

# Kings in Generalized Tournaments

BY

Cheng Zhang

A Study

Presented to the Faculty

of

Wheaton College

in Partial Fulfillment of the Requirements

for

Graduation with Departmental Honors

in Mathematics

Norton, Massachusetts

May, 2018

# Abstract

This thesis explores how to find and construct kings in three generalizations of tournament: semi-complete digraphs, oriented graphs and quasi-transitive oriented graphs.

In Chapter 3 and Chapter 4, we present a way to interpret semi-complete digraphs and oriented graphs as tournaments with “ties” (we call the “ties” in semi-complete digraphs “double ties”, and the “ties” in oriented graphs “ties”). In Chapter 3, we prove there exists an  $(n, k)$  semi-complete digraph if and only if  $n \geq k \geq 1$ , and all the  $(n, k)$  semi-complete digraphs that exist can be constructed with at most 1 double tie. In Chapter 4, we prove there exists an  $(n, k)$  oriented graph for all  $n \geq k \geq 0$  except  $(1, 0)$ ,  $(2, 2)$ ,  $(3, 2)$ , and  $(4, 4)$  oriented graphs, and we prove that all the  $(n, k)$  oriented graphs that exist can be constructed with at most 1 tie.

The main focus of this thesis is quasi-transitive oriented graph, which is discussed in Chapter 5. We show an interesting fact that all the quasi-transitive oriented graphs can be condensed into tournaments by “tie component condensations”. Then, we show that the tie component condensation on a quasi-transitive oriented graph is a most efficient condensation to tournament in all the condensations to tournaments defined on all the oriented graph with the same tie structure. Finally we prove that the kings in a quasi-transitive oriented graph  $Q$  are related to the kings in the “underlying tournament of  $Q$ ” (result of  $Q$  after tie component condensation). This result gives us a way to understand the properties of kings in quasi-transitive oriented graphs using the properties of king in tournaments.

# Acknowledgment

I want to first thank my thesis advisor Professor Rochelle Leibowitz for pushing me to find significant results. Without her help, I would not be able to come up with most of the important ideas in this thesis.

Then, I want to thank my parents for being supportive during my toughest days of writing this thesis. My father said to me, “Being truly innovative is always a struggle, but not everyone is great enough to ever experience this kind of hardship.” I want to share this thought with you — future thesis students. Hope it will help you get pass the hardest days of your researches.

I want to thank all friends and professors who reviewed this thesis and provided useful feedbacks. These professors are three members of my thesis committee: Professor Rochelle Leibowitz, Professor William Bloch, and Professor John Gildea, as well as friends, including my dearest girlfriend and best friend, Shi Shen; two of my most valuable and talented friends, Xinyi Liu and Xinru Liu; my only friend who is good with grammar, Yuchen Wang.

I want to thank Professor William Bloch for his Real Analysis course, which drew me into mathematics. I also want to thank Professor Martin Gagné for his Theory of Computation course, which introduced me to the field of theoretical computer science. Without taking these courses, I would not pursue my Ph.D degree in Type Theory.

Finally I want to thank again all my dear friends including Shi Shen, Xinyi Liu, Xinru Liu, Fiona Xu, Weiqi Feng, Yuchen Wang, and many others for these fun and memorable four years of college. Most importantly, I want to express my gratitude to all professors who I had classes with. They not only taught me with knowledge, but also prepared me to face future challenges.

# Contents

<b>Abstract</b>	<b>i</b>
<b>Acknowledgment</b>	<b>ii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Summary of Previous Works . . . . .	1
1.2 Structure of This Thesis . . . . .	2
<b>2 Background</b>	<b>4</b>
2.1 Directed Graph . . . . .	4
2.2 Beating Relations . . . . .	4
2.3 Oriented Graph and Tournament . . . . .	8
2.4 Kings . . . . .	10
<b>3 Semi-complete Digraph</b>	<b>12</b>
3.1 Definitions . . . . .	12
3.2 Properties . . . . .	13
<b>4 Oriented Graph</b>	<b>20</b>
<b>5 Quasi-transitive Oriented Graph</b>	<b>26</b>
5.1 Definitions . . . . .	26
5.2 Ties and Tie Paths . . . . .	27
5.3 Tie Components . . . . .	32
5.4 Graph Condensations . . . . .	37

<i>CONTENTS</i>	iv
5.5 Tie Component Condensations . . . . .	44
5.6 Kings . . . . .	48
<b>6 Further Problems</b>	<b>53</b>
<b>List of Figures</b>	<b>55</b>
<b>Bibliography</b>	<b>57</b>
<b>Index</b>	<b>58</b>

# Chapter 1

## Introduction

### 1.1 Summary of Previous Works

In [6], the author introduced the idea of kings (or 2-kings) in tournaments, and proved that kings in tournaments have many nice properties. In the end of [6], the author proposed several ways to generalize kings, one of them is  $s$ -kings where  $s$  is a positive integer. When we say “king” in a graph theory context, we usually mean 2-king. In [7], the author further investigated the properties of kings in tournaments, and proved many interesting theorems about them.

With good understanding for the properties of kings in tournaments, mathematicians began to study generalized tournaments.

In [3], the author proposed a generalization of tournaments called quasi-transitive digraphs (a superset of quasi-transitive oriented graphs studied in Chapter 5). Then the same author investigated the properties of 3-kings in quasi-transitive digraphs in [2]. And [5] generalized the results in [2] to  $(k + 1)$ -kings and  $k$ -quasi-transitive digraphs.

In [1], the author surveys various types of generalized tournaments, including semi-complete digraphs, oriented graphs and quasi-transitive digraphs. However, this article did not focus on the properties of kings in these generalizations of tournaments.

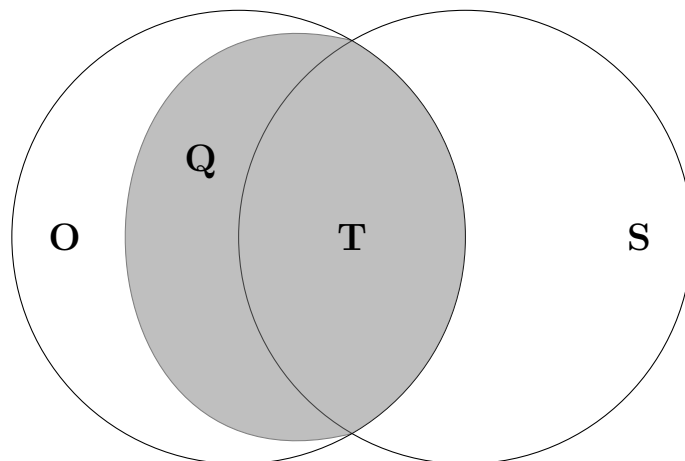


Figure 1.1: the relationship between tournaments, semi-complete digraphs, oriented graphs, and quasi-transitive oriented graphs.

