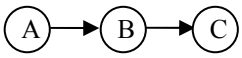
研究生：林莊傑

中華民國九十三年六月

中文摘要 (Chinese Abstract)

系統生物學是一門在分子生物學上相當新的分支。在這個領域裡，基因乃被獨立且分別地研究。相反地，我們對於這些基因是如何影響彼此感到興趣。因為基因交互作用更為複雜，而且找出這些基因如何影響彼此是一點也不容易的，所以這領域目前仍在未開發的階段。

在本篇論文中，我們應用符號邏輯來解決這個問題。也就是說，我們使用布林邏輯式子來描述基因如何來啟動或是抑制其他基因。通常來說，根據實驗結果，我們可以使用一個布林基因調控網路來描述基因如何影響彼此。假如說一個基因 A 啟動或是抑制了一個基因 B，我們就從 A 到 B 畫一條有方向性的線。例如，藉由以下的有向邊，我們可以很容易的看到，假如從 B 到 C 有一條有向邊的話，基因 A 就會影響基因 C。



在本篇論文中，我們將會說明在基因之間有所謂的隱含交互作用。也就是說，雖然從基因 X 並沒有任何路徑可到基因 Y，它們之間的交互作用仍然存在。舉例來說，基因 X 可能抑制基因 Y 。我們將會指出，隱藏交互作用的尋找問題，實際上在符號邏輯裡，就是邏輯推論的尋找問題。藉由使用解析原理法，我們能找出許多隱含的交互作用，而這些隱含的交互作用對於不熟捻的讀者而言是一點都不明顯的。

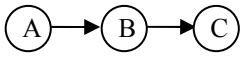
我們相信我們的方法，能被生物學家用來從他們的實驗結果中推導出更多資訊，並且帶領他們到新的研究主題。例如，他們可以做更多的實驗，來看看這些隱含交互作用是否合乎新的實驗結果。若非如此，那麼原先的實驗可能有錯。

關鍵字：基因，DNA，基因調控網路，布林基因調控網路，基因擾動，基因表現，狀態決定問題，隱含交互作用的尋找問題，符號邏輯，解析原理法

Abstract

System biology is a relatively new branch of molecular biology. In this field, genes are studied individually and separately. Instead, we are interested in how the genes interact with one another. Since gene interactions are rather complicated and it is not easy at all to find out how the genes influence each other, this field is still in its primitive stage.

In this thesis, we apply symbolic logic to the problem. That is, we use Boolean logic formulas to describe how genes activate or inhibit other genes. Usually, based upon experimental results, we can use a Boolean gene regulatory network to describe how genes influence one another. If a gene *A* activates, or inhibits, a gene *B*, we draw a directed line from node *A* to node *B*. By following the directed edges, we can easily see, for instance, that gene *A* influences gene *C* if there is a directed edge from gene *B* to gene *C*.



In this thesis, we shall show that there are so-called implicit interactions among genes. That is, although there is no path leading from gene *X* to gene *Y*, there still is an interaction between them. For instance, gene *X* may inhibit gene *Y*. We shall point out that the implicit interactions finding problem is actually the logical consequence finding problem in symbolic logic. By using the resolution principle method, we can find out many implicit interactions which are not obvious to naïve readers at all.

We believe that our approach can be used by biologists to deduce more information from their research results and lead them to new research topics. For instance, they may perform more experiments to see if these implicit interactions are compatible to the new experimental results. If not, there may be something wrong with the first round of experiments.

Keywords. gene, DNA, gene regulatory network, Boolean gene regulatory network, gene perturbation, gene expression, state determination problem, implicit interaction finding problem, symbolic logic, the resolution principle method

Acknowledgement

First, I would like to thank my advisor, Professor R. C. T. Lee, for his patient teaching, guidance and invaluable instructions that made me have a great progress in English, humane sympathy and research. For making me have common sense and feelings at literature and classical music, he also suggested me to spend time in listening to classical music, reading mystery novels so that I can appreciate beautiful things in the world which I didn't notice before. All the things he has done to me enrich my life. All the words he has said remain unshakable and become even firmer as time goes by.

Second, I am grateful to my great parents for their support and encouragement. Their support helped me concentrate on studying and made me feel free from care. Furthermore, their encouragement gave me power to move on my way to the future without shrinking back.

Third, I want to show my appreciation to my classmates, B. W. Xiao, J. P. Lu, F. L. Lin, M. C. Chen, J. R. Hsu, and B. H. Wu, for their help from beginning to end. My great progress in programming should be attributed to their assistance and advice. Besides, I also want to thank them for their giving me a lot of opportunities to gain much more knowledge.

Fourth, I deeply appreciate Prof. Huang's assistance and instruction. I gained much knowledge about biology and mathematics from him so that I can successfully accomplished this thesis. Without his help, many problems I encountered can't be solved so easily.

Finally, I would like to show my appreciation to this school, National Chi Nan University. NCNU is an excellent school with outstanding professors, diligent students and beautiful environment. It's quite fortunate that I can spend two years in doing research at this brilliant university.

Chuang Chieh Lin 林莊傑
Computer Science and Information Engineering
National Chi Nan University, R. O. C.
May, 2004

Contents

List of Figures

List of Tables

Chapter 1
Introduction

Genes are known as specific regions on a DNA sequence, and they carry information for manufacturing proteins. Since proteins keep our life, genes play a significant role in human beings' life and health. To study genes, we have to understand gene expressions, which are the processes that hereditary information of genes transforms into mRNA or proteins. We say that a gene is activated if its process of making mRNA or a protein is executed; otherwise, we say that a gene is inhibited. Gene expression of a gene *A* denotes whether *A* is activated or inhibited. It deserves to be mentioned that there exist interactions between genes, that is, a gene's product may bind to another gene on a specified region and repress or enhance it. Since a gene expression depends on the interactions between the genes, scientists are encouraged to analyze interactions between genes. In this thesis, we will emphasize on studying gene activations, inhibitions and interactions among given genes. Now, let us proceed to discuss our subject – *gene regulatory networks*.

A gene regulatory network is a network structure representing the interactions between genes. An illustration of a typical gene regulatory work in a hypothetical biochemical pathway is shown in Figure 1-1. In order to be familiar with Figure 1-1, we have to give some definitions first. A *transcription factor* T_X produced by a gene X is a protein which binds another gene Y. Without T_X, gene Y can't be activated, that is, it can't transform into mRNA or a protein. A *protein kinase* is the enzyme responsible for phosphorylation. Instead of repressing the expression of another gene, a protein kinase catalyzes the phosphorylation effect of a phosphate group and a protein produced from another gene. A *phosphatase* is the enzyme responsible for

the dephosphorylation, that is, removing phosphate groups from phosphorylated residue.

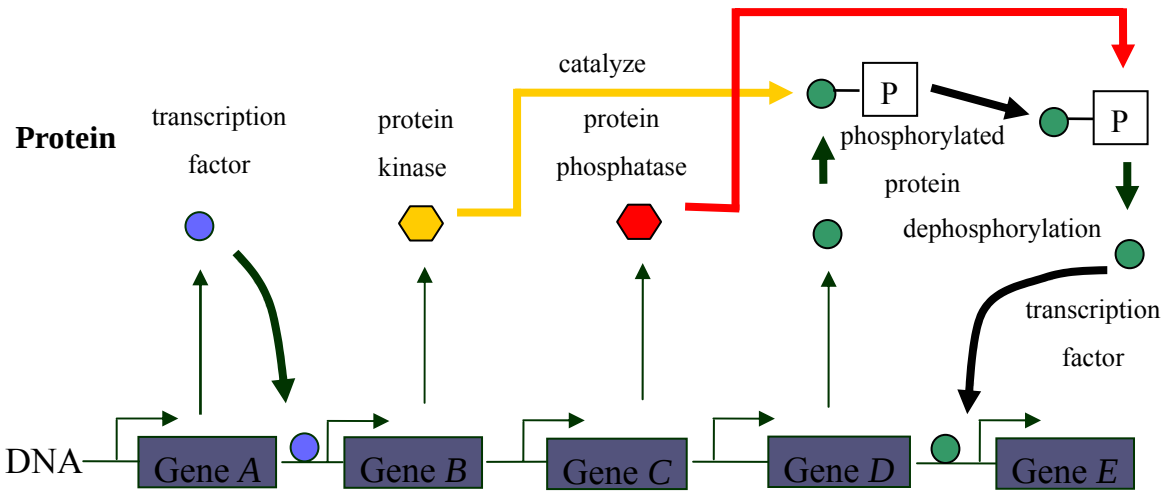


Figure 1-1 An illustration of gene regulatory networks

In Figure 1-1, the transcription factor produced by Gene *A* causes Gene *B* to be activated. Gene *B*'s product is a protein kinase which can catalyze a phosphorylation reaction. Gene *C*'s product is a protein phosphatase which can catalyze a dephosphorylation. Here P denotes a phosphate group. The product of Gene *D* is a protein originally. Due to the catalytic action caused by a protein kinase produced by Gene *B*, Gene *D*'s product is phosphorylated. And then, this phosphorylated protein is dephosphorylated because of the catalytic action caused by the phosphatase, which is Gene *C*'s product. After the dephosphorylation, a transcription factor is produced, and it activates Gene *E*. By this illustration in Figure 1-1, we can roughly understand the gene regulatory networks. Now, we shall narrate our motivation in the following section.

1.1 Motivations

Human genome sequencing was the most important target of Human Genome Project (HGP). However, after the human genome sequencing was completed, the postgenomic era and the age of functional genomics have arrived. One aspect of functional genomics is the understanding of how genes are expressed or regulated which is critically important to finding ways to fight diseases. It has been found by scientists that diseases are often related to how genes are expressed and regulated.

For instance, well-understood gene regulatory network will allow us to comprehend a specified cell's response to toxic substances. Therefore, if the gene regulatory network models are developed, manipulating gene expressions will be possible and have a great contribution in the treatments of diseases. Promise for drug discovery and delivery will be held through analyzing the gene regulatory network.

Some studies related to gene regulatory networks focus on a simple graph framework. Under this graph framework, Wagner [W2001, W2002] only discussed gene perturbations instead of clearly distinguishing gene activations from gene inhibitions. Through observing this graph framework, we find that every interaction can be symbolized and formulized by symbols and logical formulas. Therefore, we are encouraged to study symbolic logic and make use of the resolution principle method to find new information from the given perturbation data. The new information here may reveal the implicit interactions which we do not know from the original data. We will introduce symbolic logic and the resolution principle method in Chapter 2. Besides, Akutsu *et al*. [AKMM98, AMK99, LSH2002] studied a quite useful model - the Boolean gene regulatory network, which will be introduced in

Chapter 3. The Boolean gene regulatory network shows interactions between the given genes more completely. Under this model, we can study gene regulatory networks and apply the resolution principle method to find the implicit gene interactions in a Boolean gene regulatory network. In addition, we also want to know whether a gene is activated or inhibited in a Boolean gene regulatory network. Thus we are encouraged to study the state determination problem which will be also introduced in Chapter 3. This problem can be solved by the depth-first-search method [M89] with slight modifications which is implemented in a pseudo code in Appendix 3.1.

After briefly introducing the gene regulatory network and speaking of the motivations, next we shall summarize the previous work about the related research for gene regulatory networks.

1.2 Previous Work

In the field of system biology, some results of reconstructing gene regulatory networks have worked out. Lee *et al*. [LRR *et al*. 2002] discussed several types of genes in the transcriptional regulatory network, and they combined the data of binding sites and gene expression data to assemble a transcriptional regulatory network of the transcriptional regulators in the yeast named *eukaryote Saccharomyces cerevisiae*. This gene regulatory network describes potential pathways which yeast cell can use to regulate global genes. Davidson *et al*. [DRO *et al*. 2002] summarized a gene regulatory network that controls the specification of endoderm and mesoderm in the sea urchin embryo. It is quite significant that Davidson's laboratory [D] has developed a useful tool to simulate the reactions of a gene regulatory network. Gene

expressions in this gene regulatory network will change as time goes on. Kightley *et al*. [KCE2004] made use of Davidson *et al*.'s result to generate an Endomesoderm gene regulatory network and simplified it by removing the redundant edges of affection. Babu and Teichmann [BT2003] also studied the prokaryote *Escherichia coli* which has the most detailed and available information. They found that a gene would rarely be regulated by more than six other genes. Analysis of gene regulatory networks was also applied to identify drug-affected genes by Savoie *et al*., [SAW *et al*. 2003]. They successfully identified an important affected target gene of a certain yeast.

