國立臺灣大學電機資訊學院資訊工程學系

博士論文

Department of Computer Science and Information Engineering
College of Electrical Engineering and Computer Science
National Taiwan University

Doctoral Dissertation

有向圖與阿貝爾群的性質測試

Property Testing on Directed Graphs and Abelian Groups

狄彥吾

Yen-Wu Ti

指導教授：呂育道 博士

Advisor: Yuh-Dauh Lyuu, Ph.D.

中華民國 98 年 6 月

June, 2009

致 謝

首先誠摯的感謝指導教授呂育道博士，老師悉心的教導使我得以一窺資訊科學領域的深奧，不時的討論並指點我正確的方向，使我在這些年中獲益匪淺。老師對學問的嚴謹更是我輩學習的典範。

七年裡的日子，實驗室裡共同的生活點滴，學術上的討論、言不及義的閒扯...，感謝眾位學長、同學、學弟妹的共同砥礪，你/妳們的陪伴讓七年的研究生活變得絢麗多彩。

感謝周立平、戴天時學長們不厭其煩的指出我研究中的缺失，且總能在我迷惘時為我解惑，也感謝同學袁勤國、陳俊諺、張文彥、林敏順的幫忙，恭喜我們順利走過這些年。實驗室的張中平、馬德文、張經略、林宏佑、陳絢昌學弟們當然也不能忘記，各位的幫忙我銘感五內。

最後，謹以此論文獻給我已經離世的雙親。

摘 要

性質測試 (property testing)是一個在資訊學科中被廣泛研究的課題，其應用範圍涵蓋了網路拓樸與程式除錯等多個領域。Bender 與 Ron 發展了一個性質測試的演算法，用來測試某些有向圖是否滿足強連通的性質，本論文將這個演算法稱之為 BR 演算法。BR 演算法只適用於滿足特定條件的某些有向圖，是以在應用上有諸多限制。本論文發展了一個不受限制的演算法，可以測試任何一個有向圖是否具有強連通的性質。

相對於在應用上受到限制的 BR 演算法，本論文所發展用來測試強連通性質的演算法，在應用上雖然沒有限制，但是相較於 BR 演算法之效率較差。因此對於滿足 BR 演算法應用限制的少數有向圖，我們還是傾向用 BR 演算法來測試其是否有強連通的性質；至於其他不滿足限制的有向圖，我們便使用本論文發展的演算法。本論文接著發展一個演算法可以幫助我們在上述兩者間選擇合適的強連通測試演算法。

對於一個事先設定的有向圖 H，Alon 與 Shapira 證明了對於任一個有向圖 G，如果我們需要大量移除 G 的有向邊，才可以完全消除 G 裡面與 H 共構(isomorphic)的子圖(subgraph)，則在有向圖 G 中的 H 共構子圖的數目有一個下界。對於一個有向子圖，如果其圖形的任一部分都不與 H 形成共構，我們便稱之為無 H 子圖。本論文利用 Alon 與 Shapira 的研究結果，發展了一個演算法，用以測試任一個有向圖是否存在一個由 k 個點所組成的無 H 子圖。如前所述，當我們需要在兩個強連通測試演算法之間做選擇的時候，這個測試無 H 子圖的演算法就可以幫助我們選擇合適的強連通測試演算法。

本論文的最後一部分是利用本論文之前的強連通測試演算法，去發展一個關於群性質的測試演算法。一個群的生成元可以用來生成整個群，而對於一個阿貝爾群而言，其生成元的數目是一個倍受學界關注的研究題目。本論文利用前述的強連通測試演算法，發展了一個很有效率的演算法，可以測試任一個集合與一個二元運算的組合是否為一個生成元數目小於 k 的阿貝爾群。

關鍵詞：性質測試，隨機演算法，有向圖，強連通，阿貝爾群，生成元。

Abstract

Bender and Ron construct a restricted tester on the strong connectivity of digraphs (we call it the BR tester). We generalize the BR tester to test the strong connectivity of digraphs.

For any digraph H and a digraph G being far from any H-free digraph, Alon and Shapira prove a lower bound of the number of H in G. After solving the problem of testing the strong connectivity of digraphs, we use Alon and Shapira's result to construct a randomized algorithm for testing digraphs with an H-free k-induced subgraph.

Our strong connectivity tester has no restriction but must query about the input more times than the restricted BR tester. Suppose an input digraph satisfies the restrictions of the BR tester, using the BR tester to test the strong connectivity of this input digraph is more efficient than using our strong connectivity tester. If we want to test the strong connectivity of a digraph, our randomized algorithm for testing digraphs with an H-free k-induced subgraph can help us determine which tester should be used to test the strong connectivity of the digraph: the BR tester or our strong connectivity tester.

A generator set for a finite group is a subset of the group elements such that repeated multiplications of the generators alone can produce all the group elements. The number of generators of an abelian group is an important issue in many studies.

In most cases, it is not easy to identify whether a group-like structure is an abelian group with k generators for a constant k. We construct an efficient randomized algorithm that, given a finite set with a binary operation, tests if it is an abelian group with a k-generator set. If k is not too large, the query complexity of our algorithm is polylogarithmic in the size of the groundset. Otherwise the query complexity is at most the square root of the size of the groundset.

Keywords: Property testing; Strong connectivity; H-free subgraph; Abelian group; Generator.

Contents

Chapter 1

Introduction

This world is full of decision problems, and we need to make decisions every day. In computer science, a decision problem asks if an object has a predetermined property. Unfortunately, sometimes no fast algorithms exist that give the exact answer. In these cases, an approximate answer within a reasonable complexity is an attractive alternative.

A property-testing algorithm offers such answers. For a fixed property P and any object O, the property-testing algorithm determine whether O has property P, or whether O is far from having property P (i.e., far from any other object having P). It is, however, arbitrary on objects falling between the two categories. For example, the object can be a graph and the property can be 3-colorabilty. The task should be performed by querying the object (in as few places as possible). In the example, what we query is the existence of edges between two vertices.

Many recent research results concern the testing of graph properties and group properties. In computer science, the general notion of property testing is first formu-

lated by Rubinfeld and Sudan [64]. In their formulation, a property testing algorithm for property P is given oracle access to the tested object. Distance between instances is measured in terms of the fraction of arguments in the domain.

Property testing emerges naturally in the context of program checking and probabilistically checkable proofs (PCP). Specifically, in the context of program checking, one may choose to test if the program satisfies certain properties before checking that it computes a specified function. This paradigm has been followed both in the theory of program checking [22, 64], and in practice where often programmers first test their programs by verifying that the programs satisfy properties that are known to be satisfied by the functions they compute. In the context of probabilistically checkable proofs, the property tested is being a codeword with respect to a specific code. This paradigm, explicitly introduced in Babai, Fortnow, Levin and Szegedy's result, has shifted to testing Hadamard codes, and then to testing the long code [10, 13, 15, 16, 43, 44, 54, 68]. All of these papers have focused on property testing of algebraic properties such as linearity, multi-linearity and being a low-degree polynomial.

The number of generators of a group is an important issue in many studies. Knowing the number of generators of a group leads to a deeper understanding to the structure of a group. It may help us to discover the features of the groups quickly. A group in which every element commutes with its endomorphic images is called an E-group. A generator set with size k is called a k-generator set and a group for which the elements commute is called an abelian group. We know that every E-group with

a 2-generator set is abelian and all E-groups with a 3-generator set are nilpotent of class at most 2 [2]. We also know that we need at least four generators to generate a finite non-abelian E-group [1]. A group with a 2-generator set must isomorphic to a proper factor group [61]. The number of generators of a group has also been intensively studied [24, 25, 26, 48, 55, 58, 59].

This dissertation presents a method to combine the testing algorithms of digraphs and groups to test whether a group-like structure is an abelian group with a k-generator set. The first part of this dissertation is testing whether a digraph is strongly connected. Bender and Ron construct a restricted tester on the strong connectivity of digraphs (we call it the BR tester) [17]. There are some instances that do not satisfy the restrictions of the BR tester. We generalize the BR tester to test the strong connectivity of digraphs.

For any digraph H and a digraph G being far from any H-free digraph, Alon and Shapira prove a lower bound of the number of H in G. After solving the problem of testing the strong connectivity of digraphs, we use Alon and Shapira's result to construct a randomized algorithm for testing digraphs with an H-free k-induced subgraph.

Our strong connectivity tester has no restriction but must query about the input more times than the restricted BR tester. Suppose an input digraph satisfies the restrictions of the BR tester, using the BR tester to test the strong connectivity of this input digraph is more efficient than using our strong connectivity tester. If we want to test the strong connectivity of a digraph, our randomized algorithm for

testing digraphs with an H-free k-induced subgraph can help us determine which tester should be used to test the strong connectivity of the digraph: the BR tester or our strong connectivity tester.

It is not easy to identify whether a group-like structure is an abelian group with a k-generator set for any given constant k. In the last part of this dissertation, we combine the testing algorithm for the abelian property of groups and the testing algorithm for the strong connectivity of digraphs to form the testing algorithm for a group-like structure being an abelian group with a k-generator set. Our method is to use the strong connectivity of Cayley graphs to test if a finite group-like structure has a k-generator set. Before that, we should test if the input group-like structure is an abelian group. Friedl, Ivanyos and Santha construct a tester which, given a finite group-like structure, tests if it is an abelian group (we call it the FIS tester) [37]. Combining the FIS tester and our strong connectivity tester, we can construct a testing algorithm for a group-like structure being an abelian group with a k-generator set.

Chapter 2

Background

2.1 Question of property testing

We are interested in the following question of property testing:

> Let Π be a fixed property, and t be an instance. Our goal is to determine (possibly probabilistically) if t has property Π or if it is far from any instance that has property Π, where the distance between instances is measured with respect to the uniform probability distribution on the domain of t. Towards this goal, we are allowed to select some elements from t and query a specific information about t on elements of our choice.

Let T be the class of instances that satisfy property Π. Then, testing property Π corresponds to testing membership in the class T. The two most relevant parameters to property testing are the distance, hereafter denoted ϵ, and the desired confidence, denoted p. We require the tester to accept each instance in T and reject every instance that is more than ϵ away from any instance in T. We allow the tester to be

probabilistic, and make incorrect positive and negative assertions with probability at most p. The complexity measures we focus on are the query complexity (the number of queries made). We believe that property testing is a natural notion whose relevance to applications goes beyond program checking, and whose scope goes beyond the realm of testing algebraic properties.

2.2 Property testing on combinatorial objects

Working within the above framework, we venture into the domain of combinatorial objects. In particular, we study testing group properties and graph properties, and demonstrate its relevance to the notions of approximation. We hope to derive extremely efficient algorithms for testing natural properties.

We only consider the uniform probability distribution on these combinatorial objects, as well as algorithms that only obtain random samples.

2.3 Property testing and learning theory

Our formulation of testing mimics the standard frameworks of learning theory. Suppose the property Π is a set of functions. In both property testing and learning theory, one is given access to an unknown target function f. However, there are two important differences between them. First, the goal of a learning algorithm is to find a good approximation to the target function $f \in \Pi$, whereas a testing algorithm should only determine whether the target function is in Π or far away from it. This makes the task of testing seem easier than that of learning. But that is misleading

because a learning algorithm should perform well only when the target function belongs to Π, whereas a testing algorithm must perform well in such cases as well as on functions far away from Π.