## 1.2 Structure of This Thesis

The core content of this paper is in Chapters 2 to 5. The latter three chapters focus on kings in three generalizations of tournaments: semi-complete digraph (see Definition 3.2), oriented graph (see Definition 2.14), and quasi-transitive oriented graph (see Definition 5.1), respectively.

It is helpful to know the relation of these three generalizations of tournaments. In Fig. 1.1, we show the relationship between these graphs. The “O” represents the set of all oriented graphs; the “Q” represents the set of all quasi-transitive oriented graphs; the “T” represents the set of all tournaments; the “S” represents the set of all semi-complete digraphs. Tournaments are a subset of all three generalized tournaments. Quasi-transitive oriented graphs are a subset of oriented graphs, and the intersection between oriented graphs and semi-complete digraphs is exactly all the tournaments (hence the name “semi-complete oriented graph”, see Definition 2.15).

In Chapter 2, we discuss some of the terminology and previously-proven theorems that are used in other chapters. Chapters 3 to 5 are all independent from each other. We recommend the reader start with Chapter 2, and then choose the chapter that interests them the most.

Chapters 3 and 4 are two relatively easy chapters, compared to Chapter 5. These two chapters mainly focus on constructing  $(n, k)$  semi-complete digraphs and  $(n, k)$

oriented graph (see Definition 2.18).

Chapter 5 is the most interesting chapter. Unfortunately, it is harder than Chapters 3 and 4. This chapter is more proof-heavy than the previous two chapters. In this chapter we prove the four most important results in this thesis; namely Corollaries 5.21 and 5.22 (these 2 results can be merged into one), Theorems 5.24 and 5.29.

We provide a list of figure (see Page 55) and a index (see Page 58) at the end of the thesis, to help the reader quickly navigate this thesis.



# Chapter 2

## Background

### 2.1 Directed Graph

**Definition 2.1.** a *directed graph*, *digraph* or *graph* consists of a vertex set  $V(G)$ , and an edge set (ordered pair of vertices)  $E(G)$ .

For example, in Fig. 2.1, if we call this digraph  $G$ , then the vertex set or  $V(G)$  is  $\{a, b, c, d, e\}$ , and the edge set or  $E(G)$  is  $\{(a, a), (a, b), (b, c), (c, e), (e, a), (e, c)\}$ . We sometimes refer to “edges” as “arrows”.

### 2.2 Beating Relations

To simplify the notation, we think of an edge as a *beating relation*:

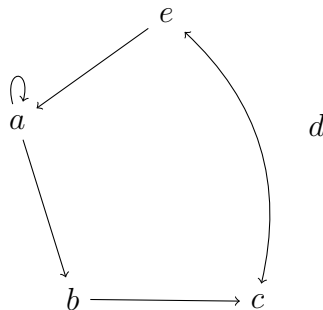


Figure 2.1: example of a directed graph.

**Definition 2.2.** In a directed graph  $G$ , if  $(a, b) \in E(G)$  then we say  $a$  **beats (dominates)**  $b$  or  $a \rightarrow b$ .

**Definition 2.3.** A **path** or a **walk** from vertex  $a_0$  to vertex  $a_n$  is a sequence of vertices  $[a_0, a_1, \dots, a_{n-1}, a_n]$  such that for all  $0 \leq k < n$ ,  $a_k \rightarrow a_{k+1}$ . We will sometimes write this path as  $a_0 \rightarrow a_1 \rightarrow \dots \rightarrow a_{n-1} \rightarrow a_n$ .

**Definition 2.4.** For a given path  $P$ , if the sequence has  $n + 1$  vertices, then we say the **length of path**  $P$  is  $n$ . A special case is a path consisting of one single vertex, which is a path of length 0.

The length of a path measures the number of edges on this path. For example, in Fig. 2.1, there is a path  $P = e \rightarrow a \rightarrow b \rightarrow c$ . And there are edges  $(e, a)$ ,  $(a, b)$ ,  $(b, c)$  on this path, therefore the length of this path is 3. There is another path  $P' = e \rightarrow c$  goes from  $e$  to  $c$ . This path  $P'$  has length 1. Therefore, path  $P'$  is shorter than path  $P$ , and if we go through every possible path from  $e$  to  $c$ , we can find out that  $P'$  is the *shortest path* from  $e$  to  $c$ .

**Definition 2.5.** In a directed graph  $G$ ,  $a$  **beats (dominates)  $b$  by  $n$  steps**, if the *shortest path* from  $a$  to  $b$  has length  $n$ .

In Fig. 2.1,  $e \rightarrow a \rightarrow b$ , and  $e$  does not beat  $b$  by 1 step, therefore  $e$  beats  $b$  by 2 steps.

However, although  $e \rightarrow a \rightarrow b \rightarrow c$ ,  $e$  does not beat  $c$  by 3 steps, because the shortest path from  $e$  to  $c$  is  $e \rightarrow c$ . Therefore,  $e$  beats  $c$  by 1 step.

**Definition 2.6.** In a directed graph, vertex  $a$  is **adjacent** to vertex  $b$  if  $a \rightarrow b$  or  $b \rightarrow a$  or both.

**Definition 2.7.** In a directed graph, for two distinct vertices  $a, b$ , if  $a$  is not adjacent to  $b$ , then  $a$  **ties**  $b$ .

In Fig. 2.1,  $c$  is adjacent to  $e$ ;  $a$  is adjacent to  $b$ ;  $b$  is adjacent to  $c$ . Whereas,  $c$  ties  $a$ , because there is no edge  $(c, a)$  and no edge  $(a, c)$  in this digraph.

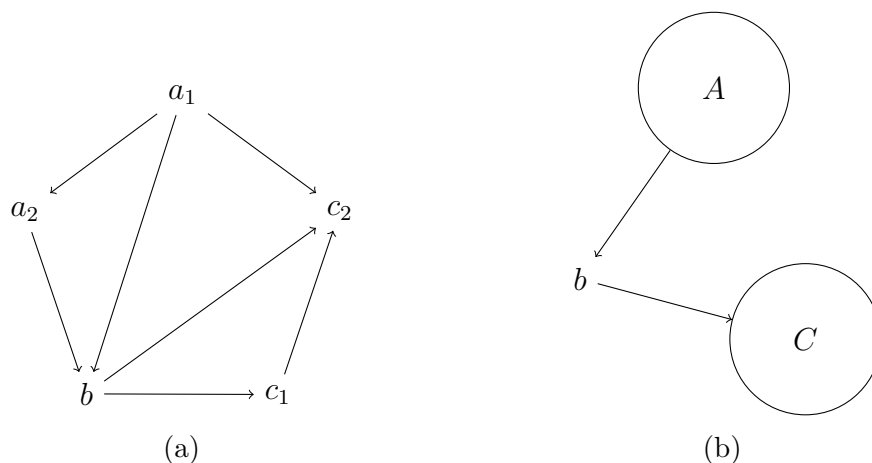


Figure 2.2: we can draw graph (a) as graph (b).