For more theoretical research, some models and methods established to analyze gene regulatory network have been proposed. Wagner [W2001, W2002] presented an algorithm to reconstruct a gene regulatory network from perturbation data under a simple graph framework, and simplified a gene regulatory network to the most parsimonious gene regulatory network. The time complexity of this algorithm is $O(n^2)$ where n denotes the number of given genes. Yet every edge in the gene regulatory network does not show which kind of affection was caused. In regard to dealing with steady-state expression data, Kyoda *et al*. [KMOK2000] proposed an inference method determining a gene regulatory structure consistent with the data. Bay *et al*. [BSPL2003] discussed linear causal models to modify gene regulatory networks and improve them to be consistent with biological knowledge. Genetic algorithms have been also applied to reconstruct gene regulatory networks [AI2001, CLA2003]. Several models for genetic algorithms have been proposed.

Concerning the logical inference model, Woolf and Wang [WW2000] developed an algorithm which uses fuzzy logic to transform gene expressions into qualitative descriptors. In addition to the approach of applying fuzzy logic, there is another quite important model – the Boolean network model. For the Boolean network

model, there are some significant topics for analyzing gene regulatory networks, such as solving the consistency problem, solving the identification problem, the symbolic logic model and the Boolean circuit model. Karp *et al*. [KSY99] represented the biological pathways as Boolean circuits and gave algorithms for choosing a set of experiments of gene expressions to the mating pathways. Besides, Lähdesmäki *et al*., Aburatani *et al*. and Akutsu *et al*.'s have also made progress in this topic. [ASN *et al*. 2003, AKMM98, AMK99, LSH2002]. Especially, Akutsu *et al*. [AKMM98, AMK99, LSH2002] demonstrated the consistency problem and the identification problem under the Boolean network model. Simple algorithms have been proposed to solve the above two problems with some limitations.

Next, we shall introduce symbolic logic in the following chapter. Symbolic logic will be very useful to analyze the gene regulatory network. We shall use symbolic logic through this thesis.

Chapter 2

Symbolic Logic and the Resolution Principle Method

In symbolic logic, sentences in natural languages are expressed by logical formulas. For example, let us see the sentences as follows:

F_1: "If it rains, the ground will be wet.

F_2: "It the ground is wet, we won't go outside."

F_3: "It rains."

If we use symbols to represent it, what formula will we get? Let P, Q and R represent "It rains," "The ground is wet" and "we will go outside" respectively. Let "$\neg$", "$\wedge$" and "$\rightarrow$"represent "not", "and" and "imply" respectively. Then, F_1, F_2 and F_3 can be represented as follows:

F_1: $P \rightarrow Q$

F_2: $Q \rightarrow \neg R$

F_3: P

Thus, F_1, F_2 and F_3 have been transformed into logical formulas. By logic deduction, we can obtain F_4 as follows:

F_4: $\neg R$

We say that F_4 logically follows from F_1, F_2 and F_3. In other words, F_4 can be regarded as new information.

Given a set of Boolean logical formulas, how can we find new information? This will be discussed in the next section.

2.1 The Resolution Principle Method

To introduce the resolution principle method, some notations and definitions need to be introduced first.

For symbolic logic, the symbols, such as *A*, *B* and *C* are called *atoms*. Formulas are defined recursively as follows:

(1) An atom is a formula.

(2) If *G* is a formula, then $\neg G$ is also a formula.

(3) If G and H are formulas, then $G \wedge H$, $G \vee H$, $G \rightarrow H$ and $G \leftrightarrow H$ are formulas, where $\vee$, $\wedge$, $\rightarrow$ and $\leftrightarrow$ dente "or", "and", "imply" and "if and only if" respectively.

(4) All formulas are generated by applying the above three rules.

We define that an atom or the negation of an atom is a *literal*. For example, *A*. $\neg B$, *C* are all literals. Suppose we have formulas F_1, F_2, …, F_n, then $F_1 \vee F_2 \vee \cdots \vee F_n$ is called the *disjunction* of F_1, F_2, …, F_n and $F_1 \wedge F_2 \wedge \cdots \wedge F_n$ is called the *conjunction* of F_1, F_2, …, F_n. A disjunction of literals is called a *clause*. For example, $A \vee B$, $\neg X \vee Y \vee Z$ and $\neg E \vee \neg F$ are all clauses. A formula *F* is said to be in a *conjunctive normal form* if and only if *F* has the form $F_1 \wedge F_2 \wedge \cdots \wedge F_n$, *n*

≥ 1, where each F_i is a clause, $i = 1, 2, \cdots, n$. For example, $(A \vee \neg B \vee \neg C) \wedge (\neg P \vee Q \vee R)$ is a formula in a conjunctive normal form, and $A \wedge (\neg B \vee \neg C)$ is also a formula in a conjunctive normal form. Given a formula *G*, let A_1, A_2, …, A_n be atoms occurring in the formula *G*. Then an *interpretation* of *G* is an assignment of truth values to A_1, A_2, …, A_n in which every A_i, $1 \le i \le n$, is assigned either **T** or **F**, but not both. A formula is said to be *valid* if and only if it is true under all its interpretations while a formula is said to be *inconsistent* if and only if it is false under all its interpretations.

In this thesis, "*A*" stands for "gene *A* is activated" while "$\neg A$" stands for "gene *A* is not activated", that is, "gene *A* is inhibited". For "$A \to B$", "$\neg A \to B$", "$A \to \neg B$" and "$\neg A \to \neg B$", we have the following explanations.

"$A \to B$" means "If *A* is activated, *B* will be activated."

"$\neg A \to B$" means "If *A* is inhibited, *B* will be activated."

"$A \to \neg B$" means "If *A* is activated, *B* will be inhibited."

"$\neg A \to \neg B$" means "If *A* is inhibited, *B* will be inhibited."

Note that $A \to B$ is equivalent to $\neg A \vee B$. Similarly, $\neg A \to B$ is equivalent to $A \vee B$, $A \to \neg B$ is equivalent to $\neg A \vee \neg B$ and $\neg A \to \neg B$ is equivalent to $A \vee \neg B$.

Next, we have to define what a logical consequence is in symbolic logic. Given formulas F_1, F_2, …, F_n and a formula *G*, *G* is said to be a *logical consequence* of F_1, F_2, …, F_n if and only if whenever $F_1 \wedge F_2 \wedge \cdots \wedge F_n$ is true then *G* is also true. That is, *G* is a *logical consequence* of F_1, F_2, …, F_n if and only if the formula $(F_1 \wedge F_2 \wedge \cdots \wedge F_n \to G)$ is valid.

The resolution principle method is a method for deducing logical consequences

from a given set of clauses. We define the resolution principle method as follows.

For any two clauses C_1 and C_2, if there is a literal L_1 in C_1 that is complementary to a literal L_2 in C_2, then delete L_1 and L_2 from C_1 and C_2 respectively, and construct the disjunction of the remaining clauses. The constructed clause is a logical consequence of C_1 and C_2. [CL73]

Let us see an example. Consider the following clauses.

$$A \vee \neg B \tag{2.1-1}$$

$$A \vee \neg C \tag{2.1-2}$$

$$B \vee \neg C \tag{2.1-3}$$

$$D \vee \neg B \tag{2.1-4}$$

$$D \vee \neg E \tag{2.1-5}$$

By applying the resolution principle method, we can derive a logical consequence from clauses (2.1-1) and (2.1-3). This logical consequence is shown in clause (2.1-6) as follows.

$$A \vee \neg C \tag{2.1-6}$$

Formula (2.1-6) is obtained by eliminating $\neg B$ in formula (2.1-1) and B in (2.1-3).

By applying the resolution principle method, we can derive a logical consequence from clauses (2.1-3) and (2.1-4). This logical consequence is shown in formula (2.1-7) as follows.

$$D \vee \neg C \tag{2.1-7}$$

Clause (2.1-7) is obtained by eliminating B in (2.1-3) and $\neg B$ in (2.1-4). This clause is equivalent to $C \rightarrow D$ or $\neg D \rightarrow \neg C$.

Let us see another example. Consider the following set of clauses:

$$\neg A \tag{2.1-8}$$

$$B \tag{2.1-9}$$

$$\neg A \vee \neg B \tag{2.1-10}$$

$$C \tag{2.1-11}$$

$$\neg B \tag{2.1-12}$$

$$C \tag{2.1-13}$$

If we apply the resolution principle method, we can derive a logical consequence '□', which is an empty clause, from formula (2.1-9) and (2.1-12) by eliminating B and $\neg B$. Note that '□' denotes a clause which is inconsistent. Therefore, once a '□' appears after the resolution principle method is performed, we can say that there must be something wrong with the original data, because only a false statement is allowed to imply another false statement.

After introducing the resolution principle method, let us continue to the following section which introduces regulators and targets of a gene and states the experimental results of applying the resolution principle method to the data of genes, their regulators and targets.

2.2 Analysis of Regulators and Targets

In this section, we shall discuss a rather simple regulator and target model. At the

beginning, let us give some definitions first. The genes which directly activate or inhibit a gene X are called the *regulators* or *input* of X while the genes which a gene X directly activates or inhibits are called the *targets* or *output* of X. In this section, we consider a problem as follows. Suppose that we are only given a set of genes and the regulators and targets of each gene, how can we find some information new by symbolic logic from the given data? In order to solve this problem, we have to know the relations between a gene and its regulators and those between it and its targets. Let us introduce the possible relations as follows.

Given a gene X, there are six types of the relations between X, regulators of X and targets of X. These four types are shown in Figure 2.2-1, 2.2-2, 2.2-3, 2.2-4 2.2-5 and 2.2-6. Let us begin discussing the four types.

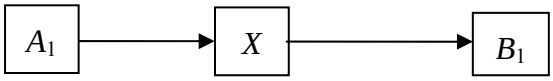


Figure 2.2-1 Type 1: Single input and single output

Type 1 (Corresponding to Figure 2.2-1): In Figure 2.2-1, we obtain that A_1 affects X and X affects B_1. We can formulize this event by

$$(A_1 \to X) \wedge (\neg A_1 \to \neg X) \wedge (X \to B_1) \wedge (\neg X \to \neg B_1) \qquad (2.2\text{-}1)$$

Thus, we have

$$\neg A_1 \vee X \qquad (2.2\text{-}2)$$

$$A_1 \vee \neg X \qquad (2.2\text{-}3)$$

$$\neg X \vee B_1 \tag{2.2-4}$$

$$X \vee \neg B_1 \tag{2.2-5}$$

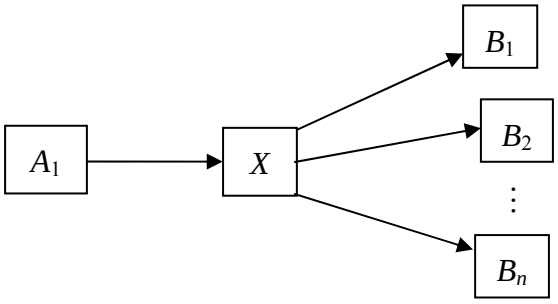


Figure 2.2-2 Type 2: Single input and multiple output

Type 2 (Corresponding to Figure 2.2-2): In Figure 2.2-2, we obtain that A_1 affects X and X affects B_1, B_2, …, and B_n. We can formulize this event by

$$(A_1 \to X) \wedge (\neg A_1 \to \neg X) \wedge (X \to B_1) \wedge (\neg X \to \neg B_1) \wedge (X \to B_2) \wedge (\neg X \to \neg B_2) \wedge \cdots \wedge (X \to B_n) \wedge (\neg X \to \neg B_n). \tag{2.2-6}$$

Thus, we have

$$(A_1 \to X) \tag{2.2-7}$$

$$(\neg A_1 \to \neg X) \tag{2.2-8}$$

$$(X \to B_1) \tag{2.2-9}$$

$$(\neg X \to \neg B_1) \tag{2.2-10}$$

$$(X \to B_2) \tag{2.2-11}$$

$$(\neg X \to \neg B_2) \tag{2.2-12}$$

$$\vdots$$

$$(\neg X \to \neg B_n) \tag{2.2-13}$$

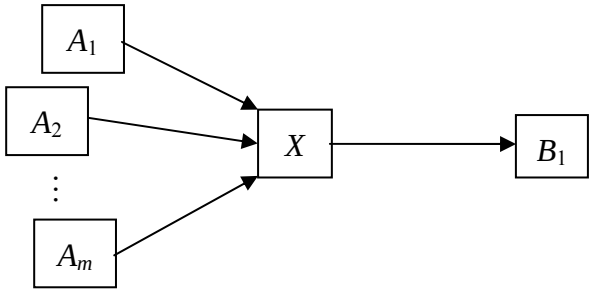


Figure 2.2-3 Type 3: Multiple input and single output

Type 3 (Corresponding to Figure 2.2-3): In Figure 2.2-3, we obtain that A_1, A_2, …, $A_{m\text{-}1}$ and A_m jointly affect X and X affects B_1. We can formulize this event by

$$((A_1 \wedge A_2 \wedge \cdots \wedge A_m) \rightarrow X) \wedge (\neg A_1 \rightarrow \neg X) \wedge (\neg A_2 \rightarrow \neg X) \wedge \cdots \wedge (\neg A_m \rightarrow \neg X) \wedge (X \rightarrow B_1) \wedge (\neg X \rightarrow \neg B_1). \quad (2.2\text{-}14)$$