Goldreich, Goldwasser and Ron show that the relation between learning and testing is nontrivial. On one hand, proper learning (i.e., when the hypothesis of the learning algorithm must belong to the same class as the target function) implies testing. On the other hand, there are function classes for which testing is harder than (nonproper) learning (i.e., when the hypothesis is not required to belong to the same class as the target function), provided NP $\not\subseteq$ BPP [41].

Chapter 3

Testing of Digraph Properties

3.1 Property testing on digraphs

We define property testing for digraphs next. Let Π be a property of digraphs, that is, a family of digraphs closed under isomorphism. A digraph $G = (V, E)$ is ϵ-close to having property Π if there exists a digraph $G' = (V, E')$ having property Π such that the symmetric difference between E and E' is at most $\epsilon\binom{|V|}{2}$. We say that a digraph G is ϵ-far from having property Π if it is not ϵ-close to having property Π.

An ϵ-tester (or simply a tester) for a digraph property Π is a randomized algorithm that is given a size parameter N, a distance parameter ϵ and the ability to make queries as to whether a directed edge of the input digraph G with N vertices exists. The total number of queries is called the query complexity of the tester. Let $\{g_i\}$ be the set of digraphs with N vertices that satisfy Π. The algorithm needs to distinguish with probability at least 2/3 between the case of G having Π and the case of G differing from any g_i in at least $\epsilon\binom{N}{2}$ edges [7]. In the latter case, G is

said to be ϵ-far from property Π. More specifically, T is an ϵ-tester for Π if for every $G = (V, E)$ and every ϵ, the following two conditions hold:

(1) if G has property Π, then $\Pr[T \text{ accepts } G] \geq 2/3$;

(2) if G is ϵ-far from having property Π, then $\Pr[T \text{ accepts } G] \leq 1/3$.

The probability 2/3 can be replaced by any constant smaller than 1 because the algorithm can be repeated if necessary. A graph property is testable if the property has a tester and the total number of queries is $o(N^2)$.

3.2 Research work related to graph property testing

A testing algorithm for graph property Π can make queries on the incidence relations of vertices in an input graph G. Property Π is $\Omega(N^2)$-evasive if there is no deterministic testing algorithm with query complexity $o(N^2)$ that can correctly decide if the input has Π. Holt and Reingold are the first to consider the complexity of recognizing graph properties from their adjacency matrix representations [47]. They show that the graph properties of connectivity and the existence of cycles are both $\Omega(N^2)$-evasive. An important open problem in this area is the Aanderaa-Rosenberg conjecture: Every nontrivial monotone graph property without self-loops is $\binom{N}{2}$-evasive [23, 29, 47, 53]. Rivest and Vuillemin resolve a weaker version of the Aandreaa-Rosenberg conjecture [63]. The weaker version says that every nontrivial monotone graph property has decision tree complexity $\Omega(N^2)$.

The testing of graph properties is pioneered by Goldreich, Goldwasser and Ron [41]. They showe that all graph properties describable by the existence of a partition of a certain type are testable. For a fixed digraph H with at least one edge, let P_H denote the property of the input digraph being H-free. In other words, the digraph G has P_H if and only if it contains no subgraphs isomorphic to H. Alon and Shapira prove that P_H is testable with a total number of queries bounded only by a function of ϵ, independent of N [7]. This result has been improved later by Alon and Shapira [6]. Alon, Fischer, Krivelevich and Szegedy show that every first-order undirected graph property without a quantifier alternation of type "$\forall$ $\exists$" has ϵ-testers whose query complexity is independent of the size of the input digraph [4]. More recently, Alon, Fischer, Newman and Shapira prove a very general result for undirected graphs, which says that the property defined by having any given Szemerédi-partition is testable with a constant number of queries [5]. Moreover, a purely combinatorial characterization of the graph properties is testable with a constant number of queries. The testing of other graph and combinatorial properties has also been intensively studied [30, 33, 40, 42, 51].

3.3 Reduction between group properties and digraph properties

In this section, we define Cayley graphs and introduce the reduction between group properties and digraph properties.

Definition 1 ([46]) *Let G be a group, $\circ$ be the group multiplication and S be a*

subset of the group's elements not containing the identity element. The Cayley graph associated with S is defined as the digraph having one vertex associated with each group element G and directed edges (g, h) whenever $g \circ h^{-1} \in S$.

The properties of Cayley graphs have been extensively studied in graph theory. These properties are used to develop algebraic settings for studying certain structural and algorithmic properties of the interconnection networks that underlie parallel architectures, including the hypercube, butterfly, cube-connected cycles, multiple rings and star networks [3, 8, 21, 27, 65]. Cayley graphs have also been used to study the gossiping problem in communication networks [20].

We say that a digraph is strongly connected if there is a directed path from every vertex u in the digraph to every other vertex v.

Theorem 2 ([9]) *The Cayley graph associated with a subset of a group's elements (but not containing the identity element) is strongly connected iff the subset generates the group.*

Our method relies on the strong connectivity of Cayley graphs to test if a finite group-like structure $s = (\Gamma, \circ, i, 1)$ has a k-generator set. A subset of a groundset or a vertex set with size k will be called a k-subset from now on. By Theorem 2, for an input group-like structure $s = (\Gamma, \circ, i, 1)$, if we can test whether there exists a k-subset of Γ with a corresponding strongly connected Cayley graph, then we can test whether s has a k-generator set. In the next chapter, we develop an algorithm for testing strong connectivity of digraphs.

Chapter 4

Testing Strong Connectivity on Digraphs

4.1 Strongly connected component

In this chapter, we develop an algorithm for testing strong connectivity of digraphs and we will rely on the strong connectivity of Cayley graphs to test if a finite group-like structure $s = (\Gamma, \circ, i, 1)$ has a k-generator set in the following chapter. For a digraph $G = (V, E)$ with indegree and outdegree bounded by $d < |V|$, Bender and Ron develop a tester on the strong connectivity of digraphs (we call it the BR tester) [17].

Theorem 3 ([17]) *(1) If G is ϵ-far from being strongly connected with indegree and outdegree bounded by d, then the BR tester will reject it with probability at least $2/3$. (2) The BR tester has one-sided error. (3) The query complexity is $O(1/\epsilon)$.*

On the other hand, suppose there is no bound on the indegree and outdegree of the digraph G. We construct another tester to test the strong connectivity of

digraphs in this chapter that is slightly different from the BR tester. To begin with, testing strong connectivity is trivial when the distance parameter is greater than $2/n$ for the following reason: We can always make a digraph connected by adding at most $n-1$ edges. Hence every digraph with n vertices is $(2/n)$-close to being strongly connected because $(2/n)\binom{n}{2} = n-1$. On the other hand, ϵ should be greater than $1/\binom{n}{2}$ for, otherwise, $\epsilon\binom{n}{2} < 1$. To make sure that the problem is not trivial, we assume $1/\binom{n}{2} < \epsilon < 2/n$ from now on.

A strongly connected component of a digraph $G = (V, E)$ is a maximal subgraph $H = (V', E')$ such that there is a directed path from each vertex $p \in V'$ to every other vertex $q \in V'$. Denote the set of strongly connected components of G by $C = \{C_1, C_2, \ldots, C_m\}$. For vertices $u \in C_i$ and $v \in C_j$, $i \neq j$, such that $e = (u, v) \in E$, we call e an outgoing edge of C_i and an incoming edge of C_j. We call a strongly connected component a source if it has only outgoing edges, a sink if it has only incoming edges, an isolation if it has neither outgoing nor incoming edges, and a transferrer if it has both outgoing and incoming edges.

Lemma 4 *If a digraph G with n vertices is ϵ-far from being strongly connected, then the total number of sources, sinks, and isolations in G exceeds ϵn^2.*

Proof: Assume the claim of the lemma is wrong and proceed to obtain a contradiction. Let set $\mathcal{T}$ contain all transferrers and set $\mathcal{R}$ consisting of the remaining strongly connected components. We divide the problem up into two cases.

Case 1. $\mathcal{T}$ is empty.

A strongly connected component in G is either a source, a sink, or an isola-

tion, and hence a member of $\mathcal{R}$. Pick a vertex from each strongly connected component of G. As G has $|\mathcal{R}|$ strongly connected components, $|\mathcal{R}|$ vertices are chosen. Turn these $|\mathcal{R}|$ vertices into a directed cycle by adding at most $|\mathcal{R}|$ directed edges to G. Now, for any ordered pair of vertices (u, v) in V, there is a directed path from u to v; hence G is strongly connected. Recall that an ϵ-far digraph differs from the class of strongly connected digraphs by at least $\epsilon n^2 + 1$ edges. However this is a contradiction, since $|R| \leq \epsilon n^2$. See Fig. 4.1 for illustration.

Case 2. $\mathcal{T}$ is not empty.

A chain of strongly connected components $(C_1, C_2, \ldots, C_n)$ consists of strongly connected components such that between any two adjacent strongly connected components C_i and C_{i+1}, there is a directed edge from a vertex of C_i to a vertex of C_{i+1}.

For any $T \in \mathcal{T}$, there is a longest chain of strongly connected components S in $\mathcal{T}$ containing T (i.e., all members of S are transferrers, and T is one of them). Let S^h be the head of the chain S. S^h is a strongly connected component with both incoming and outgoing edges because $S^h \in \mathcal{T}$. There is no directed edge from other strongly connected components in $\mathcal{T}$ to S^h for, otherwise, S would not be the longest chain containing T. Therefore, all the incoming edges of S^h are outgoing edges of some sources. By the same token, all the outgoing edges of the tail S^t of S are incoming edges of some sinks. Consequently, for any $T \in \mathcal{T}$, we can always find a vertex p from a source and a vertex q

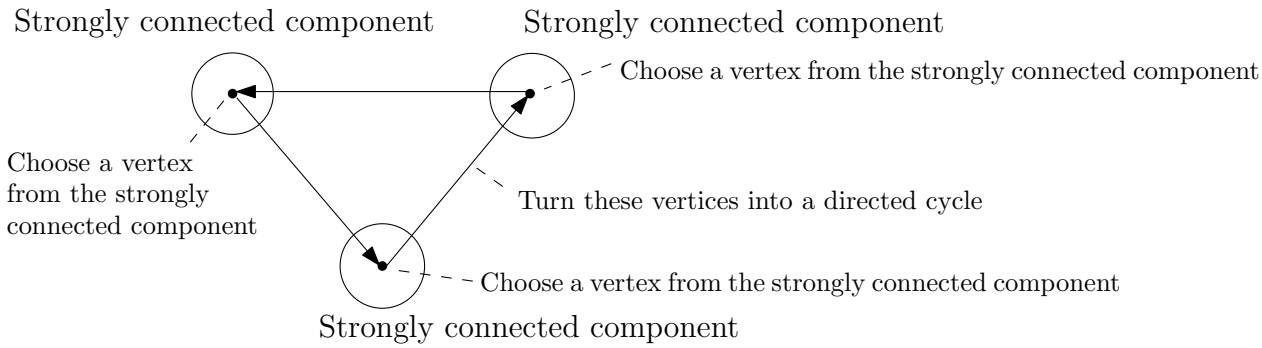

Figure 4.1: Testing strong connectivity of digraphs.

from a sink such that there is a directed path from p to q passing through a vertex of T. As a result, if all the sources and sinks are turned into one single strongly connected component, the transferrers will become part of the same component, too.