Since we don't always know the exact structure of the graph, it is sometimes useful to look at the beating relationships between sets of vertices. We then define beating, adjacency, and tie between vertex sets.

**Definition 2.8.** In digraph  $G$ , we write  $A \rightarrow B$  where  $A$  and  $B$  are disjoint subsets of  $V(G)$ , when every vertex in  $A$  beats every vertex in  $B$ . We also write  $A \rightarrow b$  as a shorthand for  $A \rightarrow \{b\}$ , and  $a \rightarrow B$  as a shorthand for  $\{a\} \rightarrow B$ .

**Definition 2.9.** In digraph  $G$ ,  $A$  is adjacent to  $B$  where  $A$  and  $B$  are disjoint subsets of  $V(G)$ , when every vertex in  $A$  is adjacent to every vertex in  $B$ . We also write “ $A$  is adjacent to  $b$ ” as a shorthand for “ $A$  is adjacent to  $\{b\}$ ”, and “ $a$  is adjacent to  $B$ ” as a shorthand for “ $\{a\}$  is adjacent to  $B$ ”.

**Definition 2.10.** In digraph  $G$ ,  $A$  ties  $B$  where  $A$  and  $B$  are disjoint subsets of  $V(G)$ , when every vertex in  $A$  ties every vertex in  $B$ . We also write “ $A$  ties  $b$ ” as a shorthand for “ $A$  ties  $\{b\}$ ” and “ $a$  ties  $B$ ” as a shorthand for “ $\{a\}$  ties  $B$ ”.

In Fig. 2.2, we show how we draw the beating relationship with subsets  $A = \{a_1, a_2\}$  and  $C = \{c_1, c_2\}$ . Notice, in Fig. 2.2b, set  $A$  and set  $C$  are not adjacent. This *does not* mean that set  $A$  ties set  $C$ ; it only means that we have not drawn out the relationship between these 2 sets in Fig. 2.2b.

We draw adjacency between sets (for example, set  $A$  is adjacent to set  $\{b\}$ ) using a solid edge without arrow. We draw tie between sets (for example, set  $A$  ties set  $C$ )

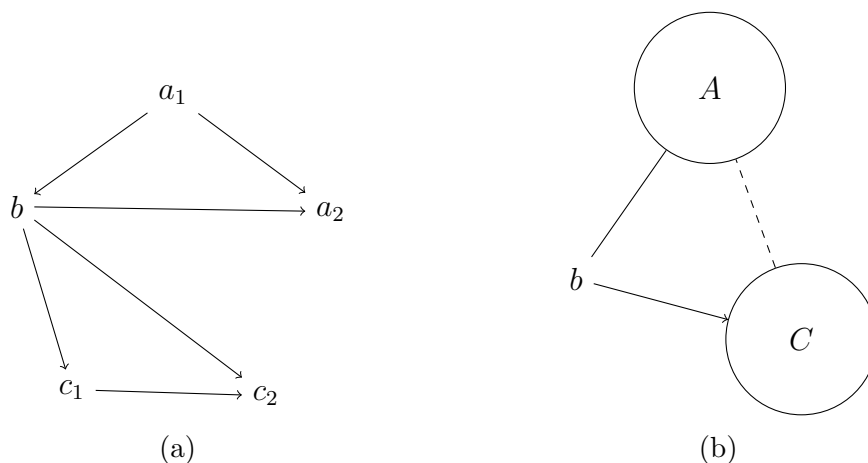
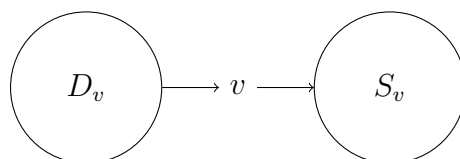


Figure 2.3: we can draw graph (a) as graph (b).

Figure 2.4:  $D_v$  is the dominant set of  $v$ ;  $S_v$  is the submissive set of  $v$ 

using a dashed edge without arrow. (see Fig. 2.3)

**Definition 2.11.** The **submissive set (dominant set)** of a vertex  $v$  in graph  $G$  is the set of vertices in  $G$  that are beaten by  $v$  (beat  $v$ ), formally it is  $\{e \in V(G) \mid v \rightarrow e\}$  (or  $\{e \in V(G) \mid e \rightarrow v\}$ ). We will denote the submissive set of  $v$  as  $S_v$ , and the dominant set of  $v$  as  $D_v$ . (See Fig. 2.4)

**Definition 2.12.** **Out-degree (in-degree)** of a vertex  $v$  in graph  $G$  is the size of the submissive set (dominant set) of  $v$ .

For example, in Fig. 2.1,  $S_a = \{a, b\}$ , therefore vertex  $a$  has out-degree 2;  $D_a = \{e, a\}$ , therefore vertex  $a$  has in-degree 2. Vertex  $d$  has in-degree 0, and out-degree 0 because  $d$  is not adjacent to any other vertex in the graph. Therefore,  $D_d$  and  $S_d$  are both empty sets.

Notice it is very natural to relate out-degree and in-degree with the “power” or “strength” of a vertex. However, we will discuss in later sections and chapters that a vertex with larger out-degree does not necessarily have more power, by our definition of

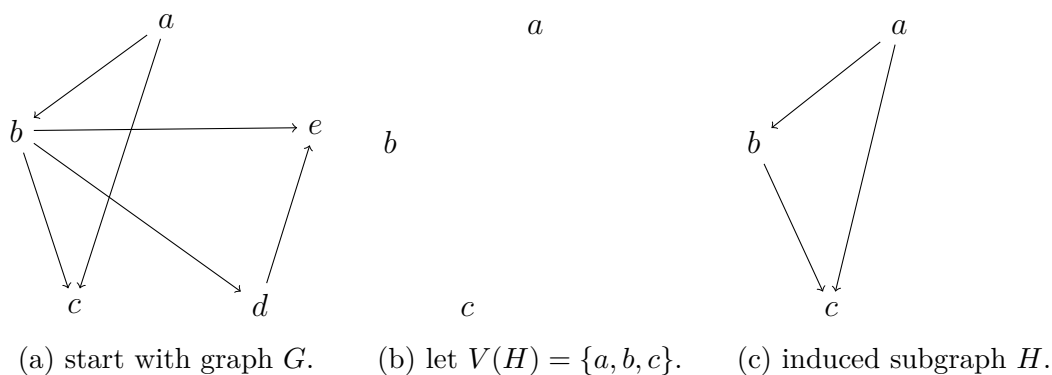


Figure 2.5: the process to create an induced subgraph.

“king” (defined in Section 2.4), However, the property that “vertices with largest out-degree are kings” is true in some nice families of graphs (see Section 2.4 and Section 3.2).

**Definition 2.13.** An *induced subgraph* or *vertex-induced subgraph*  $H$  of digraph  $G$  is a subgraph of  $G$  such that: for all pairs of vertices  $a, b \in V(H)$ , if  $(a, b) \in E(G)$ , then  $(a, b) \in E(H)$ .