Thus, we have

$$\neg A_1 \vee \neg A_2 \vee \cdots \vee \neg A_m \vee X \quad (2.2\text{-}15)$$

$$A_1 \vee \neg X \quad (2.2\text{-}16)$$

$$A_2 \vee \neg X \quad (2.2\text{-}17)$$

$$\vdots$$

$$A_m \vee \neg X \quad (2.2\text{-}18)$$

$$\neg X \vee B_1 \quad (2.2\text{-}19)$$

$$X \vee \neg B_1 \quad (2.2\text{-}20)$$

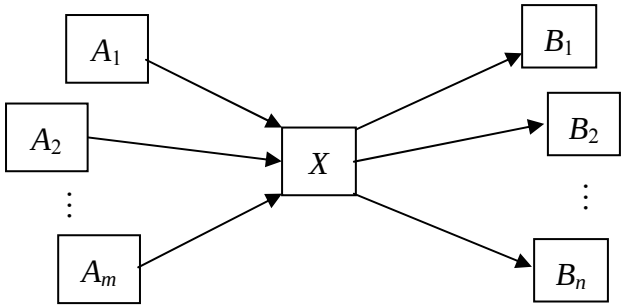


Figure 2.2-4 Type 4: Multiple input and multiple output

Type 4 (Corresponding to Figure 2.2-4): In Figure 2.2-4, we obtain that A_1, A_2, …, A_{m-1} and A_m jointly affect X and X affects B_1, B_2, …, and B_n. We can formulize this event by

$$((A_1 \wedge A_2 \wedge \cdots \wedge A_m) \rightarrow X) \wedge (\neg A_1 \rightarrow \neg X) \wedge (\neg A_2 \rightarrow \neg X) \wedge \cdots \wedge (\neg A_m \rightarrow \neg X)$$
$$\wedge (X \rightarrow B_1) \wedge (\neg X \rightarrow \neg B_1) \wedge \cdots \wedge (X \rightarrow B_n) \wedge (\neg X \rightarrow \neg B_n) \tag{2.2-21}$$

Thus, we have

$$\neg A_1 \vee \neg A_2 \vee \cdots \vee \neg A_m \vee X \tag{2.2-22}$$
$$A_1 \vee \neg X \tag{2.2-23}$$
$$A_2 \vee \neg X \tag{2.2-24}$$
$$\vdots$$
$$A_m \vee \neg X \tag{2.2-25}$$
$$\neg X \vee B_1 \tag{2.2-26}$$
$$X \vee \neg B_1 \tag{2.2-27}$$
$$\vdots$$
$$\neg X \vee B_n \tag{2.2-28}$$
$$X \vee \neg B_n \tag{2.2-29}$$

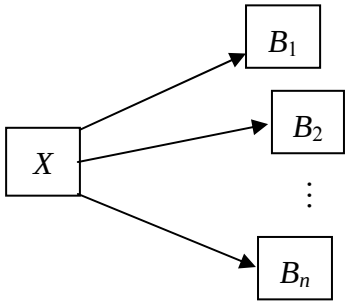


Figure 2.2-5 Type 5: $B_1,\ldots,B_n$ can't affect X

Type 5 (Corresponding to Figure 2.2-5): In Figure 2.2-5, we obtain that there is no regulator of X among these n genes B_1, B_2, …, and B_n while X affects B_1, B_2, …, and B_n. We can formulize this event by

$$(X \to B_1) \wedge (\neg X \to \neg B_1) \wedge \cdots \wedge (X \to B_n) \wedge (\neg X \to \neg B_n) \tag{2.2-30}$$

Thus, we have

$$\neg X \vee B_1 \tag{2.2-31}$$

$$X \vee \neg B_1 \tag{2.2-32}$$

$$\vdots$$

$$\neg X \vee B_n \tag{2.2-33}$$

$$X \vee \neg B_n \tag{2.2-34}$$

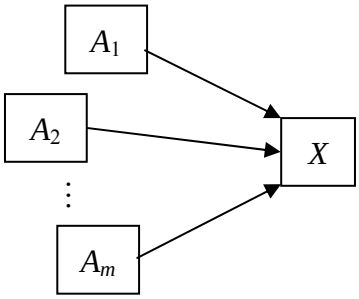


Figure 2.2-6 Type 6: $A_1,\ldots,A_m$ can't be affected by X

Type 6 (Corresponding to Figure 2.2-6): In Figure 2.2-6, we obtain that A_1, A_2, …, A_{m-1} and A_m jointly affect X but no gene among A_1, A_2, …, A_{m-1} can be affected by X. We can formulize this event by

$$((A_1 \wedge A_2 \wedge \cdots \wedge A_m) \rightarrow X) \wedge (\neg A_1 \rightarrow \neg X) \wedge (\neg A_2 \rightarrow \neg X) \wedge \cdots \wedge (\neg A_m \rightarrow \neg X) \quad (2.2\text{-}35)$$

Thus, we have

$$\neg A_1 \vee \neg A_2 \vee \cdots \vee \neg A_m \vee X \quad (2.2\text{-}36)$$
$$A_1 \vee \neg X \quad (2.2\text{-}37)$$
$$A_2 \vee \neg X \quad (2.2\text{-}38)$$
$$\vdots$$
$$A_m \vee \neg X \quad (2.2\text{-}39)$$

After transforming the original data to be in conjunctive normal forms, we can apply the resolution principle method to deduce new information. Let see an example to comprehend our method more clearly. In Figure 2.2-7, suppose that there are three genes and their corresponding regulators and targets.

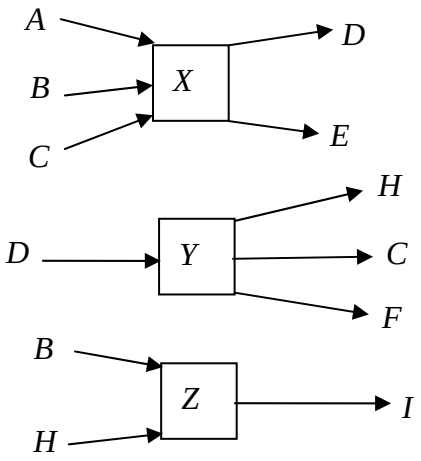


Figure 2.2-7 An illustration of genes with their targets and regulators

For gene X, we have

$$\neg A \vee \neg B \vee \neg C \vee X \quad (1)$$

$$A \vee \neg X \quad (2)$$

$$B \vee \neg X \quad (3)$$

$$C \vee \neg X \quad (4)$$

$$\neg X \vee D \quad (5)$$

$$X \vee \neg D \quad (6)$$

$$\neg X \vee E \quad (7)$$

$$X \vee \neg E \quad (8)$$

For gene Y, we have

$$D \vee \neg Y \quad (9)$$

$$\neg D \vee Y \quad (10)$$

$$Y \vee \neg H \quad (11)$$

$$\neg Y \vee H \tag{12}$$

$$Y \vee \neg C \tag{13}$$

$$\neg Y \vee C \tag{14}$$

$$Y \vee \neg F \tag{15}$$

$$\neg Y \vee F \tag{16}$$

For gene *Z*, we have

$$\neg B \vee \neg H \vee Z \tag{17}$$

$$B \vee \neg Z \tag{18}$$

$$H \vee \neg Z \tag{19}$$

$$Z \vee \neg I \tag{20}$$

$$\neg Z \vee I \tag{21}$$

In order to find some new information which is between "*D* and *E*", "*H*, *C*, and *F*", "*X* and *Y*", "*Y* and *Z*" and "*X* and *Z*", we apply the resolution principle method. Then some of the Resolvents are as follows: (Note that we have not listed all of the Resolvents.)

(5)&(8) $D \vee \neg E$ (22)

(6)&(7) $\neg D \vee E$ (23)

(5)&(10) $\neg X \vee Y$ (24)

(6)&(9) $X \vee \neg Y$ (25)

(11)&(14) $\neg H \vee C$ (26)

(12)&(13) $\neg C \vee H$ (27)

(13)&(16) $\neg C \vee F$ (28)

(14)&(15)	$C \vee \neg F$	(29)
(26)&(28)	$\neg H \vee F$	(30)
(27)&(29)	$H \vee \neg F$	(31)
(11)&(19)	$Y \vee \neg Z$	(32)
(12)&(17)	$\neg B \vee \neg Y \vee Z$	(33)

Clause (22) is equivalent to $E \rightarrow D$ and $\neg D \rightarrow \neg E$, thus we derive that if E is activated, D will be activated and if D is inhibited, E will be inhibited. Similarly, we know that Clause (23) is equivalent to $D \rightarrow E$ and $\neg E \rightarrow \neg D$, thus we derive that if D is activated, E will be activated and if E is inhibited, D will be inhibited. Furthermore, we should note that Clause (33) is equivalent to $(B \wedge Y) \rightarrow Z$ and $\neg Z \rightarrow \neg B \vee \neg Y$, thus we can derive that if both B and Y are activated, Z will be activated and if Z is inhibited, B or Y will be inhibited. Certainly there are still other logical consequences, such as

(2)&(6)	$\Leftrightarrow$	$A \vee \neg D$
(3)&(8)	$\Leftrightarrow$	$B \vee \neg E$
(9)&(11)	$\Leftrightarrow$	$D \vee \neg H$
⋮	⋮	⋮

Yet these logical consequences above seem trivial since we can obtain them just by observing each gene's regulators and targets.

Some data used to reconstruct a gene regulatory network in Yu, *et al*.'s article are available on http://bioinfo.mbb.yale.edu/regulation/TIG/. [YLQG2003] Now, we list the data which is of type 5 as Table 2.2-1. Note that every word, such as YBR049W, denotes a gene of the yeast *Saccharomyces cerevisiae*.

Table 2.2-1 Some data of type 5

1. Gene: YBR049W; Targets:

YAL031C	YBL013W	YBL014C	YBL058W	YBL059CA	YDL212W	YDL213C
YDR313C	YDR322CA	YDR323C	YDR465C	YDR466W	YFL005W	YGL088W
YGL119W	YGL120C	YGL181W	YGL182C	YGR241C	YGR243W	YHR026W
YIL075C	YIL108W	YIL109C	YIR002C	YIR003W	YJL001W	YJL002C
YJR090C	YKL081W	YKL082C	YKL104C	YKL146W	YKR068C	YKR069W
YLL040C	YLR223C	YLR224W	YLR378C	YLR380W	YML001W	YML129C
YMR121C	YMR122WA	YMR186W	YMR214W	YMR281W	YNL008C	YNL090W
YNL161W	YNL169C	YNL171C	YNL262W	YNL263C	YOR149C	YOR150W
YPL182C	YPL204W	YPL206C	YPR025C	YPR026W	YPR052C	YPR053C
YPR055W	YPR076W	YPR163C	YPR164W	YPR181C	YPR182W	

2. Gene: YGL181W; Targets:

YDL098C	YEL018W	YEL019C	YLR428C	YLR430W

3. Gene: YDL106C; Targets:

YBR044C	YBR238C	YDR023W	YDR307W	YGR205W	YJL161W	YKL106W
YMR055C	YPR194C					

4. Gene: YLR403W; Targets:

YJL067W	YJL068C	YOR151C	YPL085W	YPL086C	YPL183WA	YPL184C

5. Gene: YBR083W; Targets:

YLR403W

By applying the resolution principle method to the data above, we can obtain the results as follows.

$$\text{YBR049W} \vee \neg\text{YDL098C} \quad \Leftrightarrow \quad \neg\text{YBR049W} \rightarrow \neg\text{YDL098C}$$

$$\neg\text{YBR049W} \vee \text{YDL098C} \quad \Leftrightarrow \quad \text{YBR049W} \rightarrow \text{YDL098C}$$

$$\text{YBR049W} \vee \neg\text{YEL018W} \quad \Leftrightarrow \quad \neg\text{YBR049W} \rightarrow \neg\text{YEL018W}$$

$$\neg\text{YBR049W} \vee \text{YEL018W} \quad \Leftrightarrow \quad \text{YBR049W} \rightarrow \text{YEL018W}$$