Therefore, if we can turn all members of $\mathcal{R}$ into one single strongly connected component $\mathcal{R}'$, then for all $T_i, T_j \in \mathcal{T}$ and all vertices $x \in T_i$ and $y \in T_j$, there exists a directed path from x through some vertices in $\mathcal{R}'$ to y. So we can ignore the transferrers and concentrate on how to make all members of $\mathcal{R}$ become one single strongly connected subgraph. We have thus reduced this case to case 1, where $\mathcal{T}$ is empty. The number of directed edges we need to add to G is thus at most $|\mathcal{R}|$, too. If $|R|$ is no greater than ϵn^2, we get a contradiction. **Q.E.D.**

Lemma 5 *A digraph ϵ-far from being strongly connected must have at least $(\epsilon n^2)/2$ connected components each containing fewer than $\lceil \frac{1}{n}(\frac{2}{\epsilon} - 1) \rceil$ vertices.*

Proof: Assume the claim is wrong. Lemma 4 says that this digraph has at least $\epsilon n^2 + 1$ strongly connected components. Since every strongly connected component has at least one vertex,

$$\begin{aligned} n &\geq \left(\epsilon n^2 + 1 - (\epsilon n^2)/2\right) \cdot \left\lceil \frac{1}{n}\left(\frac{2}{\epsilon} - 1\right)\right\rceil + \frac{\epsilon n^2}{2} \cdot 1 \\ &> \frac{\epsilon n^2}{2}\frac{1}{n}\left(\frac{2}{\epsilon} - 1\right) + \frac{\epsilon n^2}{2} \\ &= \frac{\epsilon n}{2}\left(\frac{2}{\epsilon} - 1\right) + \frac{\epsilon n^2}{2} \\ &= n - \frac{\epsilon n}{2} + \frac{\epsilon n^2}{2} \end{aligned}$$

a contradiction. **Q.E.D.**

4.2 Tester construction

For a digraph with no bound on the indegree and outdegree of each vertex, the algorithm that tests whether it is strongly connected appears in Fig. 4.2 (we call it the CONN tester). It is obviously that testing strong connectivity is trivial when the distance parameter is greater than $2/n$. To make sure that the problem is not trivial, we assume $1/\binom{n}{2} < \epsilon < 2/n$. The following theorem analyzes the algorithm.

Theorem 6 *(1) Our algorithm has one-sided error. (2) Suppose G is ϵ-far from being strongly connected, then our algorithm will reject it with probability at least $2/3$. (3) The query complexity is $O\left(\frac{\log_{1-\frac{\epsilon n}{2}} 1/3}{\epsilon}\right)$.*

Let $S = \emptyset$, $m = \lceil \log_{1-(\epsilon n)/2} 1/3 \rceil$, $x = \lceil \frac{1}{n}(\frac{2}{\epsilon} - 1) \rceil$
while $\mid S \mid < m$ **do**
  Pick an arbitrary vertex u from V and add it to S
  Perform BFS on G starting from u and always use a visited vertex's incoming edges and stop when we reach x vertices
  if we run out of new vertices **then**
    REJECT
  else
    Perform BFS starting from u and always use a visited vertex's outgoing edges and stop when we reach x vertices
    if we run out of new vertices **then**
      REJECT
    end if
  end if
end while
ACCEPT

Figure 4.2: An ϵ-tester for the strong connectivity of digraphs in the case that there is no prior bound on the indegree and outdegree of each vertex.

Proof: Since $1/\binom{n}{2} < \epsilon < 2/n$, $x \leq n$. For a strongly connected digraph G, given an arbitrary vertex u of V, a BFS always reaches at least x vertices by starting from u. In other words, our algorithm never rejects G, a yes instance; hence it has only one-sided error.

For a no instance G, according to Lemma 5, there are at least $(\epsilon n^2)/2$ strongly connected components, each containing fewer than x vertices. The probability that an arbitrarily chosen vertex belongs to one of these $(\epsilon n^2)/2$ strongly connected components is at least $\frac{(\epsilon n^2)/2}{n} = (\epsilon n)/2$. Our algorithm outputs ACCEPT for a no instance only when the while-loop executes m times. The probability is at most $[1-(\epsilon n)/2]^m = [1-(\epsilon n)/2]^{\lceil \log_{1-(\epsilon n)/2} \frac{1}{3} \rceil} \leq 1/3$. Hence, G is rejected with probability at least $2/3$.

Finding all of a vertex's incoming (outgoing) neighbors takes at most n queries on the incoming (outgoing, respectively) edges given the adjacency matrix. Since the while-loop is repeated at most m times, the query complexity is bounded by:

$$m \cdot 2 \cdot \left\lceil \frac{1}{n} \left(\frac{2}{\epsilon} - 1 \right) \right\rceil \cdot n < \left\lceil \log_{1-(\epsilon n)/2} \frac{1}{3} + 1 \right\rceil \cdot \left\lceil \frac{2}{n} \left(\frac{2}{\epsilon} - 1 \right) \right\rceil \cdot n$$
$$= O\left(\frac{\log_{1-(\epsilon n)/2} \frac{1}{3}}{\epsilon} \right). \qquad \textbf{Q.E.D.}$$

According to Theorem 6, the query complexity of the CONN tester is much higher than the BR tester. Suppose the indegree and the outdegree of an input digraph are both bounded by a given constant, we can use the BR tester to test the strong connectivity of the input in order to reduce the query complexity; otherwise, we use the CONN tester. In the next chapter, we investigate the method to test whether a

digraph contains H-free k-induced subgraphs. If we do not know the upper bounds of the indegree and the outdegree of an input digraph, our result in the next chapter can help us determine what algorithm to use and then use that algorithm to test the strong connectivity of the input.

Chapter 5

Testing Whether a Digraph Contains H-free k-induced Subgraphs

5.1 Existence of H-free k-induced subgraphs is $\Omega(N^2)$-evasive

In this section, we show that the query complexity of any deterministic algorithm for the existence of H-free k-induced subgraphs is $\Omega(N^2)$.

First, we need some results concerning Turán numbers. For any integer N and a fixed graph H, let $\mathrm{ex}(N, H)$ denote the maximum number of edges that an N-vertex graph may have if it contains no isomorphic copy of H. This is the Turán number of H. Furthermore, we will denote by $b_{r,s}$ the complete undirected bipartite graph between a set of r vertices and another set of s vertices. The following fact is well-known.

Fact 7 ([54]) *For $r \leq s$, $\mathrm{ex}(N, b_{r,s}) = O(N^{2-(1/r)})$.*

If we replace the undirected edges of $b_{r,s}$ by directed edges with an arbitrary direction, a complete bipartite digraph $d_{r,s}$ results. The next theorem shows that it is $\Omega(N^2)$-evasive to determine if there is a $d_{r,s}$-free k-induced subgraph in a digraph. In our model, whenever an algorithm queries a pair of vertices x, y in the input graph, it actually means that the algorithm queries the existence of edges (x, y) and (y, x) simultaneously. For a set S, we say that a subset $T \subseteq S$ is a k-subset of S if $|T| = k$. Suppose a digraph G contains a subgraph isomorphic to a digraph H. Then we say G contains a copy of H.

Theorem 8 *For any constant $\rho < 1$, $k < N/2$ and any complete bipartite digraph $d_{r,s}$, no algorithm can determine whether a digraph contains a $d_{r,s}$-free k-induced subgraph with query complexity $\leq \rho\binom{k}{2}$ if $k = \lambda N$ with λ being a constant.*

Proof: Suppose there exists an algorithm A that determines if a digraph contains a $d_{r,s}$-free k-induced subgraph with $\rho\binom{k}{2}$ queries. For the rest of the proof, assume $k = O(N)$ and k is large enough so that

$$(1-\rho)\binom{k}{2} \geq \mathrm{ex}(k, b_{r,s}) = O(k^{2-(1/r)}). \tag{1}$$

Start with a digraph G_1 with N vertices that contains no copies of $d_{r,s}$ (this is easy to construct). Let G_1 be the input of A. Obviously, all k-induced subgraphs of G_1 are $d_{r,s}$-free. Let $G_2 = (V_2, E_2)$ be a graph with N isolated vertices. Every time A queries a pair of vertices x, y in G_1, we add that edge to G_2 if there is an edge

between them. When A stops, the resulting G_2 has no k-induced subgraphs which contain $d_{r,s}$, just like G_1. For those vertex pairs of G_1 that are not queried by A, we add an edge (but without the directions) to G_2. For each k-induced subgraph of G_2, at least $(1-\rho)\binom{k}{2}$ undirected edges are added. According to Fact 7, every k-induced subgraph of G_2 must contain a copy of $b_{r,s}$ with the undirected edges alone because of Eq. (1).

Now, we select a k-induced subgraph K_1 in G_2 and replace one copy of $b_{r,s}$ in K_1 by $d_{r,s}$. Let $V_{b,1}$ be the vertex set of this copy of $d_{r,s}$, and define $h = |V_{b,1}| = r + s$. For each subset of V_2 with size k that contains $V_{b,1}$, its induced subgraph has a copy of $d_{r,s}$ too. There are $\binom{N-h}{k-h}$ such k-subsets of V that contain $V_{b,1}$. Let $k/N = \lambda$. Recall that λ is a constant. Now, the ratio of the number of all such k-subsets to the number of k-induced subgraphs of G_2 is $\binom{N-h}{k-h}/\binom{N}{k}$. Note that

$$\frac{\binom{N-h}{k-h}}{\binom{N}{k}} = \frac{k(k-1)\cdots(k-h+1)}{N(N-1)\cdots(N-h+1)}.$$

As h is a constant and $k < N/2$, it is not hard to prove that there is a number $m > 0$ such that for every $N > m$ it holds that

$$\begin{aligned}\frac{k}{N} > \frac{k-1}{N-1} > \cdots > \frac{k-h+2}{N-h+2} > \frac{k-h+1}{N-h+1} \\ = \frac{(k/N) - (h/N) + 1/N}{1-(h-1)/N} > \frac{\lambda}{1+\lambda}.\end{aligned}$$

Thus if N is large enough,

$$\frac{\binom{N-h}{k-h}}{\binom{N}{k}} = \frac{k(k-1)\cdots(k-h+1)}{N(N-1)\cdots(N-h+1)} > \left(\frac{\lambda}{1+\lambda}\right)^h.$$

We conclude that at least $\left(\frac{\lambda}{1+\lambda}\right)^h \binom{N}{k}$ k-induced subgraphs contain a copy of $d_{r,s}$.