In Fig. 2.5, we show the process of creating an induced subgraph. We redraw the vertices in  $V(H)$ , that is  $a, b, c$ , and then simply copy the edges between those vertices, that is,  $(a, b)$ ,  $(b, c)$ ,  $(a, c)$  from  $G$  to get  $H$ .

This definition can be viewed in 2 ways:

- Take the vertices  $a, b, c$  together with the edges between them to form subgraph  $H$ .
- Take away vertices  $e, d$  together with all the edges having  $e$  or  $d$  as an end point, the rest of the graph is the subgraph  $H$ .

For all the families of graphs discussed in this paper (oriented graph, semi-complete digraph, tournament, quasi-transitive oriented graph), induced subgraphs preserve the property of the original graph.

## 2.3 Oriented Graph and Tournament

**Definition 2.14.** An *oriented graph* is a digraph, such that:

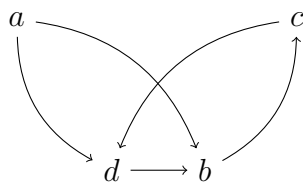


Figure 2.6: example of an oriented graph.

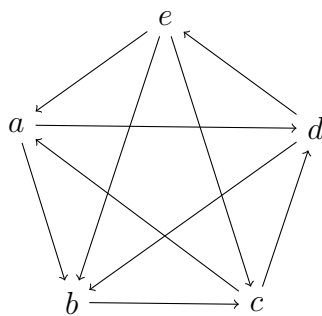


Figure 2.7: example of a tournament.

1. for all vertices  $a$ ,  $a$  does not beat itself.
2. for all pairs of adjacent vertices  $a, b$ , if  $a$  beats  $b$ , then  $b$  does not beat  $a$ .

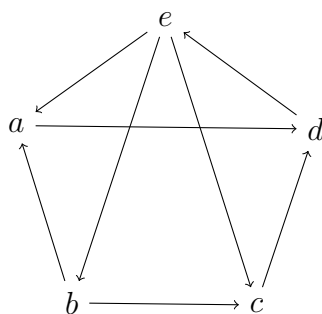
We show an oriented graph in Fig. 2.6. This definition of oriented graph can be viewed as a mathematical model of a round-robin tournament with ties allowed. A team cannot compete with itself, therefore property 1 holds; each pair of teams  $a$  and  $b$  competes exactly once, either  $a$  beats  $b$ ,  $b$  beats  $a$ , or there is a tie between  $a$  and  $b$ . therefore property 2 holds.

Therefore, it is very natural to view an edge going from  $a$  to  $b$  as  $a$  “beats”  $b$ , and two non-adjacent vertices  $a, b$  as a tie between  $a$  and  $b$ .

**Definition 2.15.** A **tournament** is an oriented graph without ties.

We give an example of a tournament in Fig. 2.7. A tournament can be interpreted as, what we call in the real world, a round-robin tournament, where every pair of vertices compete and the competition results in exactly one winner and one loser.

In [6], the author uses tournament to model a flock of chicken, between every two chickens there is a “pecking relation” (we call it the “beating relation”). Given two chickens, one chicken has to peck the other chicken, or be pecked by the other chicken, but not both.

Figure 2.8: an example with kings  $e$  and  $d$ .

## 2.4 Kings

**Definition 2.16.** a *king* in a digraph is a vertex that beats every other vertex by 1 or 2 steps.

In Fig. 2.8, the kings in this graph are vertices  $e$  and  $d$ . Vertex  $e$  beats  $a, b, c$  by one step, and beats  $d$  by 2 steps, so  $e$  is a king. Vertex  $d$  beats  $e$  by one step, and  $e$  beats  $a, b, c$ , so  $d$  beats  $a, b, c$  by 2 steps, which makes  $d$  a king.

The other vertices ( $a, b, c$ ) are not kings, because  $a$  cannot beat  $c$  by one or two steps ( $a \rightarrow d \rightarrow e \rightarrow c$  is a shortest path);  $b$  cannot beat  $e$  by one or two steps ( $b \rightarrow c \rightarrow d \rightarrow e$  is a shortest path);  $c$  cannot beat  $b$  by one or two steps ( $c \rightarrow d \rightarrow e \rightarrow b$  is a shortest path).

There are two points to note in this example:

- Fig. 2.8 shows that larger out-degree does not associate with more “power” or “strength”. Vertex  $d$  only has out-degree 1, and  $d$  is a king. Whereas vertex  $b$  has out-degree 2, but  $b$  is not a king. Vertex  $d$  is more “powerful” than  $b$ , even though  $d$  has smaller out-degree.
  - A vertex with the *highest* out-degree in a digraph may not be a king. In Fig. 2.9, vertex  $a$  has out-degree 3, which is the highest out-degree in this graph. However, it is not a king, since it cannot beat  $k$  by 1 or 2 steps
  - Also, a vertex with *lowest* out-degree in a digraph may be a king. In Fig. 2.9, the vertex  $k$  has the smallest out-degree in the graph, however it is the only king in this graph.

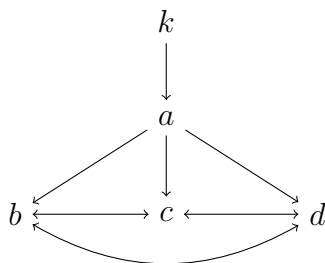
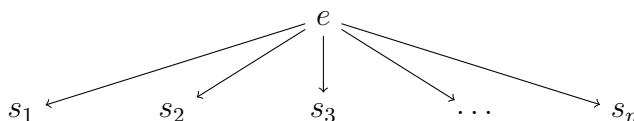


Figure 2.9: the vertex with smallest out-degree is the only king.

Figure 2.10: vertex  $e$  is the emperor of this graph.

- Fig. 2.8 shows that a digraph may have more than one king. In fact, [6] proved the probability that every vertex in a tournament is a king approaches 1 as the number of vertices in the tournament approaches  $\infty$ .

**Definition 2.17.** An **emperor** in a digraph is a vertex that beats every other vertex. (see Fig. 2.10)

**Definition 2.18.** an  $(n, k)$  **digraph** ( $(n, k)$  **oriented graph**,  $(n, k)$  **tournament**, etc.) is a digraph (oriented graph, tournament, etc.) with  $n$  vertices and  $k$  kings.

The following work has been done on kings in tournaments. These theorems will be useful when we investigate properties of kings in generalized tournaments in later chapters.