$YBR049W \vee \neg YEL019W \Leftrightarrow \neg YBR049W \rightarrow \neg YEL019W$

$\neg YBR049W \vee YEL019W \Leftrightarrow YBR049W \rightarrow YEL019W$

$YBR049W \vee \neg YLR428C \Leftrightarrow \neg YBR049W \rightarrow \neg YLR428C$

$\neg YBR049W \vee YLR428C \Leftrightarrow YBR049W \rightarrow YLR428C$

$YBR049W \vee \neg YLR430W \Leftrightarrow \neg YBR049W \rightarrow \neg YLR430W$

$\neg YBR049W \vee YLR430W \Leftrightarrow YBR049W \rightarrow YLR430W$

$YBR083W \vee \neg YJL067W \Leftrightarrow \neg YBR083W \rightarrow \neg YJL067W$

$\neg YBR083W \vee YJL067W \Leftrightarrow YBR083W \rightarrow YJL067W$

$YBR083W \vee \neg YJL068W \Leftrightarrow \neg YBR083W \rightarrow \neg\ YJL068W$

$\neg YBR083W \vee YJL068W \Leftrightarrow YBR083W \rightarrow YJL068W$

$YBR083W \vee \neg YOR151C \Leftrightarrow \neg YBR083W \rightarrow \neg YOR151C$

$\neg YBR083W \vee YOR151C \Leftrightarrow YBR083W \rightarrow YOR151C$

$YBR083W \vee \neg YPL085W \Leftrightarrow \neg YBR083W \rightarrow \neg YPL085W$

$\neg YBR083W \vee YPL085W \Leftrightarrow YBR083W \rightarrow YPL085W$

$YBR083W \vee \neg YPL086W \Leftrightarrow \neg YBR083W \rightarrow \neg YPL086W$

$\neg YBR083W \vee YPL086W \Leftrightarrow YBR083W \rightarrow YPL086W$

$YBR083W \vee \neg YPL183WA \Leftrightarrow \neg YBR083W \rightarrow \neg YPL183WA$

$\neg YBR083W \vee YPL183WA \Leftrightarrow YBR083W \rightarrow YPL183WA$

$YBR083W \vee \neg YPL184C \Leftrightarrow \neg YBR083W \rightarrow \neg YPL184C$

$\neg YBR083W \vee YPL184C \Leftrightarrow YBR083W \rightarrow YPL184C$

The result above can be shown in Figure 2.2-8.

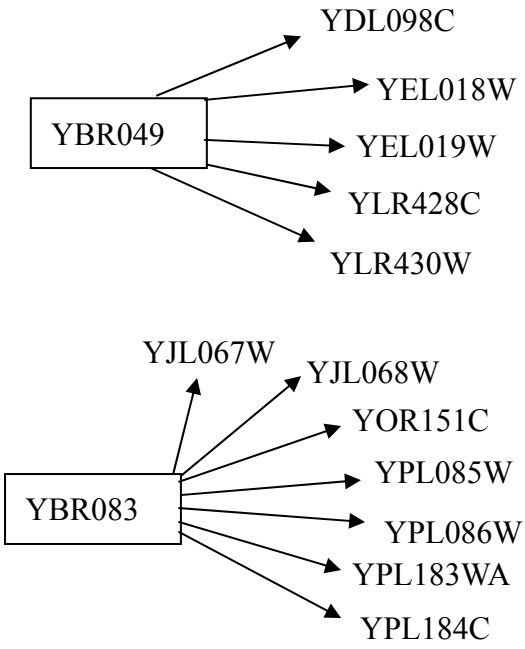


Figure 2.2-8 The result shown in a graphic way

We should notice that the result above all can't be easily and directly obtained from the original data.

After introducing the resolution principle method and the possible 6 types of relations between genes, regulators and targets, we shall proceed to study the important model – the Boolean gene regulatory network in the following chapter. Certainly, the resolution principle method will play a quite important role.

Chapter 3
Symbolic Logic on Boolean Gene Regulatory Network

In Chapter 2 we have introduced the regulator and target model. Next we are going to introduce a more complicated model – Boolean gene regulatory network. The general concept of the Boolean gene regulatory network is that every gene is treated as a button. While a button is pushed, other genes may react. The interactions between genes defined here are again activations and inhibitions. Note that we say *a gene A is activated by a gene B* if *A* is activated because of *B*'s affection, while we say that *A* is *inhibited* by a gene *B* if *A* is inhibited because of *B*'s affection.

In this chapter, we will define the Boolean gene regulatory network in Section 3.1. We will introduce the state determination problem in Section 3.2. Finally, we will introduce the implicit interaction finding problem at Section 3.3.

3.1 Boolean Gene Regulatory Network

The Boolean gene regulatory network model is different from the regulator and target model introduced in Chapter 2. From the regulator and target model, we only know the regulators and targets of each given gene, that is, the perturbation data. However, the Boolean gene regulatory network model uses a network structure to represent the interactions between given genes. A Boolean gene regulatory network is a directed graph $G(V, E)$ describing how genes interact with one another, where V denotes a set of nodes representing all genes and E denotes a set of all the directed edges

representing the interactions between genes. As we indicated before, a gene may be activated or inhibited by another gene, or a group of genes. Since the interaction is either activating or inhibiting, it is natural to represent these interactions by Boolean functions and this is why we call the network a Boolean gene regulatory network. Besides, in this model, the AND operation is used.

An example of the kind of Boolean gene regulatory network is shown in Figure 3.1-1.

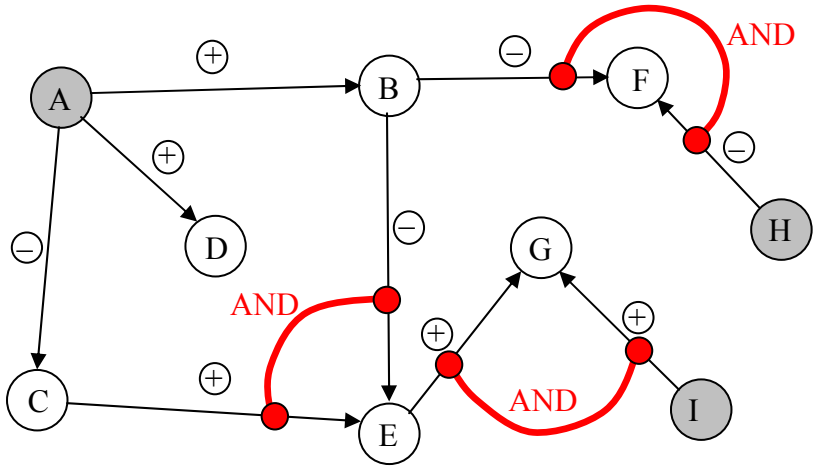


Figure 3.1-1 A Boolean gene regulatory network

In the graph representing our Boolean gene regulatory network, each node represents a gene. Each edge is a directed edge. If there is an edge directed from Node *A* to Node *B* with a sign ⊕ on the edge, it means that Gene *A* activates *B* in the following sense: If *A* is activated, *B* is activated; if *A* is not activated, *B* is not activated. If there is an edge from Node *A* to Node *B* with a sign ⊖ on the edge, it means Gene *A* inhibits *B* in the following sense: If *A* is activated, *B* is inhibited; if *A* is not activated, *B* is activated.

In the directed graph, it is possible that the indegree of a node is larger than one. This means that this particular gene is affected by more than one gene. For example, in Figure 3.1-1, both nodes E and F are such nodes. In our model, the incoming edges are all joined by AND operation. In other words, we allow the Boolean gene regulatory networks to contain AND symbols so that we can express the following kind of statements: Suppose nodes g_1, g_2, …, g_k affect g_m, that is, each g_i, $1 \le i \le k$, activates or inhibits g_m, then g_m is activated only if each g_i, causes an activation effect on g_m. For example, in Figure 3.1-1, E will be activated only if B is inhibited and C is activated; G will be activated only if E and I are both activated; F will be activated only if B and H are both inhibited. In this thesis, we will not allow OR to exist to simplify the discussion although in a general Boolean gene regulatory network, OR is allowed. We like to point out that our approach can handle the situation when OR is allowed very well.

It is appropriate and also clear for us to use symbolic logic to express the meanings of $\oplus$ and $\ominus$ as follows:

(1) For the edge $A \xrightarrow{\oplus} B$, the Boolean formula corresponding to it consists of two formulas:

$$A \to B$$

and $$\neg A \to \neg B$$

Or, equivalently,

$$\neg A \vee B$$

$$A \vee \neg B .$$

(2) For the edge $A \xrightarrow{\ominus} B$, the Boolean formula corresponding to it consists of two formulas:

$$A \rightarrow \neg B$$
$$\neg A \rightarrow B$$

Or, equivalently,

$$\neg A \vee \neg B$$
$$A \vee B$$

Consider all of the edges going out from A of the graph in Figure 3.1-1. We have the following formulas:

$$A \rightarrow B$$
$$\neg A \rightarrow \neg B$$
$$A \rightarrow D$$
$$\neg A \rightarrow \neg D$$
$$A \rightarrow \neg C$$
$$\neg A \rightarrow C$$

Consider Node E. We now have

$$(C \wedge \neg B) \rightarrow E$$
$$(\neg C \vee B) \rightarrow \neg E$$

Finally, consider Node F. We have

$(\neg B \wedge \neg H) \rightarrow F$
$(B \vee H) \rightarrow \neg F$

For the entire graph in Figure 3.1-1, we have the following logical formulas describing the regulatory network of these genes.

$$\neg A \vee B \quad (1)$$

$$A \vee \neg B \quad (2)$$

$$\neg A \vee D \quad (3)$$

$$A \vee \neg D \quad (4)$$

$$\neg A \vee \neg C \quad (5)$$

$$A \vee C \quad (6)$$

$$B \vee H \vee F \quad (7)$$

$$\neg B \vee \neg F \quad (8)$$

$$\neg H \vee \neg F \quad (9)$$

$$B \vee \neg C \vee E \quad (10)$$

$$C \vee \neg E \quad (11)$$

$$\neg B \vee \neg E \quad (12)$$

$$\neg E \vee \neg I \vee G \quad (13)$$

$$E \vee \neg G \quad (14)$$

$$I \vee \neg G \quad (15)$$

Let us further give another constraint on the graph representing our Boolean gene regulatory network: There can be no cycle in the graph. If there is a cycle, it is possible to have the case shown in Figure 3.1-2.

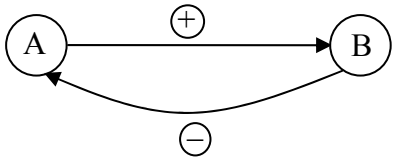


Figure 3.1-2 A cycle in a Boolean gene regulatory network

If this case is present, it means that if *A* is activated, *B* is activated. Yet if *B* is activated, *A* is inhibited. On the other hand, if *A* is not activated, *B* is not activated. If B is not activated, A will be activated. These are contradictory because there is no state which satisfies the rules. If we use symbolic logic rules, $A \xrightarrow{\oplus} B$ corresponds to the following logic formulas:

$$\neg A \vee B \qquad (\#1)$$

$$A \vee \neg B \qquad (\#2)$$

The edge $B \xrightarrow{\ominus} A$ corresponds to the following logic formulas:

$$\neg B \vee \neg A \qquad (\#3)$$

$$B \vee A \qquad (\#4)$$

It can easily be shown that clauses (#1), (#2), (#3) and (#4) are inconsistent. Therefore we do not allow cycles to simplify our discussion and avoid trouble. Similarly, self-loops are not allowed. It should be noted that not every cycle may produce inconsistency.

Let us summarize all of the rules about how we represent the Boolean gene regulatory system.

1. Each Boolean gene regulatory system is represented by a directed graph $G(V, E)$ in which every node *v* corresponds to a gene and each edge e_{ij} indicates that gene *j* is affected by gene *i*.

2. Edge $A \xrightarrow{\oplus} B$ means that if *A* is activated, *B* will be activated and if *A* is not activated, *B* will not be activated. To put this in symbolic logic format, this correspond to the following two formulas:

$$A \to B \ \ (i.e., \neg A \vee B)$$
$$\neg A \to \neg B \ \ (i.e., A \vee \neg B)$$

3. Edge $A \xrightarrow{\ominus} B$ means that if *A* is activated, *B* will not be activated and if *A* is not activated, *B* will be activated. To put this in symbolic logic format, this correspond to the following two formulas:

$$A \to \neg B \ \ (i.e., \neg A \vee \neg B)$$
$$\neg A \to B \ \ (i.e., A \vee B)$$

4. If the indegree of a node in the graph is more than one, all of the incoming edges are joined by AND operation.
5. There is no cycle in the graph.