Next we select another k-induced subgraph $K_2 = (V', E')$ with $V' \cap V_{b,1} = \emptyset$. It is worth noting that K_2 also has a copy of $b_{r,s}$, and the vertex set of $b_{r,s}$ is $V_{b,2}$. Like what we did before, we replace this copy of $b_{r,s}$ in K_2 by $d_{r,s}$. There are $\binom{N-2h}{k-h}$ such k-subsets of V that contain V_2. The ratio of the number of all such k-subsets to the number of k-induced subgraphs of G_2 is $\binom{N-2h}{k-h}/\binom{N}{k}$. Again, for N large enough,

$$\lim_{N\to\infty} \frac{\binom{N-2h}{k-h}}{\binom{N}{k}} = \lim_{N\to\infty} \frac{\binom{N-h}{k-h}}{\binom{N}{k}} > \left(\frac{\lambda}{1+\lambda}\right)^h.$$

We claim that in general, for every constant i,

$$\frac{\binom{N-ih}{k-h}}{\binom{N}{k}} = \frac{k(k-1)\cdots(k-h+1)}{N(N-1)\cdots(N-h+1)} \cdot \frac{(N-k)\cdots(N-k-(i-1)h+1)}{(N-h)\cdots(N-ih+1)}$$
$$> \left(\frac{\lambda}{1+\lambda}\right)^h. \tag{2}$$

To verify this, recall that as we showed before,

$$\frac{k(k-1)\cdots(k-h+1)}{N(N-1)\cdots(N-h+1)} > \left(\frac{\lambda}{1+\lambda}\right)^h.$$

As for

$$\frac{(N-k)\cdots(N-k-(i-1)h+1)}{(N-h)\cdots(N-ih+1)},$$

since

$$\frac{N-k}{N-h} > \frac{N-k-1}{N-h-1} > \cdots > \frac{N-k-(i-1)h+1}{N-ih+1}$$

we have

$$\frac{(N-k)(N-k-1)\cdots(N-k-(i-1)h+1)}{(N-h)(N-h-1)\cdots(N-ih+1)} > \left(\frac{N-k-(i-1)h+1}{N-ih+1}\right)^{(i-1)h}$$

Now, with $k = \lambda N$, it is easy to see that

$$\frac{N-k-(i-1)h+1}{N-ih+1} > \frac{N-k-(i-1)h+1}{N} > (1-2\lambda)$$

where the last inequality is due to $k > (i-1)h-1$. Hence, when we repeat the above process i times, at least

$$[(1-2\lambda)+(1-2\lambda)^2+\cdots+(1-2\lambda)^{(i-1)h}]\left(\frac{\lambda}{1+\lambda}\right)^h\binom{N}{k} \qquad (3)$$

k-induced subgraphs contain a copy of $d_{r,s}$. Recall that $k < N/2$. Hence $2\lambda < 1$ and formula (3) is less than $\frac{1}{2\lambda}\left(\frac{\lambda}{1+\lambda}\right)^h\binom{N}{k}$.

Since λ, h and $(\lambda/(1+\lambda))^{-h}$ are constants, we can repeat this process $2\lambda(\lambda/(1+\lambda))^{-h}$ times such that $V_{b,i} \cap V_{b,j} = \emptyset$ for $i \neq j$ and N large enough. After having repeated this process that many times, we select $2\lambda(\lambda/(1+\lambda))^{-h}h < N$ distinct vertices from V for N large enough, and, by Eq. (2), the ratio of the number of $K_{2\lambda(\lambda/(1+\lambda))^{-h}}$ to the number of all k-induced subgraphs of G_2 will be at least $2\lambda(\lambda/(1+\lambda))^{-h}$. The number of k-induced subgraphs that contain a copy of $d_{r,s}$ then is at least $2\lambda(\lambda/(1+\lambda))^{-h}\frac{(\lambda/(1+\lambda))^h}{2\lambda}\binom{N}{k} = \binom{N}{k}$. In other words, after we repeat this process $2\lambda(\lambda/(1+\lambda))^{-h}$ times and remove the remaining undirected edges, all k-induced

subgraphs of G_2 will have a copy of $d_{r,s}$. This digraph G_2 contains, therefore, no H-free k-induced subgraph. However, A cannot distinguish between G_1 and G_2 because we have only changed G_2's unqueried edges. So, A will accept G_2, which is a contradiction. **Q.E.D.**

5.2 Tester construction

Fix a digraph H with h vertices and $m \geq 1$ edges. Recall that $P_{k,H}$, where $k \geq h$, denotes the property that G contains an H-free k-induced subgraph. We will show that property $P_{k,H}$ is testable with a query complexity independent of the input size. A set with size n will be called an n-set, and a multiset with size n will be called an n-multiset. There is a function $f(\epsilon; H)$ with the following properties, which will be critical to our analysis later.

Theorem 9 ([7]) *Let H be a fixed digraph with h vertices and D be a digraph with N vertices. If at least ϵN^2 edges have to be removed from D to make it H-free, then D contains at least $f(\epsilon; H)N^h$ copies of H.*

The following corollary is immediate.

Corollary 10 *Let H be a fixed digraph with h vertices and m edges, D be a digraph with N vertices and $\sigma = \binom{\binom{h}{2}}{m}$. If at least ϵN^2 edges have to be removed from D to make it H-free, then D contains at least $f(\epsilon; H)N^h/\sigma$ h-sets whose induced subgraphs contain copies of H.*

Suppose the input N-vertex digraph $G = (V, E)$ is ϵ-far from having property $P_{k,H}$. Corollary 10 tells us that G must contain at least $f(\epsilon; H)N^h/\binom{\binom{h}{2}}{m}$ h-sets whose induced subgraphs contain copies of H. So to test property $P_{k,H}$ on G, our idea is to randomly select many h-sets from V. Suppose G contains an H-free k-induced subgraph, say (V_k, E_k). Then with enough h-sets from V, at least one of them is expected to be a subset of V_k with high probability. To verify if this is the case, we will check if an h-set S satisfies $S \subseteq V_k$ in 2 steps. First, we check the induced subgraph of S. When $S \subseteq V_k$, the induced subgraph of S contains no copies of H. If the induced subgraph of S contains no copies of H, we randomly add a number of other vertices to S (the number will be determined later) and check if there is a subset of S (with a size to be determined later) whose induced subgraph contains no copies of H. If $S \subseteq V_k$, we expect that S will pass both tests with high probability. Thus, G will be accepted by our algorithm with high probability. On the other hand, suppose G is ϵ-far from any digraph which has property $P_{k,H}$. Then we expect to find a copy of H in all the induced subgraphs of the above-mentioned h-sets S with high probability. Our algorithm is detailed in Fig. 5.1.

We shall need the Chernoff bound in later analysis.

Theorem 11 (**Chernoff bound**) *Let* $X = X_1 + X_2 + \cdots + X_n$ *be a sum of* n *independent random variables such that* $0 < Pr[\ X_i = 1\ ] < 1$ *holds for each* $i = 1, 2, ..., n$ *and* $\mu = E[X]$. *Then for any* $0 < \Delta < 1$,

$$\Pr[\ X < (1 - \Delta)\mu\ ] < e^{-\mu\Delta^2/2}$$

1: **if** $k < \sqrt{\epsilon}N$ **then**
2: **ACCEPT**
3: **end if**
4: let $\lambda = k/N$, $\kappa = \log_{1-\frac{(\sqrt{\epsilon})^h}{2}}(1/6)$, $\sigma = \binom{\binom{h}{2}}{m}$ and $\theta = \max\{\log_{\frac{6f(\epsilon;H)h!}{\sigma\lambda^2}}(2/3)^{1/\kappa}, 1\}$
5: **for** $i = 1$ to κ **do**
6: randomly select an h-set S from V
7: **if** the induced subgraph of S does not contain an H **then**
8: randomly select additional vertices $p = 6\theta h/\lambda$ times (with replacements) from $V - S$ (assume these p vertices to be $x_1, x_2, \ldots, x_p$) {note there are $\binom{p}{\theta h}$ (θh)-multisets in $\{x_1, x_2, \ldots, x_p\}$}
9: **for** $j = 1$ to $\binom{p}{\theta h}$ **do**
10: let S_j be the jth (θh)-multiset selected in step 8
11: **if** the induced subgraph of $S_j \cup S$ contains no copies of H **then**
12: **ACCEPT**
13: **end if**
14: **end for**
15: **end if**
16: **end for**
17: **REJECT**

Figure 5.1: The ϵ-tester for property $P_{k,H}$.

where e is the base of the natural logarithm.

Note that in property $P_{k,H}$, h is a constant. Hence $f(\epsilon; H)$ is a function in ϵ only. We assume that H is a fixed digraph with h vertices and m edges and recall that G is the input digraph with N vertices from now on.

Definition 12 *Let* $0 < \epsilon < 1$, $N, k \in \mathbb{N}$, $\lambda = k/N$, H *be a fixed digraph with* h *vertices,* m *be the number of edges in* H, $\sigma = \binom{h}{2}$, $\kappa = \log_{1-\frac{(\sqrt{\epsilon})^h}{2}}(1/6) = \Theta(1/\epsilon^{h/2})$, *and* $\theta = \max\{\log_{\frac{6f(\epsilon;H)h!}{\sigma\lambda^2}}(2/3)^{1/\kappa}, 1\} = \Theta(f(\epsilon; H))$ *when* $f(\epsilon; H)$ *is only dependent on* $1/\epsilon$. *If the value of* $f(\epsilon; H)$ *is large enough such that* $\left(\frac{f(\epsilon;H)h!}{\binom{h}{2}\lambda}\right)^{\theta} \geq (\lambda/6)^{\theta}(2/3)^{1/\kappa}$, *then we say* $f(\epsilon; H)$ *satisfies condition 1.*

Fact 13 ([7]) *For a connected* H, $f(\epsilon; H)$ *has a polynomial dependency on* $1/\epsilon$ *if and only if the core of* H *is either an oriented tree or a directed cycle of length 2.*

By Fact 13, $f(\epsilon; H)$ has a polynomial dependency on $1/\epsilon$ for many H. Since the value of $f(\epsilon; H)$ is independent of h and m and $\left(\frac{2}{3}\right)^{1/(\theta\kappa)} \leq 1$, assuming $f(\epsilon; H) = O((1/\epsilon)^j)$, we can find a smaller $\epsilon = O\left(\left[\frac{\binom{h}{2}}{\frac{m}{h!}}\frac{\lambda^2}{6}\left(\frac{2}{3}\right)^{1/(\theta\kappa)}\right]^{-j}\right)$ such that

$$f(\epsilon; H) \geq \frac{\binom{h}{2}}{\frac{m}{h!}}\frac{\lambda^2}{6}\left(\frac{2}{3}\right)^{1/(\theta\kappa)}$$

i.e.,

$$\left(\frac{f(\epsilon; H)h!}{\binom{h}{2}\lambda}\right)^{\theta} \geq (\lambda/6)^{\theta}(2/3)^{1/\kappa};$$

hence $f(\epsilon; H)$ satisfies condition 1.

Claim 14 *Assume* $0 < \epsilon < 1$, $N, k \in \mathbb{N}$ *and* $k \geq \sqrt{\epsilon}N$. *Suppose the input digraph* $G = (V, E)$ *with* N *vertices contains an* H*-free* k*-induced subgraph, say* $K = (V_k, E_k)$. *The probability of* $S \subseteq V_k$ *for a random* h*-subset* $S \subseteq V$ *is greater than* $(\sqrt{\epsilon})^h/2$ *for* N *large enough.*

Proof: The probability of $S \subseteq V_k$ for a random h-set S is

$$\frac{\binom{k}{h}}{\binom{N}{h}} = \frac{k(k-1)\cdots(k-h+1)}{N(N-1)\cdots(N-h+1)}.$$

Since $k \geq \sqrt{\epsilon}N$, the above probability is at least

$$\frac{\sqrt{\epsilon}N(\sqrt{\epsilon}N-1)\cdots(\sqrt{\epsilon}N-h+1)}{N(N-1)\cdots(N-h+1)} > \frac{(\sqrt{\epsilon})^h}{2}$$

for N large enough. **Q.E.D.**

Claim 15 *Let* $0 < \epsilon < 1$, $N, k \in \mathbb{N}$, $\lambda = k/N$, H *be a fixed digraph with* h *vertices and* m *be the number of edges in* H. *Suppose the input graph* $G = (V, E)$ *with* N *vertices is* ϵ*-far from any digraph having property* $P_{k,H}$. *The probability of finding an* h*-set whose induced subgraph contains copies of* H *is at least* $f(\epsilon; H)h!/\left[\binom{h}{2}\binom{m}{\lambda}\right]$.