**Theorem 2.1.** A tournament has only one king if and only if that king is an emperor. [6]

**Theorem 2.2.** A tournament cannot have exactly 2 kings. [6]

**Theorem 2.3.** In a tournament, a vertex with largest out-degree is a king. [6]

**Corollary 2.4.** In a tournament, there always exists at least 1 king. [6]

**Theorem 2.5.** for all integers  $n \geq k \geq 1$ ,  $(n, k)$ -tournaments exist with the following exceptions:  $(n, 2)$  with any  $n \geq 2$  and  $(4, 4)$ . [6]



# Chapter 3

## Semi-complete Digraph

### 3.1 Definitions

In Chapter 2, we introduced one way to model “tie” as the non-adjacency between 2 vertices. However, if we think of “beating” as a weak ordering (like “subset” relation), then we can define “double tie” to capture this idea:

**Definition 3.1.** *There exists a **double tie** between vertices  $a$  and  $b$  if  $a \rightarrow b$  and  $b \rightarrow a$ .*

With the definition of double tie, we can then formalize another model of round-robin tournament with double ties. This kind of graph is called “semi-complete digraph”.

**Definition 3.2.** *A **semi-complete digraph** is a digraph where every vertex is adjacent to every other vertex, but does not beat itself. see Fig. 3.1*

This definition says that a semi-complete digraph is a digraph such that between each pair of vertices, there exists at least one edge between them.

**Definition 3.3.** *In a semi-complete digraph, if  $a \rightarrow b$  and  $b$  does not beat  $a$ , then vertex  $a$  **strictly beats** vertex  $b$ .*

Tournaments are a special case of semi-complete digraphs. In tournaments, for each pair of vertices  $\{a, b\}$ ,  $a$  strictly beats  $b$  or  $b$  strictly beaten by  $a$ . We can view

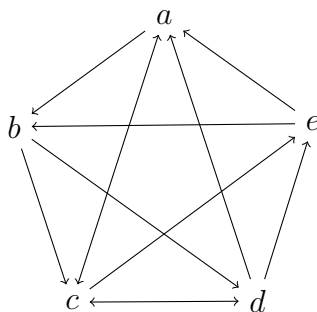


Figure 3.1: an example of semi-complete digraph.

a tournament as a semi-complete digraph that only has strict beatings; no double tie allowed.

**Definition 3.4.** *In a semi-complete digraph, the **double tie set of vertex  $v$**  is the set of vertices that double ties  $v$ . We denote the double tie set of  $v$  as  $DT_v$ .*

For example, in Fig. 3.1, there is a double tie between  $a$  and  $c$ , and another double tie between  $c$  and  $d$ . Every other beating relation between any other pair of vertices are “strict beating” relations. For example,  $a$  strictly beats  $b$ ,  $d$  strictly beats  $e$ , and  $b$  strictly beats  $d$ .

For vertex  $c$ ,  $DT_c$  (double tie set of  $c$ ) is  $\{a, d\}$ . The set of vertices that strictly beat  $c$  is  $\{b\}$ , which can be expressed as  $D_c - DT_c$ . The set of vertices that are strictly beaten by  $c$  is  $\{e\}$ , which can be expressed as  $S_c - DT_c$ .

## 3.2 Properties

Although tournaments are a special case of semi-complete digraphs, many useful properties of tournaments are also true for semi-complete digraphs.

**Lemma 3.1.** *For every vertex  $v$  in any semi-complete digraph  $G$ ,  $\{S_v - DT_v, DT_v, D_v - DT_v, \{v\}\}$  forms a partition of  $G$ .*

*Proof.*  $D_v - DT_v$  is the set of vertices that strictly beat  $v$ ,  $S_v - DT_v$  is the set of vertices that are strictly beaten by  $v$ ,  $DT_v$  is all the vertices that double tie with  $v$ . They are clearly disjoint by definition, and all of them are disjoint to  $\{v\}$  by definition.

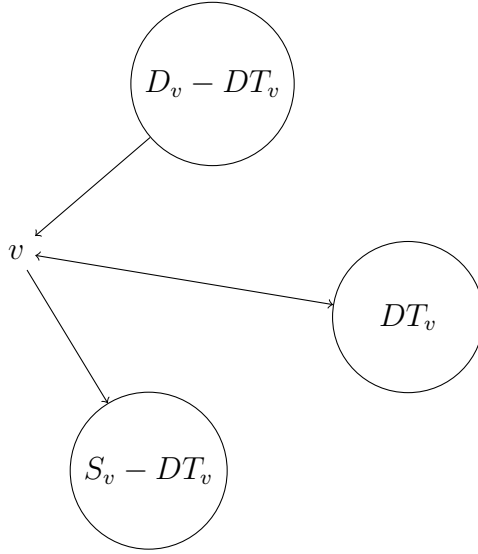


Figure 3.2: illustration of Lemma 3.1.

Every vertex in  $V(G)$  needs to be in one of the sets  $S_v - DT_v, DT_v, D_v - DT_v, \{v\}$ , because every other vertex needs to be adjacent to  $v$ .  $\square$

See Fig. 3.2 for a visualization for the proof of Lemma 3.1.

Lemma 3.1 and Fig. 3.2 are useful in proofs of many properties of semi-complete digraphs.

When we move on to general oriented graphs, the lack of this property in oriented graphs will make the structure of oriented graphs harder to work with.

**Theorem 3.2.** *All vertices with the maximum out-degree in a semi-complete digraph are kings.*

*Proof.* Let  $v$  be a vertex with the maximum out-degree in a semi-complete digraph  $G$ .

Suppose  $v$  is not a king in  $G$ . Since  $v$  is not a king and  $v$  beats every vertex in  $S_v$  by exactly 1 step, then by Lemma 3.1, there exists a vertex  $d \in D_v - DT_v = V(G) - S_v - \{v\}$  that is not beaten by  $v$  by 1 or 2 steps. Therefore there cannot be any vertex in  $S_v$  that beats  $d$ . Because  $d$  is adjacent to every vertex in  $S_v$ ,  $d$  strictly dominates  $v$  and  $S_v$ . Hence  $|S_d| \geq 1 + |S_v|$ . Therefore  $|S_d| > |S_v|$ , that is,  $d$  has larger out-degree than  $v$ . Contradiction.  $\square$

Fig. 3.3 shows that there can be more than one vertex with maximum out-degree. In

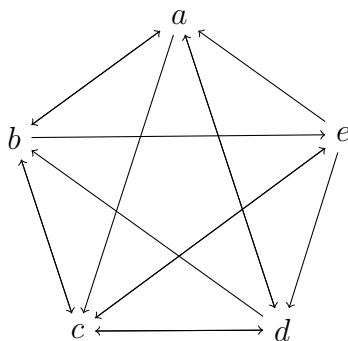


Figure 3.3: every vertex has the same out-degree.

fact, in this graph, every vertex has the same out-degree 3, therefore all of the vertices have the maximum out-degree. By Theorem 3.2, all vertices in this graph are kings.