In the following, we shall discuss a problem concerning with our model: *the state determination problem*. Since in our model, there is no cycle, there must be some nodes whose indegrees are 0 (See Appendix 3.2.). That is, for these nodes, there are only outgoing edges, no incoming nodes. In other words, these genes only

control others and are not controlled by any other genes. We shall call these genes *key regulators*. We shall now show that once the states of all of these genes are known, the states of all genes can be found. That is, once we know whether the key regulators are activated or inhibited, we will be able to determine whether each other gene is activated or inhibited. (A mathematical proof can be seen at Appendix 3.2.).

Since there is no cycle in the directed graph, we may view the graph as a staged graph with the key regulators in the first stage. For instance, we may redraw the graph in Figure 3.1-1 as a staged graph as shown in Figure 3.1-3, Figure 3.1-4, Figure 3.1-5 and Figure 3.1-6. The states of all nodes are determined stage by stage. Note that each node in a staged graph is assigned a value either '1' or '0', where 1 denotes that the node is activated and 0 denotes that it is inhibited. Now, let us see Example 3.1-1.

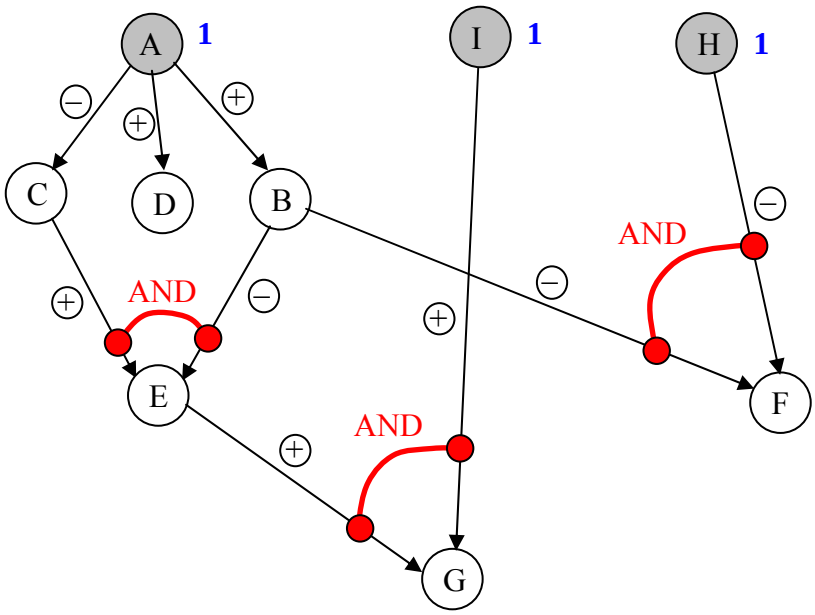


Figure 3.1-3 The staged graph of the first stage

Example 3.1-1

At the first stage shown in Figure 3.1-3, *A*, *H* and *I* are assumed to be activated.

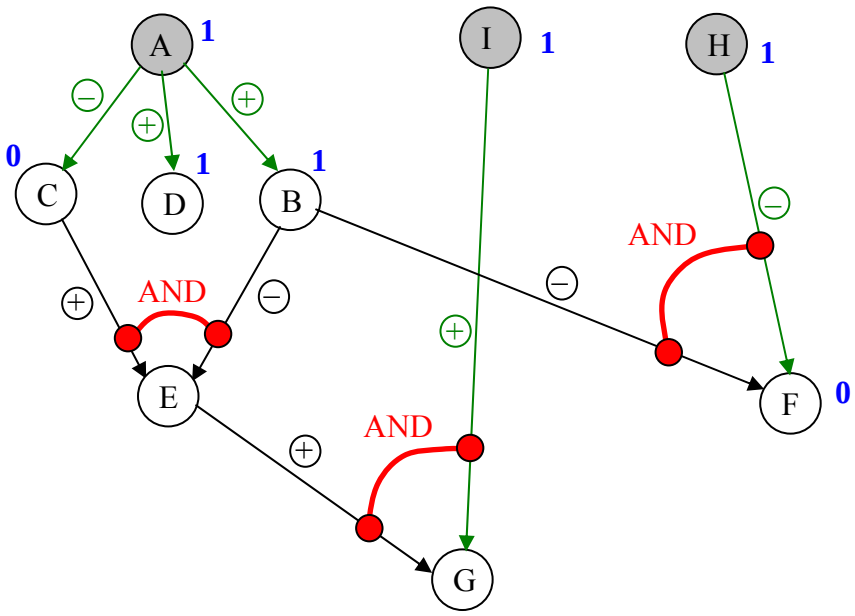


Figure 3.1-4 The staged graph of the second stage

At the second stage shown in Figure 3.1-4, nodes *B* and *D* are activated and *C* is inhibited since *A* is activated. *F* is inhibited since one of the conditions for *F* to be activated is that *H* has to be inhibited.

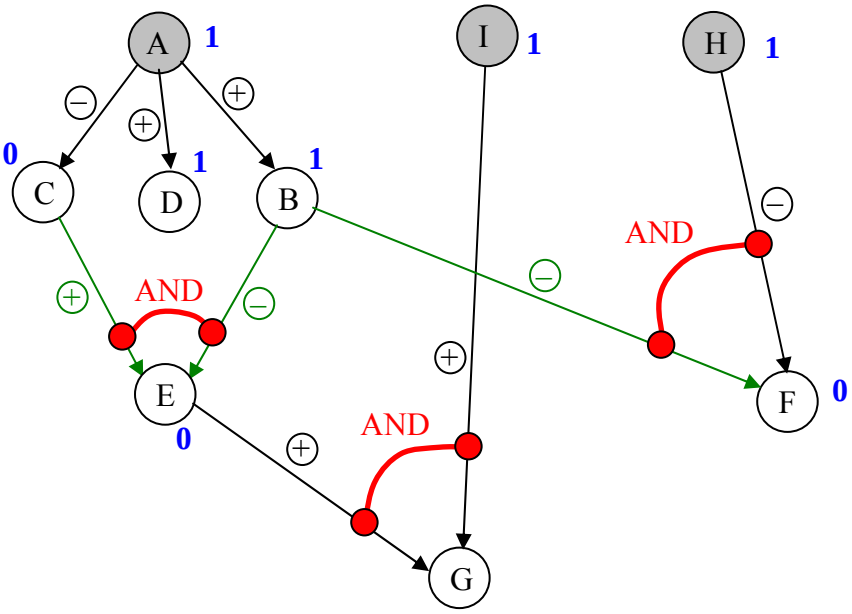


Figure 3.1-5 The staged graph of the third stage

At the third stage shown in Figure 3.1-5, *E* is inhibited because one of the necessary conditions for *E* to be activated is that *C* has to be activated.

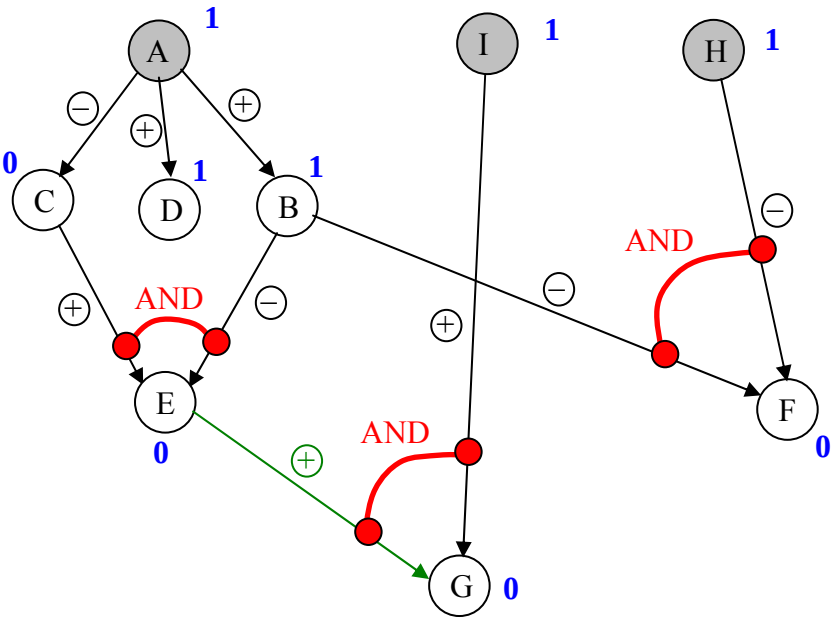


Figure 3.1-6 The staged graph of the fourth stage

At the fourth stage shown in Figure 3.1-6, G is inhibited since one of the necessary conditions for G to be activated is that E is activated. At this stage, the states of all nodes are determined. We now summarize the result in Table 3.1-1 as follows.

Table 3.1-1 The result of Example 3.1-1

B	Activated
C	Inhibited
D	Activated
E	Inhibited
F	Inhibited
G	Inhibited

Now we can determine the states of all nodes stage by stage by viewing the graph as a staged graph and the states of all nodes are known at the final stage. To solve the state determination problem, that is, to determine the state of each node, the depth-first-search method with some modifications [M89] can be adopted. (See Appendix 3.1 for the detail and the algorithm)

Let us see Example 3.1-2 to understand more.

Example 3.1-2

A Boolean gene regulatory network $G(V,E)$ is given and shown in Figure 3.1-7 as follows.

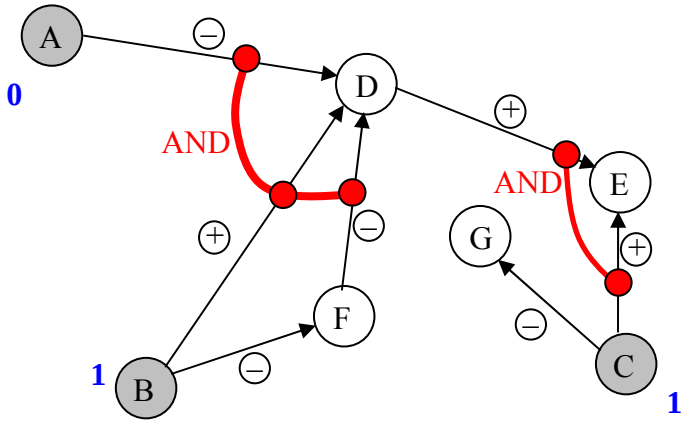


Figure 3.1-7 A Boolean gene regulatory network

We may view the graph in Figure 3.1-7 as a staged graph with the key regulators *A*, *B* and *C*. Thus, we may redraw the graph in Figure 3.1-7 as a staged graph as shown in Figure 3.1-8, Figure 3.1-9, Figure 3.1-10 and Figure 3.1-11.

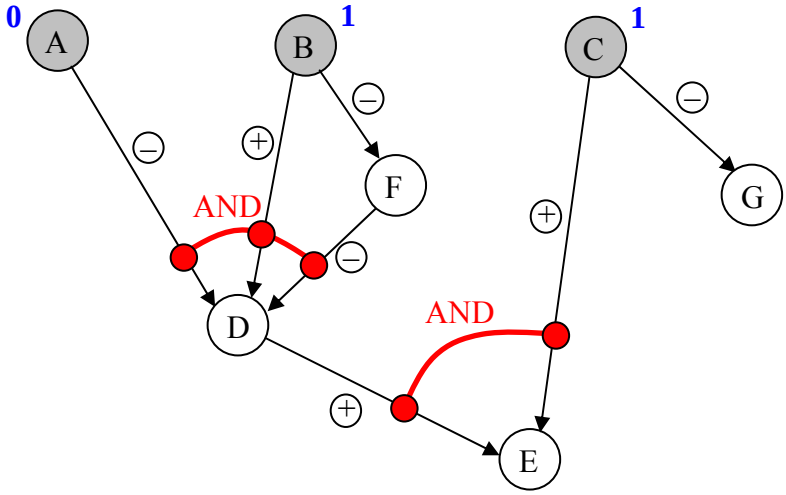


Figure 3.1-8 The staged graph of the first stage

At the first stage shown in Figure 3.1-8, *A* is assumed to be inhibited, and *B* and *C* are assumed to be activated.