Proof: By Corollary 10, each k-induced subgraph of G contains at least $f(\epsilon; H)N^h/\left[\binom{h}{2}\binom{m}{}\right]$ h-sets whose induced subgraphs contain copies of H. Therefore, by dividing V into N/k k-sets, we can find at least $\left[f(\epsilon; H)N^h/\binom{h}{2}\binom{m}{}\right](N/k) = f(\epsilon; H)N^h/\left[\binom{h}{2}\binom{m}{\lambda}\right]$ h-sets whose induced subgraphs contain copies of H in G, and the probability of

finding an h-set whose induced subgraph contains copies of H is at least

$$\frac{\frac{f(\epsilon;H)N^h}{\binom{\binom{h}{2}}{m}\lambda}}{\binom{N}{h}} = \frac{f(\epsilon;H)N^h \cdot \frac{1}{\lambda} \cdot h!}{N(N-1)\cdots(N-h+1)\binom{\binom{h}{2}}{m}}$$
$$> \frac{f(\epsilon;H)h!}{\binom{\binom{h}{2}}{m}\lambda}. \qquad \textbf{Q.E.D.}$$

The following theorem proves the testability of $P_{k,H}$.

Theorem 16 *Let $0 < \epsilon < 1$, $0 < k < N$ be an integer and H be a fixed digraph. If $f(\epsilon;H)$ satisfies condition 1, the property $P_{k,H}$ is testable with a query complexity independent of the input size.*

Proof: Suppose $k < \sqrt{\epsilon}N$. Then the number of edges in a k-induced subgraph is less than ϵN^2. The input graph G, therefore, cannot be ϵ-far from any digraph which has property $P_{k,H}$, and we can simply accept it. Assume $k \geq \sqrt{\epsilon}N$ for the rest of the proof.

Suppose the input digraph $G = (V, E)$ contains an H-free k-induced subgraph, say $K = (V_k, E_k)$. The probability that the algorithm accepts G is at least the probability of selecting a subset of V_k in step 6 of the algorithm in Fig. 5.1 and the tester accepts in step 12 for some j.

By Claim 14, the probability of $S \nsubseteq V_k$ is at most $1 - (\sqrt{\epsilon})^h/2$. As we independently select κ h-sets S, the probability of $S \nsubseteq V_k$ for all κ of them is at most $\left[1 - (\sqrt{\epsilon})^h/2\right]^\kappa = 1/6$. Assume $S \subseteq V_k$ from now on. We randomly select p other vertices (with replacements) in step 8. Denote the jth such (θh)-multiset by S_j. The algorithm then checks if the induced subgraph of $S_j \cup S$ contains a copy of H.

Let event B mean $S_j \cup S$ contains a copy of H for all j. Given $S \subseteq V_k$, if more than θh vertices are selected from V_k in step 8, then event B will not occur (note that $\theta \geq 1$). Thus the probability of event B is at most the probability that the algorithm selects fewer than θh vertices from V_k in step 8. Let y be the number of vertices of these p vertices selected in step 8 that belong in V_k (with multiplicity counted). Then $\Pr[\text{ event } B\ ] \leq \Pr[\ y < \theta h\ ]$. We estimate the upper bound of the above probability by the Chernoff bound. As the probability of selecting a vertex in V_k is $k/N = \lambda$ and the total number of selections is $p = 6\theta h/\lambda$, we have $\mu = E[\ y\ ] = (6\theta h/\lambda)\lambda = 6\theta h$. Rewrite $\Pr[\text{ event } B\ ] = \Pr[\ y < (1-\Delta)6\theta h\ ]$, where $\Delta = 5/6$. By the Chernoff bound, $\Pr[\text{ event } B\ ] \leq e^{-\mu\Delta^2/2} = e^{-6\theta h(5/6)^2/2} = e^{-25\theta h/12}$. Since $\theta h > 1$, $\Pr[\text{ event } B\ ] < e^{-2} < 1/6$. Hence the probability that we select an h-set from V_k in step 6 that leads to acceptance in step 12 is at least $(1-1/6)(1-1/6) > 2/3$. The probability that a digraph G which has property $P_{k,H}$ will be rejected is thus less than $1/3$. See Fig. 5.2 for illustration.

On the other hand, suppose the input graph $G = (V, E)$ is ϵ-far from any digraph which has property $P_{k,H}$. Obviously, the probability that the algorithm accepts is equal to the probability that we find an h-set S whose induced subgraph does not contain an H, and after we randomly select p additional vertices (with replacements), there exists a (θh)-multiset S_j from those p selected vertices such that the induced subgraph of $S_j \cup S$ contains no copies of H. By Claim 15, the probability of finding an h-set that contains copies of H is at least $f(\epsilon; H)h!/\left[\binom{\binom{h}{2}}{m}\lambda\right]$. For each (θh)-multiset S_j, at least θ disjoint h-sets are checked; hence the probability that $S_j \cup S$

contains copies of H is at least $\left(\frac{f(\epsilon;H)h!}{\binom{\binom{h}{2}}{m}\lambda}\right)^{\theta} = (\lambda/6)^{\theta} \cdot (2/3)^{1/\kappa}$. We then test $\binom{p}{\theta h}$ (θh)-multisets in step 12. Since

$$\binom{p}{\theta h} = \frac{(6\theta h/\lambda)!}{(\theta h)!} = \frac{(6\theta h/\lambda)[(6\theta h/\lambda)-1]\cdots[(6\theta h/\lambda)-\theta h]}{(\theta h)!} > (6/\lambda)^{\theta h} > (6/\lambda)^{\theta},$$

the probability that the induced subgraph of $S_j \cup S$ contains copies of H for all j is at least $(6/\lambda)^{\theta} \cdot (\lambda/6)^{\theta} \cdot (2/3)^{1/\kappa} = (2/3)^{1/\kappa}$. So, for each h-set S that passes the test in step 7, the probability that S does not lead to acceptance in step 12 is at least $(2/3)^{1/\kappa}$. Hence, regardless whether S passes the test in step 7, the probability that none of the S leads to acceptance in step 12 is at least $\left[(2/3)^{1/\kappa}\right]^{\kappa} = 2/3$. Therefore, the probability that the algorithm accepts the input is less than $1/3$.

The query complexity of step 7 is $O(h^2)$ and the query complexity from step 9 to step 10 is $O\left(\binom{p}{\theta h}\binom{\theta}{2}\right)$. Since $\binom{p}{\theta h}\binom{\theta h}{2} > h^2$, the query complexity is $O\left(\kappa\binom{p}{\theta h}\binom{\theta h}{2}\right)$. This value is independent of N. Hence the theorem follows. **Q.E.D.**

The value of $f(\epsilon;H)$ decreases extremely fast with ϵ, and is independent of n [7]. Although it is difficult to compute the exact value of $f(\epsilon;H)$ in general, we can estimate a lower bound of $f(\epsilon;H)$ by Szemerédi's regularity lemma, and $[(1-\epsilon)/(2+h)]^h$ is one such lower bound. In our algorithm in Fig. 5.1, $f(\epsilon;H)$ is just a coefficient. The soundness of our algorithm in Fig. 5.1 is proved in Theorem 16. We can replace $f(\epsilon;H)$ by $[(1-\epsilon)/(2+h)]^h$ in step 4 of our algorithm in Fig. 5.1 without changing the validity of Theorem 16. The consequence is that our algorithm needs to query more edges in the input digraph, but the total number of queried edges remains independent of the input size.

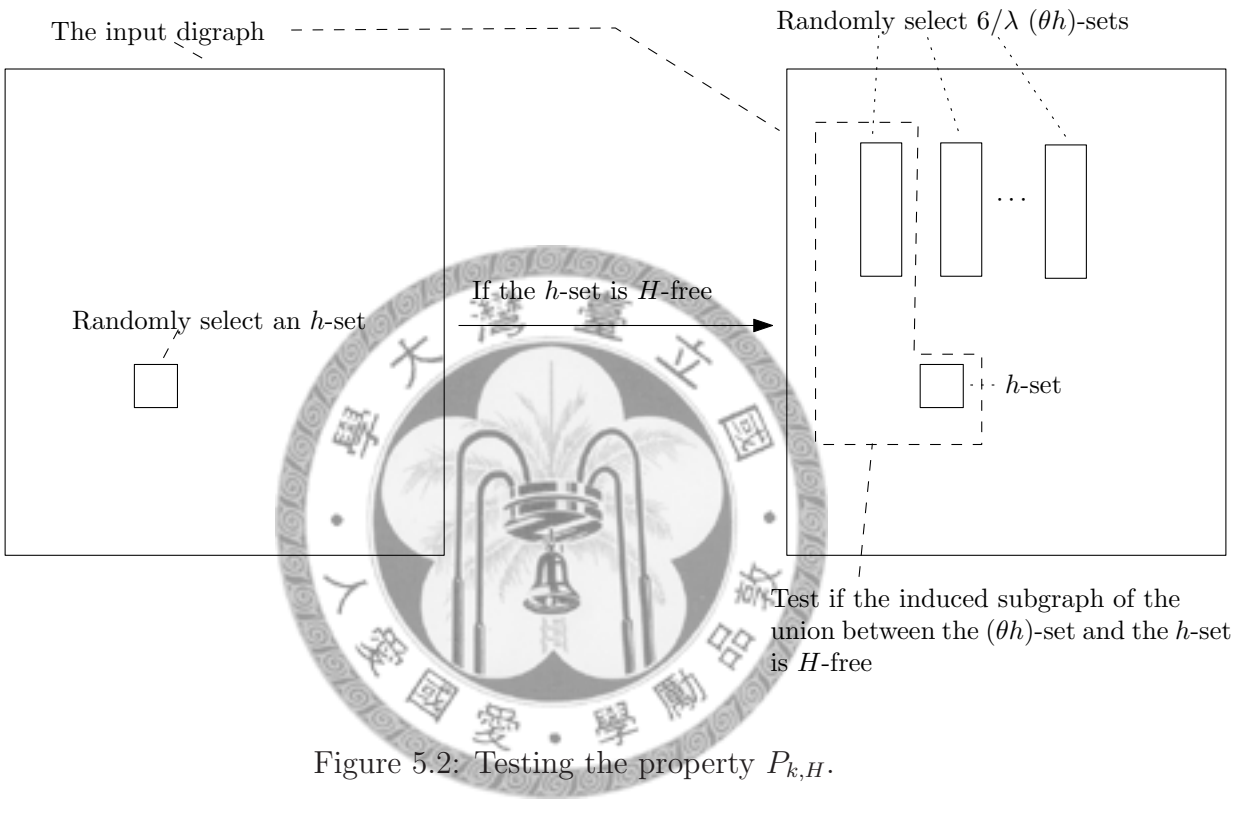

Figure 5.2: Testing the property $P_{k,H}$.