**Corollary 3.3.** *For any non-empty semi-complete digraph  $G$ , there exists at least one king.*

*Proof.* This corollary is a result of Theorem 3.2. because out-degrees are non-negative integers, and  $V(G)$  is not empty. Thus there exists at least one vertex with maximum out degree.  $\square$

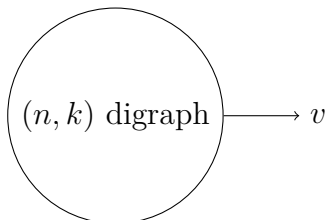
**Theorem 3.4.** *In a semi-complete digraph  $G$ , every vertex with a non-empty dominant set is beaten by a king.*

*Proof.* Let  $v$  be a vertex in  $G$ , such that  $D_v$  is not empty. Consider the subgraph induced by the vertices in  $D_v$ ; this subgraph is also a semi-complete digraph. By Corollary 3.3, there is a king  $k$  in the induced subgraph of  $D_v$ . We will show that  $k$  is also a king of  $G$ .

- $k$  dominates  $D_v$  by 1 or 2 steps, because  $k$  is a king in  $D_v$ .
- $k$  dominates  $v$  by exactly 1 step, because  $k$  is in  $D_v$ .
- $k$  dominates  $DT_v$  and  $S_v$  by 1 or 2 steps, because  $k$  dominates  $v$ , which beats all vertices in  $DT_v$  and  $S_v$ .

Then, by Lemma 3.1, we know that  $k$  dominates every vertex in the graph by 1 or 2 steps. Therefore  $k$  is a king in  $G$  and  $k$  beats  $v$ .  $\square$

$$a \leftrightarrow b$$

Figure 3.4: both vertices  $a$  and  $b$  are emperors and kings.Figure 3.5: an  $(n + 1, k)$  semi-complete digraph by adding a new vertex  $v$ 

**Corollary 3.5.** *If a semi-complete digraph has only one king, then that king is an emperor.*

*Proof.* Suppose  $G$  is a semi-complete digraph with only one king  $k$ , and  $k$  is not an emperor.

Then there exists  $v$  in the graph, such that  $k$  does not beat  $v$ . Therefore,  $v \rightarrow k$ , since there is no tie in the graph. Therefore,  $k$  has a non-empty dominant set. Then by Theorem 3.4,  $k$  is beaten by another king. Therefore there exists more than one king. Contradiction.  $\square$

Notice, unlike in tournaments, the converse of Corollary 3.5 is not true. In Fig. 3.4, we show we can have more than one king with some kings being emperors.

**Theorem 3.6.**  *$(n, k)$  semi-complete digraphs exist for all  $n \geq k \geq 1$ , where  $n, k$  are integers.*

*Proof.* We prove this theorem by induction. Construct an  $(n, k)$  semi-complete digraph:

- When there is only one vertex in the graph then the graph is a  $(1, 1)$  semi-complete digraph.
- See Fig. 3.5. We can add one vertex that is not a king by adding a vertex that is strictly beaten by all the vertices in the original graph. In other words, we can construct a  $(n + 1, k)$  semi-complete digraph from any  $(n, k)$  semi-complete digraph.

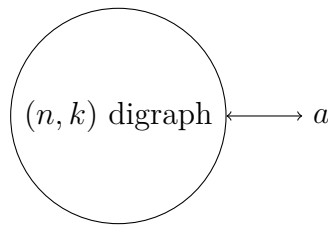


Figure 3.6: an  $(n + 1, k + 1)$  semi-complete digraph by adding a new king  $a$ .

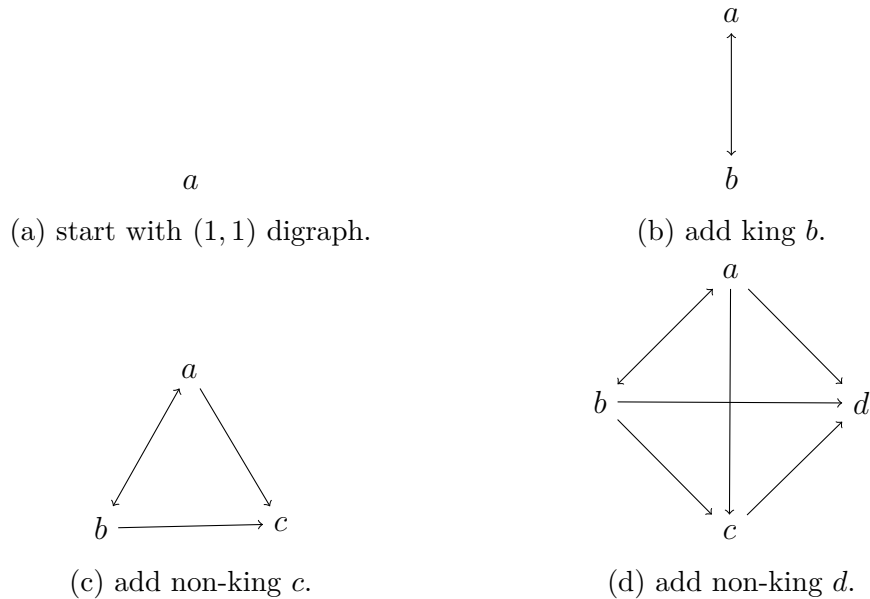


Figure 3.7: to construct a  $(4, 2)$  semi-complete digraph.

- See Fig. 3.6. We can add one king by adding one vertex that double ties all the vertices in the original graph. In other words, we can construct an  $(n + 1, k + 1)$  semi-complete digraph from any  $(n, k)$  semi-complete digraph.

Therefore, we can obtain any  $(n, k)$  semi-complete digraph by: starting with a  $(1, 1)$  semi-complete digraph, first add  $k - 1$  kings to get a  $(k, k)$  semi-complete digraph; then add  $n - k$  non-king vertices to get an  $(n, k)$  semi-complete digraph.  $\square$

In Fig. 3.7, we give an example of how to construct a  $(4, 2)$  semi-complete digraph using the inductive algorithm we introduced in Theorem 3.6:

1. Start with a single vertex, which is a  $(1, 1)$  semi-complete digraph.
2. Add a king  $b$ , by letting it double tie with  $a$  which gives us a  $(2, 2)$  semi-complete digraph.

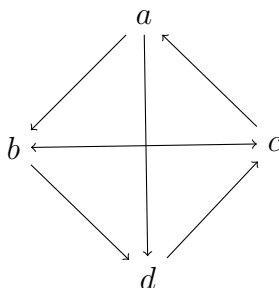


Figure 3.8: a  $(4, 4)$  semi-complete digraph with only one double tie

3. Add a non-king  $c$ , by letting every vertex in the  $(2, 2)$  semi-complete digraph beat  $c$ . which gives us a  $(3, 2)$  semi-complete digraph.
4. Add a non-king  $d$ , by letting every vertex in the  $(3, 2)$  semi-complete digraph beat  $d$ , which gives us a  $(4, 2)$  semi-complete digraph.

Using the method in Theorem 3.6, when we add the  $i_{th}$  king, we are creating  $(i - 1)$  ties, so adding  $k$  kings would require a total of  $1 + 2 + \dots + (k - 1)$  double ties. The summation simplifies to  $\frac{k(k-1)}{2}$ . Thus, the number of double ties blows up with the increase of  $k$  (number of kings).