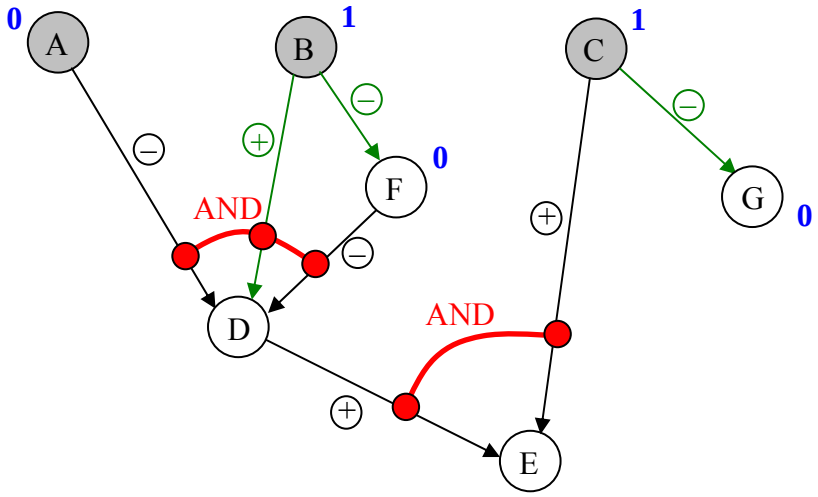


Figure 3.1-9 The staged graph of the second stage

At the second stage shown in Figure 3.1-9, *F* is inhibited since *B* is inhibited. *G* is inhibited since *C* is activated.

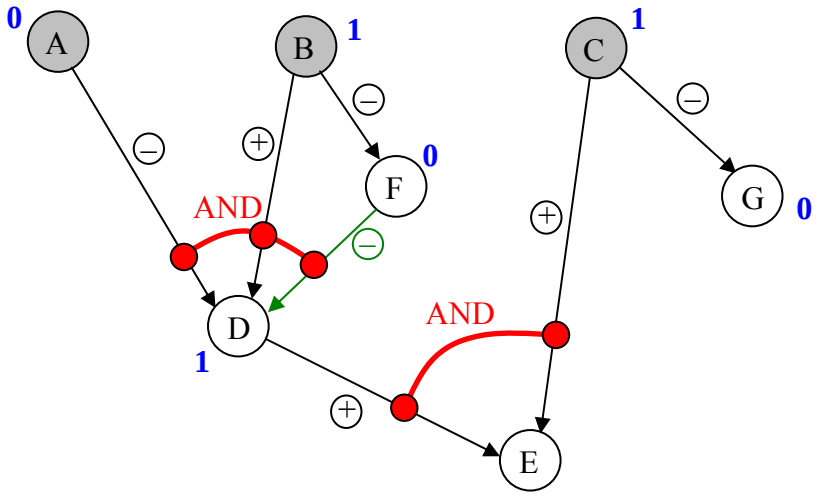


Figure 3.1-10 The staged graph of the third stage

At the third stage shown in Figure 3.1-10, *D* is activated since all of the necessary conditions for *D* to be activated are satisfied.

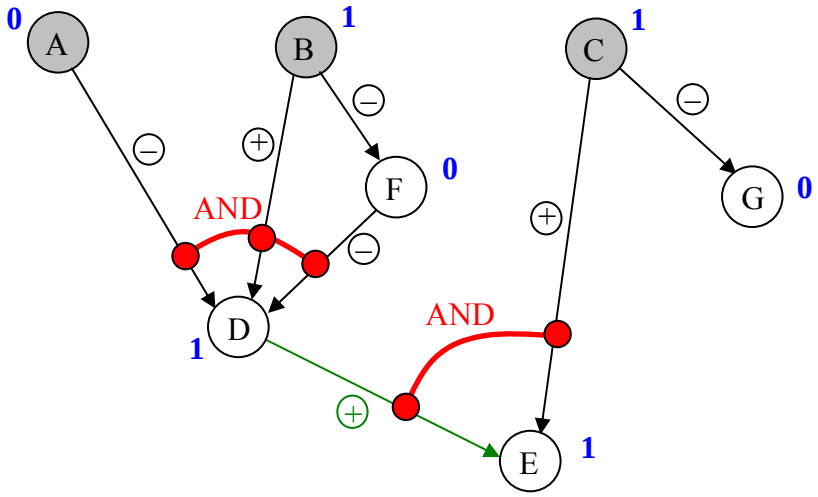


Figure 3.1-11 The staged graph of the fourth stage

At the fourth stage shown in Figure 3.1-11, *E* is activated since all of the necessary conditions for *D* to be activated are satisfied. This is the final stage since the states of all the nodes in *G* are all determined. We now summarize the result in Table 3.1-2 as follows.

Table 3.1-2 The result of Example 3.1-2

D	Activated
E	Activated
F	Inhibited
G	Inhibited

3.2 The State Determination Problem Seen as a Logical Consequences Finding Problem

Let us redraw the Boolean gene regulatory network in Figure 3.1-1, in Figure 3.2-1.

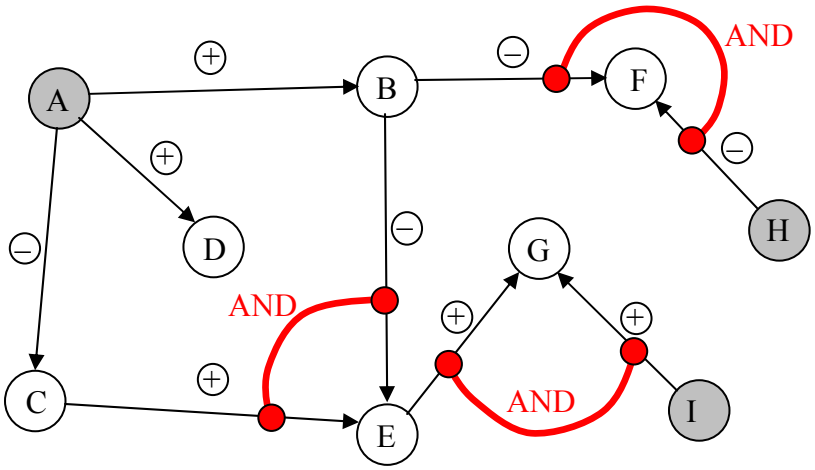


Figure 3.2-1 A Boolean gene regulatory network

Let us now describe the network by logic formulas as follows as we did in Section 3.1.

$$\neg A \vee B \quad (1)$$

$$A \vee \neg B \quad (2)$$

$$\neg A \vee D \quad (3)$$

$$A \vee \neg D \quad (4)$$

$$\neg A \vee \neg C \quad (5)$$

$$A \vee C \quad (6)$$

$$B \vee H \vee F \quad (7)$$

$$\neg B \vee \neg F \quad (8)$$

$$\neg H \vee \neg F \quad (9)$$

$$B \vee \neg C \vee E \quad (10)$$

$$C \vee \neg E \quad (11)$$

$$\neg B \vee \neg E \quad (12)$$

$$\neg E \vee \neg I \vee G \quad (13)$$

$$E \vee \neg G \quad (14)$$

$$I \vee \neg G \tag{15}$$

To solve the state determination problem, we may simply apply the resolution principle method. Assume that *A*, *I* and *H* are all activated. Then we add the following three clauses:

$$A \tag{16}$$

$$I \tag{17}$$

$$H \tag{18}$$

By applying the resolution principle method introduced in Chapter 2, we shall have:

(16)&(1)	B	(19)
(16)&(3)	D	(20)
(16)&(5)	$\neg C$	(21)
(19)&(8)	$\neg F$	(22)
(21)&(11)	$\neg E$	(23)
(23)&(14)	$\neg G$	(24)

Thus we now have the following conclusion.

Table 3.2-1 The result corresponding to Figure 3.2-1

B	Activated
C	Inhibited
D	Activated

E	Inhibited
F	Inhibited
G	Inhibited

This result is the same as the result of Example 3.1-1.

Let us redraw the Boolean gene regulatory network in Figure 3.1-7, in Figure 3.2-2.

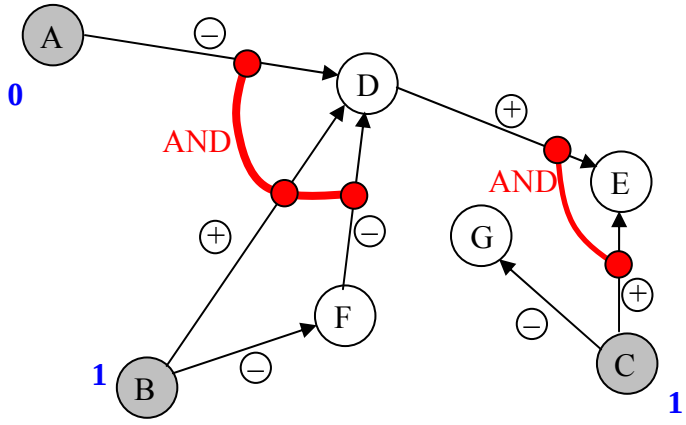


Figure 3.2-2 A Boolean gene regulatory network

For the graph in Figure 3.2-2, we have the following logical formulas describing the regulatory system of these genes.

$$A \vee \neg B \vee F \vee D \tag{25}$$

$$\neg A \vee \neg D \tag{26}$$

$$B \vee \neg D \tag{27}$$

$$\neg F \vee \neg D \tag{28}$$

$$\neg B \vee \neg F \tag{29}$$

$$B \vee F \tag{30}$$

$$\neg C \vee \neg G \tag{31}$$

$$C \vee G \tag{32}$$

$$\neg D \vee \neg C \vee E \tag{33}$$

$$D \vee \neg E \tag{34}$$

$$C \vee \neg E \tag{35}$$

Again, in order to solve the state determination problem, we may apply the resolution principle method. Assume that A is inhibited, while B and C are both activated. Then we add the following three more clauses:

$$\neg A \tag{36}$$

$$B \tag{37}$$

$$C \tag{38}$$

By applying the resolution principle method introduced in Chapter 2, we shall have:

(38)&(31)	$\neg G$	(39)
(36)&(25)	$\neg B \vee F \vee D$	(40)
(37)&(40)	$F \vee D$	(41)
(29)&(37)	$\neg F$	(42)
(41)&(42)	D	(43)
(41)&(33)	$\neg C \vee E \vee F$	(44)
(44)&(38)	$E \vee F$	(45)
(45)&(42)	E	(46)

Thus we now have the following conclusion.

Table 3.2-2 The result corresponding to Figure 3.2-2

D	Activated
E	Activated
F	Inhibited
G	Inhibited

This result is just the same as the result of Example 3.1-2.

3.3 The Implicit Interaction Finding Problem in a Boolean Gene Regulatory Network

Let us consider the Boolean gene regulatory network shown in Figure 3.3-1.

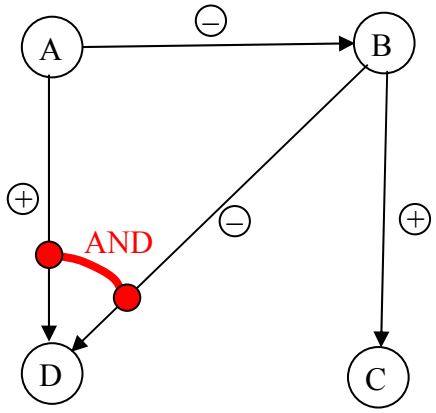


Figure 3.3-1 A Boolean gene regulatory network

From this directed graph, a naïve reader may easily see that, for instance, if *A* is activated, *B* is not activated. Actually, we can see more. For instance, we can

actually show that if D is activated, C must be inhibited. The reason is the following: One of the conditions that D is activated is that B is inhibited. Yet, if B is inhibited, C must be inhibited. Thus the conclusion is drawn.

The implicit interaction finding problem is to derive more interactions from a Boolean gene regulatory network. This can be done by using the resolution principle method introduced in Chapter 2. Let's see an example as follows.

The basic interactions which can be directly derived from the directed graph in Figure 3.3-1 are as follows:

$$A \to \neg B \quad \Leftrightarrow \quad \neg A \vee \neg B \tag{1}$$

$$\neg A \to B \quad \Leftrightarrow \quad A \vee B \tag{2}$$

$$\neg B \to \neg C \quad \Leftrightarrow \quad B \vee \neg C \tag{3}$$

$$B \to C \quad \Leftrightarrow \quad \neg B \vee C \tag{4}$$

$$A \to D \quad \Leftrightarrow \quad \neg A \vee D \tag{5}$$

$$\neg A \to \neg D \quad \Leftrightarrow \quad A \vee \neg D \tag{6}$$

$$B \to \neg D \quad \Leftrightarrow \quad \neg B \vee \neg D \tag{7}$$

$$\neg B \to D \quad \Leftrightarrow \quad B \vee D \tag{8}$$

$$(A \wedge \neg B) \to D \quad \Leftrightarrow \quad \neg A \vee B \vee D \tag{9}$$

$$\neg(A \wedge \neg B) \to \neg D \quad \Leftrightarrow \quad (A \wedge \neg B) \vee \neg D \tag{10}$$

Note that

$$(10) \quad \Leftrightarrow \quad (A \vee \neg D) \wedge (\neg B \vee \neg D),$$

so from (10) we have

$$A \vee \neg D \tag{11}$$

$$\neg B \vee \neg D \tag{12}$$

Because (11) is the same as (6), (12) is the same as (7), and (10) is equivalent to the conjunction of (11) and (12), we can eliminate (10), (11) and (12). Then, we can derive the following logical consequences from formulas (1) to (9) as follows.