Let $k > \sqrt{\epsilon}N$, ϵ be a constant and $H_1 = (V_1, E_1)$ be a digraph where $V_1 = \{v_1, v_2, \ldots, v_{d+1}\}$ and $E_1 = \{(v_1, v_{d+1}), (v_2, v_{d+1}), \ldots, (v_d, v_{d+1})\}$. It is commonly called a star graph. We can use the algorithm in Fig. 5.1 to test whether the input digraph contains an H_1-free k-induced subgraph. Obviously if a digraph is accepted by our algorithm, then the maximum indegree of this digraph is bounded by d with high probability. Similarly, let $H_2 = (V, E_2)$ be a digraph where $E_2 = \{(v_{d+1}, v_1), (v_{d+1}, v_2), \ldots, (v_{d+1}, v_d)\}$. We can use the algorithm in Fig. 5.1 to test whether the maximum outdegree of the input digraph is bounded by d with high probability. If an input digraph is accepted by our algorithm for both H_1 and H_2, then we know that this digraph satisfies the restrictions of the BR tester. In this case, we use the BR tester to test strong connectivity of the input digraph. The total query complexity of testing strong connectivity is the sum of the query complexities of the algorithm in Fig. 5.1 and the BR tester. Since the query complexities of the algorithm in Fig. 5.1 and the BR tester are both independent of the input size, the sum of the query complexities of both algorithms remains independent of the input size. The query complexity of our strong connectivity tester in Fig. 4.2 is the square root of the input size. Hence, the sum of the query complexities of the algorithm in Fig. 5.1 and the BR tester is less than the query complexity of our strong connectivity tester in Fig. 4.2. Since the main efficiency parameter of a method to solve a property testing problem is its query complexity, our strong connectivity tester is not the most efficient one for all digraphs. It is better to use the algorithm in Fig. 5.1 to determine which tester (the BR tester or our strong connectivity tester) should

be used to test the strong connectivity of the digraph.

Chapter 6

Testing of Group Properties

6.1 Finite group-like structure

A finite group-like structure s is a four-tuple $(\Gamma, \circ, i, 1)$, where Γ is the groundset of s, $\circ$ is a binary operator, i is the inverse operator, and 1 is the identity element. Finite groups are finite group-like structures where $\circ$ is the group multiplication [37]. Let S be a family of finite group-like structures and $\Pi \subseteq S$. We say a finite group-like structure s has property Π (or s satisfies property Π) if it is an element of Π. An ϵ-tester for a property Π is a randomized algorithm that is given a finite group-like structure s and a distance parameter ϵ. The tester can make queries as to the results of operations on elements of s. The total number of queries is the query complexity of the tester. Let the property Π be $\{s_i\}$. Given an upper bound M on the size of the groundset, the tester needs to distinguish with probability at least $2/3$ between the case of s having Π and the case of the minimum cost to transform s to any s_i being at least ϵM^2 (we will define the cost metric shortly). In the latter case, s is

said to be ϵ-far from having property Π. The probability $2/3$ can be replaced by any constant smaller than 1 as the algorithm can be repeated if necessary.

The cost used for transforming group-like structures will be similar to the edit distance for strings. As a result, it will make sense to "correct" group-like structures by modifying the operations and sizes of their groundsets. A table of size k is a $k \times k$ matrix K whose element in row i and column j is denoted by k_{ij} for $1 \leq i, j \leq k$. Three operations transform the table K into a new table. An exchange operation at place (i, j) modifies the value k_{ij} and leaves others unchanged. The cost of an exchange is 1. An insert operation at index i, where $1 \leq i \leq k+1$, transforms K into new table of size $k+1$ by inserting $2k+1$ elements to make a new row and a new column of index i. The cost of an insert is $2k+1$. A delete operation at index i, where $1 \leq i \leq k$, transforms K into a new table of size $k-1$ by deleting the ith row and the ith column. The cost of a delete is $2k-1$. Let $\circ : \Gamma \times \Gamma \longrightarrow \Gamma$ be a binary operator, where $\Gamma = \{g_1, g_2, \ldots, g_k\}$ is a finite set of size k. A table K of size k is said to represent $\circ$ if $k_{ij} = g_i \circ g_j$ for $1 \leq i, j \leq k$ [37].

6.2 Research work related to group property testing

A group-like structure property is testable if the property has an ϵ-tester (or simply a tester) and the cost is sublinear in the input size M^2. The first testers are constructed for algebraic problems under the name of self-testers [22, 57]. Blum, Luby and Rubinfeld construct the first homomorphism tester for abelian groups [22] (we call

it the BLR tester). The BLR tester is extended to non-abelian groups by Ben-Or, Coppersmith, Luby and Rubinfeld [18]. Several works have dealt with reducing the $2\log|G|$ random bits per basic trial of the BLR tester in abelian groups [19, 45, 66]. Improved analyses relating the distance to the rejection probability have been given for testing homomorphism [13, 15, 16]. More recently, Friedl, Ivanyos and Santha construct a tester which, given a finite group-like structure, tests if it is an abelian group (we call it the FIS tester) [37]. The query complexity of the tester is polylogarithmic in the size of the groundset. For reading convenience, we denote group-like structures by their groundsets and operators.

6.3 Tester construction

Friedl, Ivanyos and Santha modify the quantum algorithm of Cheung and Mosca [28] to test if a finite group-like structure $s = (\Gamma, \circ, i, 1)$ is an abelian group (we call it the FIS tester earlier). We use the FIS tester in the first part of our testing algorithm. If it fails, we reject s. If the FIS tester does not reject s, then we test whether there exists a k-subset of groundset Γ whose Cayley graph is strongly connected.

Definition 17 ([31]) *Given a prime number p, a p-group is a group in which each element has a power of p as its order. A Sylow p-subgroup of a finite group G is a maximal p-subgroup of G (i.e., a Sylow p-subgroup is not a proper subgroup of any other p-subgroup of G).*

Denote a finite group being generated by a set $\{s_1, \ldots, s_m\}$ as $\langle s_1, \ldots, s_m \rangle$ and a finite group being generated by a set S as $\langle S \rangle$. We next present a series of useful

results before proving the main result of this paper.

Fact 18 ([31]) *(1) Let G be a finite group whose order is a multiple of a prime p. Write $|G| = p^n s$, where $n > 0$ and p does not divide s. Then G has a Sylow p-subgroup with order p^n. (2) For any cyclic group H, the number of generators of H is $\phi(|H|)$, where ϕ is Euler's totient function. (3) If G is a finite group and p is a prime dividing $|G|$, then G has an element of order p. (4) If G is a finite group and H is a subgroup of G, then the order of H divides the order of G.*

Lemma 19 ([31]) *Let $s = (\Gamma, \circ, i, 1)$ be a finite group and $g' \subset \Gamma$. If g' does not generate the group s, then the Cayley graph associated with g' is disconnected and each connected component of the Cayley graph represents a coset of the subgroup generated by g'.*

Lemma 20 *Let $s = (\Gamma, \circ, i, 1)$ be a finite group, k be an integer, $1 \leq k < |\Gamma|$ and $C = \{c_1, \ldots, c_q\}$ be a generator set of s with $q > k$ such that no proper subset of C is a generator set of s. Let the order of c_i be m_i and $\phi(m_1) \geq \phi(m_2) \geq \cdots \phi(m_q) \geq 1$. Then the Cayley graph associated with any k-set of C is not strongly connected. Furthermore, for every k-set, we have to add at least $|\Gamma| / \prod_{x=1}^{k} c_{i_x}^{m_{i_x}}$ edges to its Cayley graph to make the graph strongly connected.*

Proof: Since $c_i^{m_i} = 1$ for all $1 \leq i \leq q$, without loss of generality, we assume that C does not contain the identity element. By Theorem 2, the Cayley graph associated with any k-subset of C is not strongly connected.

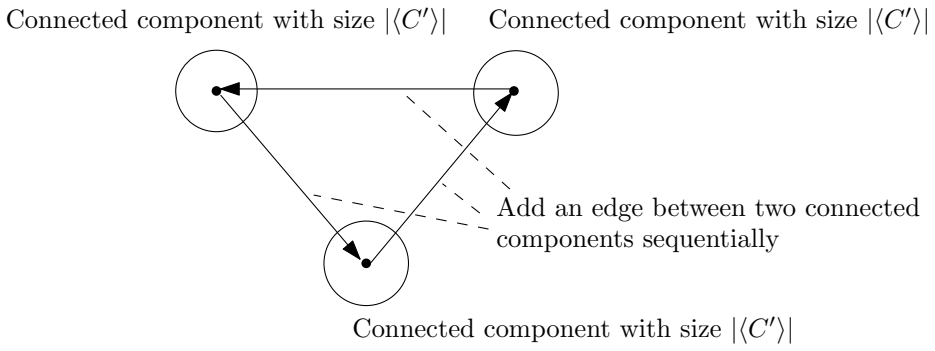

Figure 6.1: Cayley graph associated with C'.

Let $C' = \{c_{i_1}, c_{i_2}, \ldots, c_{i_k}\}$ be an arbitrary k-subset of C. For the Cayley graph associated with C', by Lemma 19, the size of every connected component is $|\langle C' \rangle|$, there are $|\Gamma|/|\langle C' \rangle|$ different connected components and there is no egde between any two different connected components.

We know that $|\langle C' \rangle| \leq \prod_{x=1}^{k} c_{i_x}^{m_{i_x}}$. Hence there are at least $|\Gamma|/\prod_{x=1}^{k} c_{i_x}^{m_{i_x}}$ connected components in the Cayley graph associated with C'. Furthermore, we have to add at least that many edges to make the Cayley graph associated with C' strongly connected. See Fig. 6.1 for illustration. **Q.E.D.**

Lemma 21 *Let F_k be the family of finite groups with a k-generator set. Suppose a finite group $s = (\Gamma, \circ, i, 1)$ is ϵ-far from every $f \in F_k$, $p_1^{q_1} p_2^{q_2} \cdots p_y^{q_y}$ is the prime factorization of $|\Gamma|$, $\phi(p_1^{q_1}) > \phi(p_2^{q_2}) > \cdots > \phi(p_y^{q_y})$, $p_1^{q_1} p_2^{q_2} \cdots p_y^{q_y} \leq (1-\epsilon)M^2$ and $p_1^{q_1} p_2^{q_2} \cdots p_{y+1}^{q_{y+1}} > (1-\epsilon)M^2$. Then for any k-subset K of Γ, we have to add at least $|\Gamma|/(1-\epsilon)M^2$ edges to the Cayley graph associated with K to make it strongly connected.*

Proof: By Fact 18(1), s contains Sylow p_i-subgroup s_i for $i = 1, 2, \ldots, y$. By Fact 18(4), no Sylow p_i-subgroup is a subgroup of any Sylow p_j-subgroups for all $i \neq j$. So, if a set g generates s, then $\bigcup_{i=1}^{y} g_i \subseteq g$ where g_i generate s_i. Since s is ϵ-far from every group from F_k, s has no k-generator set and $|g| > k$. Let f be an element in F_k that is closest to s. Since f has a generator set with size k, it does not contain all s_l for $1 \leq l \leq y$; otherwise, s is an element of F_k and it is a contradiction. Hence, if we want to modify s to be f, we have to remove some s_m from s or reduce some s_m's size. So, since s is ϵ-far from f and $p_1^{q_1} p_2^{q_2} \cdots p_z^{q_z} \leq (1-\epsilon)M^2$, for any k-subset $K \subseteq \Gamma$, the elements in K generate at most z Sylow p_l-subgroups with $1 \leq l \leq z$. By Lemma 20, we have to add at least $|\Gamma|/(1-\epsilon)M^2$ edges to make the Cayley graph associated with s strongly connected. **Q.E.D.**

Note that for the input group-like structure $(\Gamma, \circ, i, 1)$, the input size is the size of the table corresponding to the function $\circ : \Gamma \times \Gamma \to \Gamma$. Thus, the upper bound of the table is M^2, and M^2 is the input size. Let k_i be a k-subset of Γ and G_{k_i} be k_i's Cayley graph. Assume that there is a bound on the indegree and outdegree of each vertex in every G_{k_i}. We will prove that our algorithm in Fig. 6.2 will work and the query complexity is polylogarithmic in the input size.