Therefore we want to investigate how to use double ties more “efficiently”: what is the minimum number of double ties needed to get an  $(n, k)$  semi-complete digraph.

**Lemma 3.7.** *All  $(n, 2)$  semi-complete digraphs with  $n \geq 2$  can be constructed with only one double tie.*

*Proof.* use the method described in Theorem 3.6.

First add 1 king to a  $(1, 1)$  semi-complete digraph to get a  $(2, 2)$  semi-complete digraph (one double tie added). Then add  $(n - 2)$  non-king vertices to get an  $(n, 2)$  semi-complete digraph (no double tie added).

Therefore we add only one double tie to construct any  $(n, 2)$  semi-complete digraph. □

**Lemma 3.8.** *There exists a  $(4, 4)$  semi-complete digraph with only one double tie.*

*Proof.* Fig. 3.8 gives an example of a  $(4, 4)$  semi-complete digraph with only one double tie. The only double tie is between  $b$  and  $c$ . □

**Theorem 3.9.** *for all  $n \geq k \geq 1$ , there exists an  $(n, k)$  semi-complete digraph with at most one double tie.*

*Proof.* Recall Theorem 2.5, there exists an  $(n, k)$  tournament with the exception of  $(n, 2)$  and  $(4, 4)$ . Because tournaments are also semi-complete digraphs, therefore we can use Theorem 2.5 along with Lemma 3.7 and Lemma 3.8 to get the result.

The proofs of Theorem 2.5, Lemma 3.7, and Lemma 3.8 all provide constructions for these semi-complete digraphs. Therefore we can construct an  $(n, k)$  semi-complete digraph with at most one double tie, for all  $n \geq k \geq 1$ . □



# Chapter 4

## Oriented Graph

The properties of semi-complete digraphs basically inherit the properties of tournaments, and most of the proofs are almost the same.

We now move on to another family of graphs that are less similar to tournaments: oriented graphs. We will investigate general properties of oriented graphs in this chapter. In Chapter 5, we will focus on specific families of oriented graphs.

Recall Definition 2.14, an oriented graph is a digraph that doesn't have self-loops (a vertex beats itself) and double ties.

**Theorem 4.1.** *If there is an emperor  $k$  in an oriented graph  $G$ , then  $k$  is the only king in the graph.*

*Proof.* The emperor  $k$  beats every vertex in one step, therefore it is a king.

Because there is no double ties in an oriented graph, if  $k$  is an emperor, then no vertex in  $G$  can beat  $k$ , hence no vertex can beat  $k$  by one or two steps. Therefore,  $k$  is the only king in the graph.  $\square$

**Corollary 4.2.** *In an oriented graph with  $n$  vertices, if there exists a vertex with out-degree  $n - 1$ , then there is only one king in the graph.*

*Proof.* A vertex with out-degree  $n - 1$  beats every vertex in the graph except itself. Therefore that vertex is an emperor, and because of Theorem 4.1, there will be only one king in this graph.  $\square$



Figure 4.1: all the oriented graph with 2 vertices.

**Lemma 4.3.** *For an oriented graph  $G$ , if we add a new edge to  $G$  to get  $G'$ , the kings in  $G$  remain kings in  $G'$ .*

*Proof.* Need to show that for every king  $k$  in  $G$ ,  $k$  is also a king in  $G'$ . We can see that for every pair of vertices  $a, b$  if  $a \rightarrow b$  in  $G$ , then  $a \rightarrow b$  in  $G'$ , since we are not removing any edges.

- If  $k$  beats a vertex  $v$  by one step in  $G$ , then  $k \rightarrow v$  in  $G$ , therefore  $k \rightarrow v$  in  $G'$
- If  $k$  beats a vertex  $v$  by 2 steps in  $G$ , then exists vertex  $a$  such that  $k \rightarrow a \rightarrow v$  in  $G$ , then  $k \rightarrow a \rightarrow v$  in  $G'$  and  $k$  beats  $v$  by 2 steps in  $G'$ .

Therefore,  $k$  is also a king in  $G'$ , then every king in  $G$  is preserved in  $G'$ . □

It follows from Lemma 4.3 that adding an edge to oriented graph  $G$  cannot decrease the number of kings.

**Lemma 4.4.** *There does not exist a  $(1, 0)$  oriented graph.*

*Proof.* If any oriented graph has only 1 vertex, then that vertex by definition is a king. Therefore, there does not exist a  $(1, 0)$  oriented graph. □

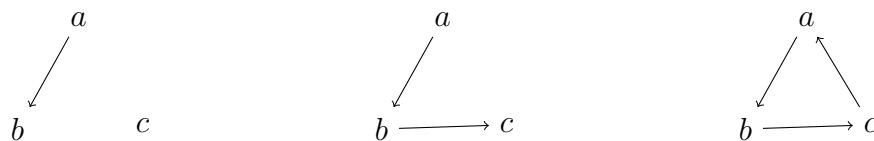
**Lemma 4.5.** *There does not exist a  $(2, 2)$  oriented graph*

*Proof.* Fig. 4.1 shows all the possible oriented graphs with 2 vertices. Notice, the oriented graph in Fig. 4.1a has 0 king, and the oriented graph in Fig. 4.1b has only 1 king. Therefore, there does not exist a  $(2, 2)$  oriented graph. □

**Lemma 4.6.** *There does not exist a  $(3, 2)$  oriented graph*

*Proof.* We consider all the oriented graphs with 3 vertices.

First, consider the graph with maximum out-degree 0. The graph with maximum out-degree 0 will have no edge, therefore there cannot be a king, since no vertex can beat other vertices by 2 steps.



(a) 3 vertices with 1 edge. (b) 3 vertices with 2 edges. (c) 3 vertices with 3 edges.

Figure 4.2: all the oriented graphs with 3 vertices and maximum out-degree 1.

Then, consider the graph with maximum out-degree 1. In Fig. 4.2, We show every oriented graph with 3 vertices and maximum out-degree 1. In Fig. 4.2a, there is no king; in Fig. 4.2b, there is 1 king; in Fig. 4.2c, every vertex is a king, therefore it has 3 kings.

Finally, we consider the oriented graphs with maximum out-degree 2. By Corollary 4.2, because we have 3 vertices in the graph, and at least one vertex has out-degree 2, there can only be 1 king.

Therefore, there is no  $(3, 2)$  oriented graph.  $\square$

**Lemma 4.7.** *There does not exist a  $(4, 4)$  oriented graph.*

*Proof.* Assume there exists a  $(4, 4)$  oriented graph. When we add edges to this  $(4, 4)$  oriented graph:

- the number of kings cannot increase, since a graph cannot have more kings than vertices.
- the number of kings cannot decrease, because of Lemma 4.3

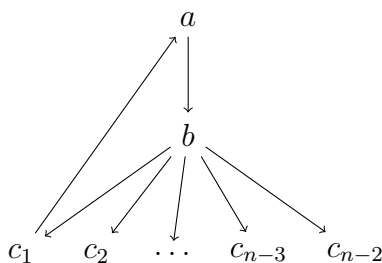
Then we can keep adding edges until every pair of vertices are adjacent and get a  $(4, 4)$  tournament. By Theorem 2.5, there do not exist  $(4, 4)$  tournaments.