(2)&(4)	$\Rightarrow$	$A \vee C$	(13)
(1)&(3)	$\Rightarrow$	$\neg A \vee \neg C$	(14)
(3)&(7)	$\Rightarrow$	$\neg D \vee \neg C$	(15)
(15)	$\Leftrightarrow$	$C \rightarrow \neg D$	(16)
	$\Leftrightarrow$	$D \rightarrow \neg C$	(17)

Therefore, we derive that either if C is activated, D will be inhibited or if D is activated, C will be inhibited. We can redraw Figure 3.3-1 in Figure 3.3-2 and 3.3-3 as follows.

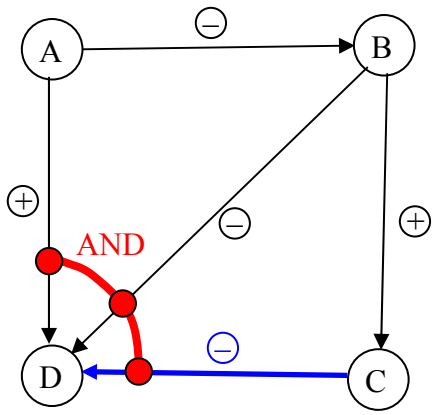


Figure 3.3-2 A Boolean gene regulatory network redrawn from Figure 3.3-1

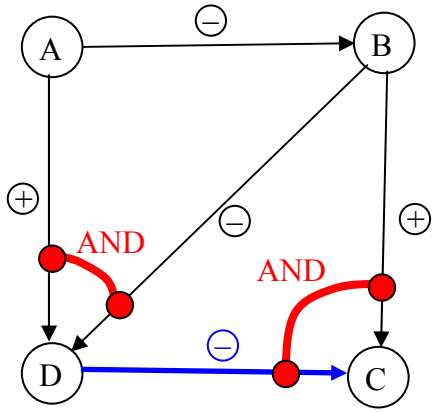


Figure 3.3-3 A Boolean gene regulatory network redrawn from Figure 3.3-1

$C \xrightarrow{\ominus} D$ or $D \xrightarrow{\ominus} C$ is a new interaction found by the resolution principle method. Here we call $C \xrightarrow{\ominus} D$ and $D \xrightarrow{\ominus} C$ *implicit interactions* in the Boolean gene regulatory network in Figure 3.3-1. It is quite interesting that we cannot know the interaction between *C* and *D* directly from the original Boolean gene regulatory network in Figure 3.3-1, because there doesn't exist any path from *C* to *D* or from *D* to *C*. Yet by using the resolution principle method, we can find such an implicit interaction. Hence, biologists may be encouraged to experiment on the implicit interactions.

Let see another example. Consider the Boolean gene regulatory network shown in Figure 3.3-4.

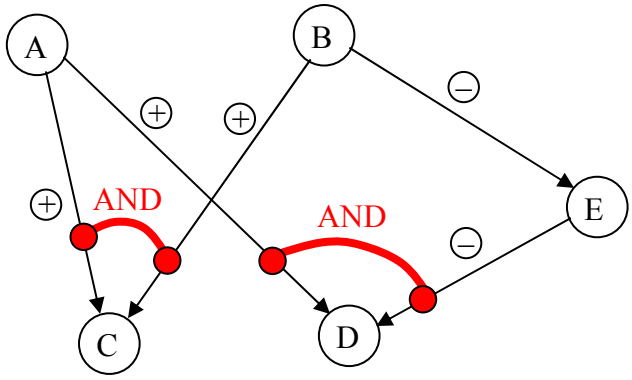


Figure 3.3-4 A Boolean gene regulatory network

The basic interactions which can be directly derived from the directed graph in Figure 3.3-4 are as follows:

$$(A \wedge \neg E) \rightarrow D \quad \Leftrightarrow \quad \neg A \vee E \vee D \qquad (18)$$

$$\neg(A \wedge \neg E) \rightarrow \neg D \quad \Leftrightarrow \quad A \vee \neg D \qquad (19)$$

$$\neg E \vee \neg D \qquad (20)$$

$$(A \wedge B) \rightarrow C \quad \Leftrightarrow \quad \neg A \vee \neg B \vee C \qquad (21)$$

$$\neg(A \wedge B) \rightarrow \neg C \quad \Leftrightarrow \quad A \vee \neg C \qquad (22)$$

$$B \vee \neg C \qquad (23)$$

$$B \rightarrow \neg E \quad \Leftrightarrow \quad \neg B \vee \neg E \qquad (24)$$

$$\neg B \rightarrow E \quad \Leftrightarrow \quad B \vee E \qquad (25)$$

Then, we can derive the following logical consequences from formulas (18) to (25) as follows.

$$(18)\&(22) \quad \Rightarrow \quad \neg C \vee D \vee E \qquad (26)$$

(23)&(24)	$\Rightarrow$	$\neg C \vee \neg E$	(27)
(24)&(26)	$\Rightarrow$	$\neg B \vee \neg C \vee D$	(28)
(21)&(25)	$\Rightarrow$	$\neg A \vee C \vee E$	(29)
(19)&(29)	$\Rightarrow$	$C \vee \neg D \vee E$	(30)
(20)&(30)	$\Rightarrow$	$C \vee \neg D$	(31)
(20)&(25)	$\Rightarrow$	$B \vee \neg D$	(32)
(27)	$\Leftrightarrow$	$C \rightarrow \neg E$	(33)
	$\Leftrightarrow$	$E \rightarrow \neg C$	(34)
(31)	$\Leftrightarrow$	$\neg C \rightarrow \neg D$	(35)
	$\Leftrightarrow$	$D \rightarrow C$	(36)

Therefore, we derive that either if C is activated, E will be inhibited or if E is activated, C will be inhibited from formulas (33) and (34). We can also derive that either if C is inhibited, D will be inhibited or if D is activated, C will be activated from formulas (35) and (36). We then redraw Figure 3.3-4 in Figure 3.3-5, 3.3-6, 3.3-7 and 3.3-8 as follows.

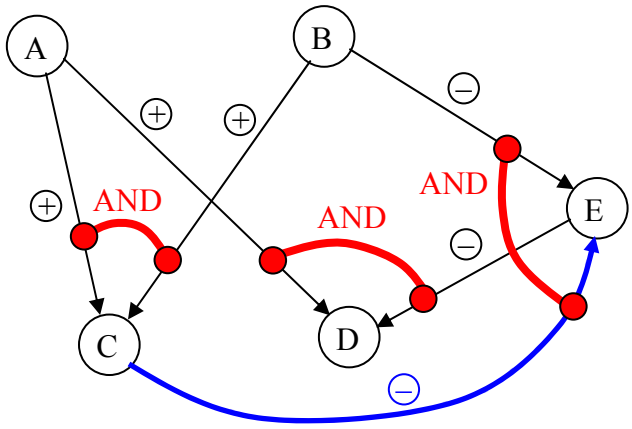


Figure 3.3-5 A Boolean gene regulatory network redrawn from Figure 3.3-4

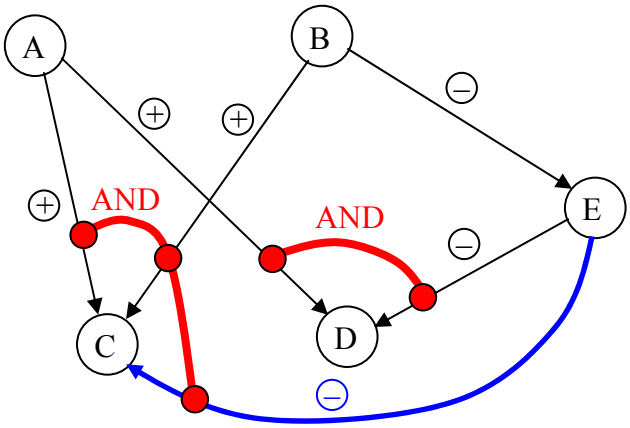


Figure 3.3-6 A Boolean gene regulatory network redrawn from Figure 3.3-4

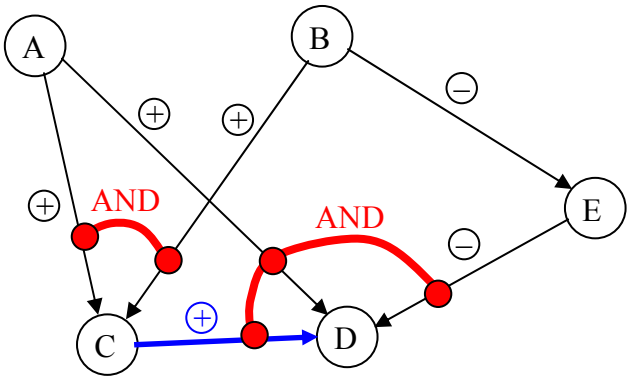


Figure 3.3-7 A Boolean gene regulatory network redrawn from Figure 3.3-4

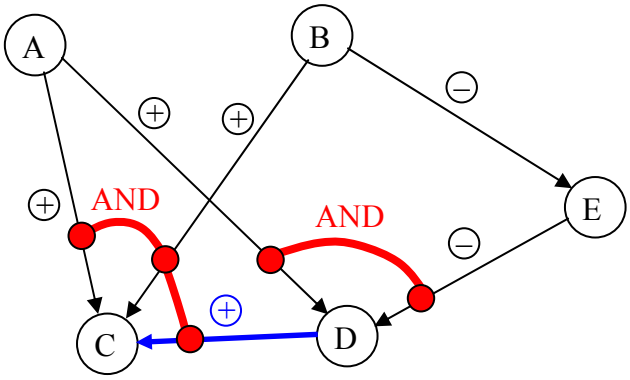


Figure 3.3-8 A Boolean gene regulatory network redrawn from Figure 3.3-4

Now, we know that the implicit interactions can be found by the resolution principle method. Compared with the traditional method using network models to find direct and indirect interactions, the resolution principle method can find even more implicit interactions.

Chapter 4
Conclusions and Future Work

In this chapter, we will summarize our discussions in this thesis. Some remarks and directions for the further research will also be given.

4.1 Conclusions

Up to now, we have discussed several models to analyze gene regulatory networks. In order to discuss the interactions between genes directly, we have introduced symbolic logic in Chapter 2. Due to the idea that we may try to find the implicit interactions between given genes, we have proposed the resolution principle method. By applying this method, we can find possible implicit interactions from the given perturbation data. The resolution principle method can also be applied to a Boolean gene regulatory network which has been introduced in Chapter 3. A lot of surprising results have been shown in the examples in this chapter.

Our method for determining the states of genes in a Boolean gene regulatory network was proposed in Chapter 3 and Appendix 3.1. As long as the key regulators are given, we can determine the state of each gene. This has been proved in Appendix 3.2 and we can agree on this inference by the examples given in Chapter 3. To sum up, we conclude that symbolic logic is quite useful in analyzing the gene regulatory networks and the resolution principle method can bring amazing new information.

4.2 Future Work

In the future, we may release the restriction that only AND gates are allowed on the Boolean gene regulatory networks. Other Boolean gates, such as exclusive-OR gates, OR gates, etc., can be also used. Thus, we can apply the resolution principle method to find all possible implicit interactions existing among given genes. Besides, since each gene interaction requires a particular time period, we can make use of this concept in our analysis. By making use of this concept, somehow we can infer that some interactions are possible while some interactions are impossible.

In the Boolean gene regulatory network model, there exists a problem called *the identification problem*, which is to determine whether there exists exact one Boolean gene regulatory network corresponding to the data containing the states of genes at some time t and those at some time $t + 1$ [AKMM98, AMK99, LSH2002]. This is an interesting problem which deserves to be studied in the future work. In addition, after the only Boolean gene regulatory network is found, we can apply our method, that is, the resolution principle method, to verify it and find the implicit interactions in it.

In addition, there are quite many other interactions between genes and their products. We can symbolize and formulize these interactions into logical statements. Through observing these statements, we may obtain some information new.

Now, we conclude our further research as follows:

(1) Find the logical consequences from the Boolean gene regulatory networks where all Boolean gates are allowed.

(2) Study the identification problem and try to give an efficient algorithm for finding the only one Boolean gene regulatory network corresponding to given data.

(3) Apply the resolution principle method to find the implicit interactions in the Boolean gene regulatory network mentioned in (2).

(4) Symbolize and formulize all possible gene interactions and then try to deduce logical consequences from them.