Theorem 22 *Let* $\mathtt{S}$ *be the family of finite group-like structures, the upper bound of groundsets be* M*, and* $F_a \subseteq \mathtt{S}$ *be the family of finite abelian groups with* k*-generator set. If* M *is large enough, for every* $\epsilon > 0$*, there exists an* ϵ*-tester for* F_a *and let* $s = (\Gamma, \circ, i, 1)$ *be the input group-like structure, the query complexity of the* ϵ*-tester for* s *is polylogarithmic in the input size.*

Use the FIS tester to test if $s = (\Gamma, \circ, i, 1)$ is ϵ-far from being a finite abelian group
if the FIS tester rejects **then**
  REJECT
end if
{Let $t < k$. In step 1, FIS tester chooses many random elements $\gamma_1, \ldots, \gamma_t$ from Γ and gives the orders $m_1, \ldots, m_t$ of $\gamma_1, \ldots, \gamma_t$, respectively}
if exist m_i, m_j are relatively prime **then**
  Let $W = \{m_j |$ all m_j are relatively prime $\}$
else
  $W = \{\max_{1 \le i \le t} m_i\}$
end if
$l = \prod_{m_j \in W} \gamma_j^{m_j}$ and $\epsilon' = \frac{\epsilon M^2 - \sum_{i=|\Gamma|-l}^{|\Gamma|}(2i-1)}{M^2}$
Let $\Phi = \max\{\frac{|\Gamma|}{(1-\epsilon)M^4}, \epsilon'\}$
Uniformly and independently select $m = \Theta(1/\epsilon k)$ vertices $a_1, a_2, \ldots, a_m$ in Γ
Let $A = \{a_1, a_2, \ldots, a_m\}$ and $i = 1$
while $i < \binom{|\Gamma|}{k} + 1$ **do**
  Select the ith k-set K_i {note that there are $\binom{|\Gamma|}{k}$ k-sets in V}
  Use the BR tester on A to test if G_{K_i} (G_{K_i} is K_i's Cayley graph) is Φ-far from being a strongly connected digraph
  if the BR tester accepts G_{K_i} **then**
    ACCEPT
  else
    $i \leftarrow i + 1$
  end if
end while
REJECT

Figure 6.2: An ϵ-tester for the proposition whether a finite group-like structure is a finite abelian group with a k-generator set. This algorithm will be used in the case that there is a prior bound on the indegree and outdegree of each vertex in every k-set's Cayley graph.

Proof: Suppose the input s is a correct instance, by [37] and Theorem 3, our algorithm in Fig. 6.2 will accept it. On the other hand, assume s is ϵ-far from every elements of F_a. The probability of our algorithm in Fig. 6.2 rejecting it is at least 2/3. But if the FIS tester does not reject s and s is ϵ-close to a finite group, our algorithm in Fig. 6.2 will still reject it with probability at least 2/3. We prove it as follows. Recall that in the beginning of our algorithm, the FIS tester picks sufficiently many random elements $\gamma_1, \ldots, \gamma_t$ from Γ, and the FIS tester only accepts s when the set $\{\gamma_1, \ldots, \gamma_t\}$ can generate a subgroup $s' = (\Gamma', \circ', i', 1')$. Since $\epsilon' = \frac{\epsilon M^2 - \sum_{i=|\Gamma|-l}^{|\Gamma|}(2i-1)}{M^2}$, we know that $\sum_{i=|\Gamma|-l}^{|\Gamma|}(2i-1) < (\epsilon - \epsilon')M^2$. So s' contains no k-generator set since, otherwise, we could transform s to be $s' \in F_a$ by removing $|\Gamma| - l$ elements of s with a cost at most ϵM^2. Then, s is ϵ-close to an element of F_a, a contradiction. Thus, let the prime factorization of $|\Gamma|$ be $p_1^{q_1} p_2^{q_2} \cdots p_y^{q_y}$ where $\phi(p_1^{q_1}) > \phi(p_2^{q_2}) > \cdots > \phi(p_y^{q_y})$ and $\Phi = \max\{\frac{\phi(p_y^{q_y})}{|\Gamma|^2}, \epsilon'\}$. If the FIS tester accepts s, by Lemma 21, all Cayley graphs of k-sets of Γ' are at least Φ-far from being strongly connected. By Theorem 2, our algorithm will reject s with probability at least 2/3.

The query complexity of the FIS tester is polylogarithmic in M and the query complexity of the BR tester is $O(1/\epsilon)$. Obviously, the query complexity of our algorithm is polylogarithmic in M. **Q.E.D.**

In an abelian group $(\Gamma, \circ, i, 1)$, for each k-subset K of Γ, the indegree and outdegree of the Cayley graph associated with K are bounded by k. Suppose k is a constant. We use the BR tester to test the input digraph. The query complexity is polylogarithmic in the input size. But if k is not a constant, we use the CONN tester

instead of the BR tester. The new algorithm appears in Fig. 6.3. Since the query complexity of the CONN tester is linear in the square root of the input size, in the worst case, the query complexity of the new algorithm is linear in the square root of the input size, too. Obviously, if we want to test if a group-like structure is an abelian group with a k-generator set deterministically, we need to query all results of the binary operation. Hence the query complexity will be the input size. The efficiencies of the deterministic algorithm and our testing algorithm are compared in Fig. 6.4.

Use the FIS tester to test if $s = (\Gamma, \circ, i, 1)$ is ϵ-far from being a finite abelian group
if the FIS tester rejects **then**
  REJECT
end if
{ Let $t < k$. In step 1, FIS tester chooses many random elements $\gamma_1, \ldots, \gamma_t$ from Γ and gives the orders $m_1, \ldots, m_t$ of $\gamma_1, \ldots, \gamma_t$, respectively}
if exist m_i, m_j are relatively prime **then**
  Let $W = \{m_j |$ all m_j are relatively prime $\}$
else
  $W = \{\max_{1 \le i \le t} m_i\}$
end if
$l = \prod_{m_j \in W} \gamma_j^{m_j}$ and $\epsilon' = \frac{\epsilon M^2 - \sum_{i=|\Gamma|-l}^{|\Gamma|}(2i-1)}{M^2}$
Let $\Phi = \max\{\frac{1}{(1-\epsilon)M^4}, \epsilon'\}$
Uniformly and independently select $m = \Theta(1/\epsilon k)$ vertices $a_1, a_2, \ldots, a_m$ in Γ
Let $A = \{a_1, a_2, \ldots, a_m\}$ and $i = 1$
while $i < \binom{|\Gamma|}{k} + 1$ **do**
  Select the ith k-set K_i {note that there are $\binom{|\Gamma|}{k}$ k-sets in V}
  if k is a constant **then**
    Use the BR tester on A to test if G_{K_i} (G_{K_i} is K_i's Cayley graph) is Φ-far from being a strongly connected digraph
    if the BR tester accepts G_{K_i} **then**
      ACCEPT
    else
      $i \leftarrow i + 1$
    end if
  else
    Use the CONN tester on G_{K_i} to test if it is Φ-far from being a strongly connected digraph
    if the CONN tester accepts G_{K_i} **then**
      ACCEPT
    else
      $i \leftarrow i + 1$
    end if
  end if
end while
REJECT

Figure 6.3: An ϵ-tester for the proposition whether a finite group-like structure is a finite abelian group with a k-generator set.

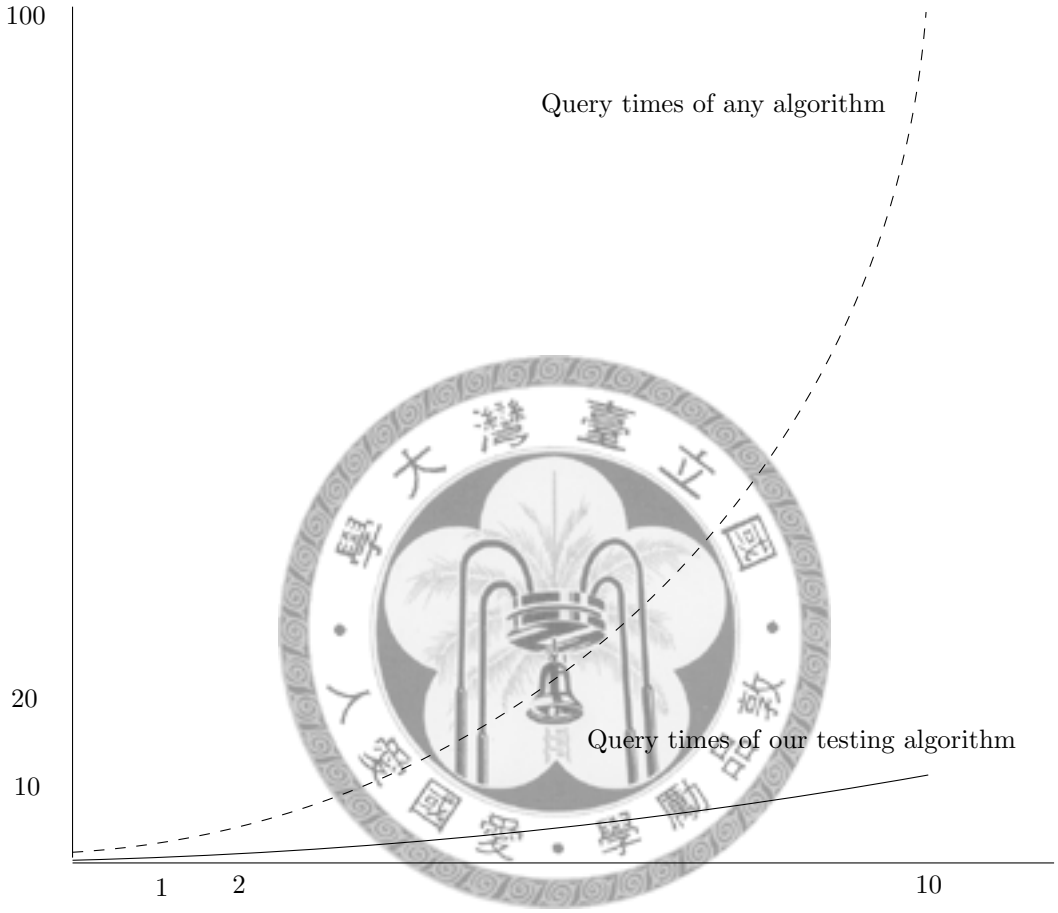

Figure 6.4: Efficiency comparison when $M = 10$.

Chapter 7

Conclusion

This dissertation develops a tester to test the strong connectivity of digraphs. Then we construct an algorithm for testing digraphs with an H-free k-induced subgraph. In the last part of this dissertation, this dissertation combines the strong connectivity tester to test if a finite group-like structure has a k-generator set. The fundamentals of the dissertation rely on the following oracles:

1. We can specify any adjacency matrix of a digraph G and ask whether an edge exists between any pair of vertices.

2. For any group-like structure $(\Gamma, \circ, i, 1)$, we can ask the result of $x \circ y$ for all $x, y \in \Gamma$.

The query complexities of our property testing algorithms are very low and our results are efficient.

Bibliography

[1] A. Abdollahi, A. Faghihi and A. M. Hassanabadi, *3-generator groups whose elements commute with their endomorphic images are abelian*, Communications in Algebra, 36(10) (2008), pp. 3783–3791.

[2] A. Abdollahi, A. Faghihi and A. M. Hassanabadi, *Minimal Number of Generators and Minimum Order of a Non-Abelian Group whose Elements Commute with Their Endomorphic Images*, Communications in Algebra, 36(5) (2008), pp. 1976–1987.