We believe that our methods proposed in this thesis will be very useful and significant in the research for gene regulatory networks.

Bibliography

[AI2001] Inference of Gene Regulatory Model by Genetic Algorithms, Ando, S. and IBA, H., Proceedings of the 2001 IEEE Congress on Evolutionary Computation Seoul, Korea, May 27-30, 2001, pp. 1-8.

[AKMM98] Identification of Gene Regulatory Networks by Strategic Gene Disruptions and Gene Overexpressions, Akutsu, T., Kuhara, S., Maruyama, O. and Miyano, S., Proceedings of the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, 1998, pp. 695-702.

[AMK99] Identification of Genetic Networks from a Small Number of Gene Expression Patterns under the Boolean Network Model, Akutsu, T., Miyano, S. and Kuhara, S., Pacific Symposium Biocomputing, 1999, pp. 17-28.

[ASN *et al*. 2003] Discovery of Novel Transcription Control Relationships with Gene Regulatory Networks Generated from Multiple-disruption Full Genome Expression Libraries, Aburatani, S., Tashiro, K., Savoie, C. J., Nishizawa, M., Hayashi, K., Ito, Y., Muta, S., Yamamoto, K., Ogawa, M., Enomoto, A., Masaki, M., Watanabe, S., Maki, Y., Takahashi, Y., Eguchi, Y., Sakaki, Y. and Kuhara, S., DNA Research, Vol. 10, 2003, pp. 1-8.

[BSPL2003] Revising Regulatory Networks: From Expression Data to Linear Causal Models, Bay, S. D., Shrager, J., Pohorille, A., and Langley, P., Journal of Biomedical Informatics, Vol. 35, 2003, pp. 289-297.

[BT2003] Evolution of Transcription Factors and the Gene Regulatory Network in Escherichia coli, Babu, M. M. and Teichmann, S. A., Nucleic Acids Research, Vol. 31, No. 4, 2003, pp. 1234-1244.

[CL73] Symbolic Logic and Mechanical Theorem Proving, Chang, C. L. and Lee, R. C. T., Academic Press, Inc., 1973.

[CLA2003] Gene Network Reconstruction Using a Distributed GA with a Backprop Local Search, Cumiskey, M. Levine, J. and Armstrong, D., Proceedings of the 1st European Workshop on Evolutionary Bioinformatics, 2003, pp. 33-43.

[D] Davidson Laboratory website. http://its.caltech.edu/~mirsky/qpcr.html

[D97] Understanding DNA and Gene Cloning – A Guide for the Curious, Drlica, K., John Wiley & Sons, Incorporation, 1997.

[DLS2000] Genetic Network Inference: from Co-Expression Clustering to Reverse Engineering, D'haeseleer, P., Liang, S. and Somogyi, R., Bioinformatics, Vol. 16, No. 8, pp. 707-726.

[DRO *et al*. 2002] A Genomic Regulatory Network for Development, Davidson, E. H., Rast, J. P., Oliveri, P., Ransick, A., Calestani, C., Yuh, C. H., Minokawa, T., Amore, G., Hinman, V., Mena, C. A., Otim, O., Brown, T., Livi, C. B., Lee, P. Y., Revilla, R., Rust, A. G., Pan, Z. J., Schilstra, M. J., Clarke, P. J. C., Arnone, M. I., Rowen, L., Cameron, R. A., McClay, D. R., Hood, L. and Bolouri, H., Science, Vol.

295, No. 5560, pp. 1669-1678.

[H69] Graph Theory, Harary, F., Addison-Wesley, Reading, MA., 1969.

[KCE2004] Inferring Gene Regulatory Networks from Raw Data - A Molecular Epistemics Approach, Kightley, D. A., Chandra, N. and Elliston, K., Pacific Symposium Biocomputing, 2004, pp. 510-520.

[KMOK2000] A Gene Network Inference Method from continuous-Value Gene Expression Data of Wild-Type and Mutants, Kyoda, K. M., Morohashi, M., Onami, S. and Kitano, H., Genome Informatics, Vol. 11, 2000, pp. 196-204.

[KSY99] Algorithms for Choosing Differential Gene Expression Experiments, Karp, R. M., Stoughton, R. and Yeung, K. Y., Proceedings of the Third Annual International Conference on Computational Molecular Biology, 1999, pp. 208-217.

[LRR *et al*. 2002] Transcriptional Regulatory Networks in Saccharomyces cerevisiae, Lee, T. I., Rinaldi, N. J., Robert, F., Odom, D. T., Joseph, Z. B., Gerber, G. K., Hannett, N. M., Harbison, C. T., Thompson, C. M., Simon, I., Zeitlinger, J., Jennings, E. G., Murray, H. L., Gordon, D. B., Ren, B., Wyrick, J. J., Tagne, J. B., Volkert, T. L., Fraenkel, E., Gifford, D. K., Young, R. A., Science, Vol. 298, 2002.

[LSH2002] On Learning Gene Regulatory Networks Under the Boolean Network Model, Lähdesmäki, H., Shmulevich, I. and Harja, O. Y., Machine Learning, Vol. 52, Issue 1-2, 2003, pp. 147-167.

[M89] Introduction to Algorithms: A Creative Approach, Manber, U., Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989.

[SAW *et al*. 2003] Use of Gene Networks from Full Genome Microarray Libraries to Identify Functionally Relevant Drug-affected Genes and Gene Regulation Cascades, Savoie, C., Aburatani, S., Watanabe, S., Eguchi, Y., Muta, S., Imoto, S., Miyano, S., Kuhara, S. and Tashiro, K., DNA Research, Vol. 10, 2003, pp. 19-25.

[W2001] How to Reconstruct a Large Genetic Network from n Gene Perturbations in fewer than n^2 Easy Steps, Wagner, A., Bioinformatics, Vol. 17, No. 12, 2001, pp. 1183-1197.

[W2002] Estimating Coarse Gene Network Structure from Large-Scale Gene Perturbation Data, Wagner, A., Genome Research, Vol. 12, 2002, pp. 309-315.

[WW2000] A Fuzzy Logic Approach to Analyzing Gene Expression Data, Woolf, P. and Wang, Y., Physiol Genomics, Vol. 3, 2000, pp. 9-15.

[YBD98] Genomic Cis-Regulatory Logic: Experimental and Computaional Analysis of a Sea Urchin Gene, Yuh, C. H., Bolouri, H. and Davidson, E. H., Science, Vol. 279, 1998, pp. 1896-1902.

[YLQG2003] Genomic Analysis of Gene Expression Relationship in Transcriptional Regulatory Networks, Yu, H., Luscombe, N. M., Qian, J. and Gerstein, M., TRENDS in Genetics, Vol. 19, No. 8, August 2003, pp. 422-427.

Appendix 3.1

Algorithm 3.1: State Determining Algorithm (Pseudo code)

Input: (1) A Boolean gene regulatory network $G(V, E)$, where V is the set of all the nodes which represent genes, E is the set of all the edges between the nodes of G. AND gate is added to the jointly incoming edges of a gene
(2) The set U of key regulators where states of all the key regulators of U are known

Output: The states of all the genes in G

```
Begin
1.      For each  u ∈ U ,
2.          Do Assign(u);
3.      End
4.      For each  g ∈ V
5.          mark(g) ← 0;
6.      End
7.      Procedure Assign(x)
8.      Begin
9.          mark(x) ← 1; /* mark x as visited */
10.         For all edges e_xy
11.         Do
12.             If mark(y) = 0
13.                 If  ¬(ψ(e_xy) ⊕ φ(x)) = 1, then
14.                     φ(y) ← 1;   Do Assign(y);
15.                 Else
16.                     φ(y) ← 0;   Do Assign(y);
17.                 End /* Corresponding to line 10 */
18.             Else   /* mark(y) = 1 */
19.                 If  φ(y) = 1
20.                     If  ¬(ψ(e_xy) ⊕ φ(x)) = 1
21.                         Then  φ(y) ← 1;   Do Assign(y);
22.                     Else
23.                         φ(y) ← 0;   Do Assign(y);
24.                     End /* Corresponding to line 17 */
25.                 Else   /*  φ(y) ← 0 */
```

26. Do Assign(y);
27. End /* Corresponding to line 16 * /
28. For $g \in V$ /* reset the marks but labels remains their value*/
29. mark(g) = ← 0;
30. End /* Corresponding to line 10 */
31. End Procedure Assign(x);

End of Algorithm 3.1

Here we denote $\phi(x)$ to be the state of x. $\phi(x) = 1$ if x is activated, and $\phi(x) = 0$ if x is inhibited. e_{xy} means the edge from x to y. $\psi(e_{xy}) = 1$ if e_{xy} is an activating edge, that is, $x \xrightarrow{\oplus} y$, and $\psi(e_{xy}) = 0$ if e_{xy} is an inhibiting edge, that is, $x \xrightarrow{\ominus} y$.

This algorithm is slightly different from the Depth-First-Search method. The spirit of this algorithm is that the state of each gene is determined by the effect of the genes which directly affect it. Assume that if a gene A's state is 1 and the edge from A to B is an activating edge, gene A will cause an activating effect on B. We can formulize the above case by $\neg(\phi(A) \oplus \psi(e_{AB})) = \neg(1 \oplus 1) = \neg 0 = 1$, where $\oplus$ denotes a logical connective, exclusive-or. Similarly, if a gene A's state is 0 and the edge from A to B is an activating edge, we can formulize the case by $\neg(\phi(A) \oplus \psi(e_{AB})) = \neg(0 \oplus 1) = \neg 1 = 0$, that is, gene A will cause an inhibiting effect on B. Therefore, we obtain four cases of effects that A causes on B. While at least one inhibiting effect that a gene A causes on gene B happens, gene B won't be activated.

Appendix 3.2

Lemma 3.1

A Boolean gene regulatory network which is free of cycles and free of self loops has at lease one node whose indegree, that is, the number of other genes that inhibits or activates it directly, is equal to 0.

Proof. Let G be a Boolean gene regulatory network with n nodes $g_1, g_2, \ldots, g_n$. Assume that there is no node in G whose indegree is equal to 0. Since every node's indegree is larger than 0 and G is acyclic, without loss of generality, we assume that g_1 at least affects g_2 in order to fit the condition that g_2's indegree is not 0, g_2 at least affects g_3 in order to fit the condition that g_3's indegree is not 0, …, g_{n-1} at least affects g_n in order to fit the condition that g_n's indegree is not 0. g_i is not allowed to affect g_1, $g_2, \ldots, g_{i-1}$, otherwise, a cycle $g_j, e_{(j)(j+1)}, g_{j+1}, \ldots, e_{(i-1)(i)}, g_i, e_{ij}, g_j$ will be produced and this contradicts to that G is acyclic. At last, no node will affect g_1 so that g_1's indegree will be 0. This contradicts our assumption. Therefore, there is at least one gene whose indegree is 0 in a Boolean gene regulatory network. **Q. E. D.**

Lemma 3.1 shows that every Boolean gene regulatory network has at least one key regulator. Next let us continue to see Theorem 3.1 which guarantees that if the states of all the key regulators in a Boolean gene regulatory network are determined at the first stage, the states of all the nodes in the Boolean gene regulatory network can all be determined.

Theorem 3.1

Assume that a Boolean gene regulatory network G and the states of all key regulators in G are given, then the states of all the nodes G can be all determined.

Proof. Since key regulators are the nodes whose indegree is 0, the other nodes' indegree must be larger than 1 or equal to 1 in the Boolean gene regulatory network *G*. Assume that there is a node, we name it g_1 W. L. O. G., which is not a key regulator and the state of g_1 can't be determined. Since g_1's indegree must be larger than 1 or equal to 1 because g_1 is not a key regulator, g_1 must be affected by at least one node, W. L. O. G. we name one of the nodes affecting g_1 g_2 and the state of g_2 must not be determined. Similarly, g_2 must be affected by at least one node g_3, except for g_1, and the state of g_3 must not be determined; …; g_i must be affected by at least one node, named g_{i+1} here, except for g_1, g_2, …, g_{i-1}, and the state of g_{i+1} must not be determined. The reason that g_i won't be affected by g_1, g_2, …, g_{i-1}, is that if g_i is affected by some g_k, $1 \le k \le i-1$, a cycle g_k, e_{ki}, g_i, $e_{(i-1)(i-2)}$, g_{i-2}, …, g_{k+1}, $e_{(k+1)(k)}$, g_k will be produced. Since the number of nodes in the *G* is finite, g_i must be affected by some key regulator whose state must not be determined. A contradiction occurs since the state of each key regulator has been assigned. Therefore, the assumption is false and we know that if the state of each gene can be determined through the key regulators in the Boolean gene regulatory network. **Q. E. D.**