[3] S. B. Akers and B. Krishnamurthy, *A group-theoretic model for symmetric interconnection networks parallel processing*, International Conference Parallel Processing, (1986), pp. 216–233.

[4] N. Alon, E. Fischer, M. Krivelevich and M. Szegedy, *Efficient Testing of Large Graphs*, FOCS (1999), pp. 656–666.

[5] N. Alon, E. Fischer, I. Newman and A. Shapira, *A Combinatorial Characterization of the Testable Graph Properties: It's All About Regularity*, STOC (2006), pp. 251–260.

[6] N. Alon and A. Shapira, *A Characterization of Easily Testable Induced Subgraphs*, SODA (2004), pp. 942–951.

[7] N. Alon and A. Shapira, *Testing Subgraphs in Directed Graphs*, STOC (2003), pp. 700–709.

[8] F. S. Annexstein, M. Baumslag and A. L. Rosenberg, *Group Action Graphs and Parallel Architectures*, SIAM J. Comput., 19(3) (1990), pp. 544-569.

[9] G. B. Arfken, Generators, *Mathematical Methods for Physicists*, 3rd ed. Orlando, Academic Press, 1985.

[10] S. Arora, C. Lund, R. Motwani, M. Sudan and M. Szegedy, *Proof Verification and Hardness of Approximation Problems*, FOCS (1992), pp 14–23.

[11] L. Babai and P. Erdős, *Representation of Group Elements as Short Products*, Annals of Discrete Mathematics, 12 (1982), pp. 27–30.

[12] L. Babai, L. Fortnow, L. A. Levin and M. Szegedy, *Checking Computations in Polylogarithmic Time*, STOC (1991), pp. 21–31.

[13] M. Bellare, D. Coppersmith, J. Håstad, M. A. Kiwi, and M. Sudan, *Linearity Testing in Characteristic Two*, IEEE Transactions on Information Theory, 42(6) (1996), pp. 1781–1795.

[14] M. Bellare, O. Goldreich and M. Sudan, *Free Bits, PCPs and Non-Approximability - Towards Tight Results*, FOCS (1995), pp. 422–431.

[15] M. Bellare, S. Goldwasser, C. Lund, and A. Russell, *Efficient Probabilistically Checkable Proofs and Applications to Approximations*, STOC (1993), pp. 294–304.

[16] M. Bellare and M. Sudan, *Improved Non-approximability Results*, STOC (1994), pp. 184–193.

[17] M. A. Bender and D. Ron, *Testing Properties of Directed Graphs: Acyclicity and Connectivity*, Random Structures and Algorithms, 20 (2002), pp. 184–205.

[18] M. Ben-Or, D. Coppersmith, M. Luby, and R. Rubinfeld, *Non-abelian Homomorphism Testing, and Distributions Close to Their Self-convolutions*, RANDOM-APPROX (2004), pp. 273–285.

[19] E. Ben-Sasson, M. Sudan, S. P. Vadhan and A. Wigderson, *Randomness-efficient Low Degree Tests and Short PCPs via Epsilon-Biased Sets*, STOC (2003), pp. 612–621.

[20] J. Bermond, T. Kodate and S. Perennes, *Gossiping in Cayley Graphs by Packets*, Combinatorics and Computer Science, (1995), pp. 301–315

[21] S. N. Bhatt, F. R. K. Chung, J. W. Hong, F. T. Leighton and A. L. Rosenberg, *Optimal Simulations by Butterfly Networks (Preliminary Version)*, STOC (1988), pp. 192–204.

[22] M. Blum, M. Luby, and R. Rubinfeld, *Self-testing/correcting with Applications to Numerical Problems*, J. Computer and System Sciences, 47(3) (1993), pp. 549–595.

[23] B. Bollobás, *Complete Subgraphs Are Elusive*, J. Comb. Theory (Series B), 21 (1976), pp. 1–7.

[24] M. R. Bridson and M. Tweedale, *Deficiency and abelianized deficiency of some virtually free groups*, Math. Proc. Camb. Phil. Soc., Vol 143 (2007), pp. 257–264.

[25] R. M. Bryant, L. Kovcás and G. R. Robinson, *Transitive permutation groups and irreducible linear groups*, Quart. J. Math. Oxford, 46(2) (1995), pp. 385–407.

[26] P. J. Cameron, R. Solomon and A.Turull, *Chains of subgroups in symmetric groups*, J. Algebra, 127 (1989), pp. 340–352.

[27] G. E. Carlsson, J. E. Cruthirds, H. B. Sexton and C. G. Wright, *Interconnection Networks Based on a Generalization of Cube-Connected Cycles*, IEEE Trans. Computers, 34(8) (1985), pp. 769–772.

[28] K. Cheung and M. Mosca, *Decomposing finite abelian groups*, Quantum Information and Computation, 1(3) (2008), pp. 26–32.

[29] C. E. Chronaki, *A Survey of Evasiveness: Lower Bounds on the Decision-Tree Complexity of Boolean Functions*, Available on http://www.ics.forth.gr/ chronaki/papers/ur/eve.ps, (2002).

[30] Y. Dodis, O. Goldreich, E. Lehman, S. Raskhodnikova, D. Ron and A. Samorodnitsky, *Improved Testing Algorithms for Monotonicity*, RANDOM-APPROX (1999), pp. 97–108.

[31] D. S. Dummit and R. M. Foote. *Abstract Algebra*, New Jersey, Prentice-Hall, Inc, 1991.

[32] D. B. A. Epstein, *Finite Presentations of Groups and 3-Manifolds*, The Quarterly Journal of Mathematics, 12(1) (1961), pp. 205–212.

[33] F. Ergun, S. Kannan, R. Kumar, R. Rubinfeld and M. Viswanathan, *Spot-Checkers*, STOC (1998), pp. 259–268.

[34] U. Feige and S. Kogan, *The Hardness of Approximating Hereditary Properties*, Available on http://research.microsoft.com/research/theory/feige/homepagefiles/hereditary.pdf, (2005).

[35] E. Fischer and L. Fortnow, *Tolerant Versus Intolerant Testing for Boolean Properties*, Electronic Colloquium on Computational Complexity (ECCC), 11(105) (2004).

[36] E. Fischer and I. Newman, *Testing versus Estimation of Graph Properties*, STOC (2005), pp. 138–146.

[37] K. Friedl, G. Ivanyos and M. Santha, *Efficient Testing of Groups*, STOC (2005), pp. 157–166.

[38] Z. Furedi, A. Naor and J. Verstraete, *On the Turan Number for the Hexagon*, Available on
http://research.microsoft.com/research/
theory/naor/homepage%20files/final-hexagons.pdf, (2004).

[39] R. Gold and J. Kim, *Bases for Cyclotomic Units*, Compositio Mathematica, 71(1) (1989), pp. 13–27.

[40] O. Goldreich, S. Goldwasser, E. Lehman, D. Ron and A. Samorodnitsky, *Testing Monotonicity*, Combinatorica, 20(3) (2000), pp. 301–337.

[41] O. Goldreich, S. Goldwasser and D. Ron, *Property Testing and Its Connection to Learning and Approximation*, J. ACM, 45(4) (1998), pp. 653–750.

[42] O. Goldreich and D. Ron, *A Sublinear Bipartiteness Tester for Bounded Degree Graphs*, Combinatorica, 19(3) (1999), pp. 335–373.

[43] J. Håstad, *Some Optimal Inapproximability Results*, STOC (1997), pp. 1–10.

[44] J. Håstad, *Testing of the Long Code and Hardness for Clique*, STOC (1996), pp. 11–19.

[45] J. Håstad and A. Wigderson, *Simple Analysis of Graph Tests for Linearity and PCP*, Random Structures and Algorithms, 22(2) (2003), pp. 139–160.

[46] D. A. Holton, and J. Sheehan. *The Petersen Graph*, Cambridge, England, Cambridge University Press, 1993.

[47] R. C. Holt, E. M. Reingold, *On the Time Required to Detect Cycles and Connectivity in Graphs*, Mathematical Systems Theory, 6(2) (1972), pp. 103–106.

[48] M. Jerrum, *A compact presentation for permutation groups*, J. Algorithms, 7 (1986), pp. 60–78.

[49] G. Karpilovsky, *The Schur multiplier*. London Mathematical Society Monographs, New Series 2. Oxford: Clarendon Press. XIV, 1987.

[50] L. H. Kauffman, *Virtual Knot Theory*, Europ. J. Combinatorics, 20 (1999), pp. 663–691.

[51] M. J. Kearns and D. Ron, *Testing Problems with Sublearning Sample Complexity*, J. Comput. Syst. Sci., 61(3) (2000), pp. 428–456.

[52] S. G. Kim, *Virtual Knot Groups*, Mathematics Subject Classification, Primary 57M25, 1991.

[53] D. G. Kirkpatrick, *Determining Graph Properties from Matrix Representations*, STOC (1974), pp. 84–90.

[54] M. A. Kiwi, *Probabilistically Checkable Proofs and the Testing of Hadamard-like Codes*, Ph.D. dissertation. Massachusetts Institute of Technology, Cambridge, Mass, 1996.

[55] L. Kovács and M. F. Newman, *Generating transitive permutation groups*, Quart. J. Math. Oxford, 39(2) (1988), pp. 361–372.

[56] T. Kövari, V. T. Sós and P. Turán, *On a Problem of K. Zarankiewicz*, Colloquium Math., 3 (1954), pp. 50–57.

[57] R. Lipton, *New Directions in Testing*. Series in Discrete Mathematics and Theoretical Computer Science ACM/AMS, (2), 1991.

[58] A. Lucchini, F. Menegazzo and M. Morigi, *Asymptotic results for transitive permutation groups*, Bull. London Math. Soc., 32 (2000), pp. 191–195.

[59] F. Menegazzo, *The Number of Generators of a Finite Group*, Irish Math. Soc. Bulletin, 50 (2003), pp. 117–128.

[60] D. R. Morrison, *On K3 Surfaces with Large Picard Number*, Inventiones Mathematicae, 75(1) (1984), pp. 105–121.

[61] B. H. Neumann, *A two-generator group isomorphic to a proper factor group*, J. London Math. Soc., s1-25(4) (1950), pp. 247–248.

[62] M. Parnas, D. Ron and R. Rubinfeld, *Tolerant Property Testing and Distance Approximation*, Electronic Colloquium on Computational Complexity (ECCC), 11(10) (2004).

[63] R. L. Rivest, J. Vuillemin, *A Generalization and Proof of the Aanderaa-Rosenberg Conjecture*, STOC (1975), pp. 6–11.

[64] R. Rubinfeld and M. Sudan, *Robust Characterizations of Polynomials with Applications to Program Testing*, SIAM J. Comput., 25(2) (1996), pp. 252–271.

[65] Y. Saad and M. H. Schultz, *Topological Properties of Hypercubes*, IEEE Transactions on computers, (1988), pp. 867–872.

[66] A. Samorodnitsky and L. Trevisan, *A PCP Characterisation of NP with Optimal Amortized Query Complexity*, STOC (2000), pp. 191–199.

[67] P. Shor, *Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer*, SIAM J. Computing, 26(5) (1997), pp. 1484–1509.

[68] L. Trevisan, *Recycling Queries in PCPs and in Linearity Tests (Extended Abstract)*, STOC (1998), pp. 299–308.

[69] H. F. Trotter, *Homology of Group Systems With Applications to Knot Theory*, The Annals of Mathematics, 2nd Ser., 76(3) (1962), pp. 464–498.