Therefore, there cannot exist  $(4, 4)$  oriented graphs.  $\square$

We have shown that  $(1, 0)$ ,  $(2, 2)$ ,  $(3, 2)$  and  $(4, 4)$  oriented graphs do not exist. These four cases turn out to be the only exceptions.

**Lemma 4.8.** *There exists an  $(n, 2)$  oriented graph for  $n \geq 4$ .*

*Proof.* We can see in Fig. 4.3 that  $c_1$  cannot dominate  $c_2$  by 1 or 2 steps, and other  $c_i (i \neq 1)$  cannot dominate any other vertex, because the out-degree of these vertices are 0.

Figure 4.3: only  $a$  and  $b$  are kings for  $n \geq 4$ .

$a$  is a king because  $a \rightarrow b$  and  $b$  beats every  $c_i$ , therefore  $a$  beats  $b$  by one step, and  $a$  beats every  $c_i$  by 2 steps.  $b$  beats every  $c_i$  by 1 step, and  $b$  beats  $a$  by 2 steps:  $b \rightarrow c_1 \rightarrow a$ .  $\square$

**Lemma 4.9.** *There exists  $(n, 0)$  oriented graph for  $n \geq 0$  except  $n = 1$*

*Proof.* For an oriented graph with  $n$  vertices and no edge, every vertex cannot beat any other vertex. Therefore, the graph has 0 kings and the graph is a  $(n, 0)$  oriented graph.  $\square$

**Theorem 4.10.** *There exists an  $(n, k)$  oriented graph for all  $n \geq k \geq 0$ , with the exception of  $(1, 0)$ ,  $(2, 2)$ ,  $(3, 2)$ , and  $(4, 4)$  oriented graph.*

*Proof.* Theorem 2.5 shows that there exists an  $(n, k)$  tournament for all  $n \geq k \geq 1$  with the exception of  $(n, 2)$ , and  $(4, 4)$ .

Because tournaments are also oriented graphs and by Lemmas 4.4 to 4.7, Lemma 4.8, and Lemma 4.9, exceptions listed above are the only exceptions.  $\square$

We generalized the result on tournaments from [6] to oriented graphs and show that there are only 4  $(n, k)$  oriented graphs that do not exist.

Following the idea from Chapter 3, one of the questions to ask is how can we use ties more “efficiently”. The construction method in the proof of Lemma 4.8 is very inefficient.

Here we present a “better” way to construct these oriented graphs that uses at most one tie.

**Lemma 4.11.** *There exists an  $(n, 2)$  oriented graph with only one tie, for  $n \geq 4$ .*

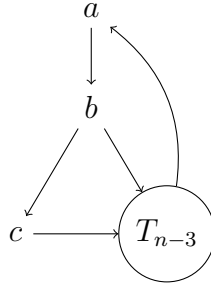


Figure 4.4: the constructive proof for Lemma 4.11



Figure 4.5:  $(2, 0)$  oriented graph.

*Proof.* See Fig. 4.4,  $T_{n-3}$  is a tournament of  $n - 3$  vertices. In this graph, the only tie is between  $a$  and  $c$  and the only kings are  $a$  and  $b$ .

$a$  is a king because,  $a \rightarrow b \rightarrow c$  and  $a \rightarrow b \rightarrow T_{n-3}$ , therefore  $a$  beats  $b$  by 1 step and  $a$  beats  $c$  and  $T_{n-3}$  by 2 steps.  $b$  is a king because,  $b \rightarrow c$  and  $b \rightarrow T_{n-3} \rightarrow a$  (because  $T_{n-3}$  is not empty), therefore  $b$  beats  $c$  and  $T_{n-3}$ , and  $b$  beats  $a$  by 2 steps.

$c$  is not a king, the shortest path from  $c$  to  $b$  is  $c \rightarrow T_{n-3} \rightarrow a \rightarrow b$ . Any vertex  $v$  in  $T_{n-3}$  cannot be a king, because the shortest path between  $v$  and  $c$  is  $v \rightarrow a \rightarrow b \rightarrow c$  which has length 3. □

**Lemma 4.12.** *There exists  $(n, 0)$  oriented graph where  $n \neq 1$  with at most 1 tie.*

*Proof.* First, we can see that a  $(0, 0)$  oriented graph exists, it is just an empty graph with no vertex and edge.

Then, in Fig. 4.5 we show that a  $(2, 0)$  oriented graph exists with one tie: it is just 2 vertices and no edge between them.

In Fig. 4.6, we give a way to construct an  $(n + 1, 0)$  oriented graph from an  $(n, 0)$  oriented graph. We denote the  $(n, 0)$  oriented graph as  $G$ .

We need to show the resulting graph in Fig. 4.6 has no king: first,  $s$  cannot be a king, because it does not beat any vertex. We then show any vertex  $v \in V(G)$  is not a king. Because  $v$  is not a king in  $G$ , there exists a vertex  $v' \in V(G)$  such that  $v$  cannot beat  $v'$  by 1 or 2 steps in  $G$ . Then  $v$  still cannot beat  $v'$  in this new graph, because  $s$  is the only added vertex, and the path  $v \rightarrow s \rightarrow v'$  do not exist. □

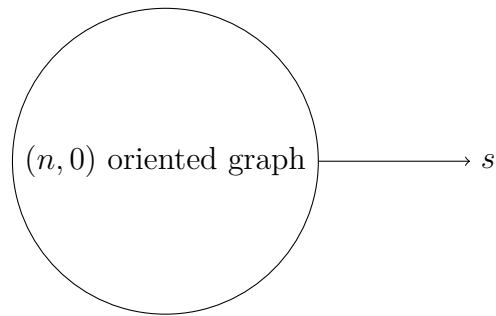


Figure 4.6: construct  $(n + 1, 0)$  oriented graph from  $(n, 0)$  oriented graph.

With Lemma 4.11 and Lemma 4.12, we can prove the following theorem.

**Theorem 4.13.** *There exists an  $(n, k)$  oriented graph with at most one tie for all  $n \geq k \geq 0$ , with the exception of  $(1, 0)$ ,  $(2, 2)$ ,  $(3, 2)$ , and  $(4, 4)$  oriented graphs.*

*Proof.* Almost the same proof as Theorem 4.10, just substitute Lemma 4.8, Lemma 4.9 with Lemma 4.11 and Lemma 4.12, respectively.  $\square$

In other words, Theorem 4.13 states that for all  $(n, k)$  oriented graphs that exist, they can be constructed with only one tie.

# Chapter 5

## Quasi-transitive Oriented Graph

### 5.1 Definitions

As mentioned in Chapter 4, in an oriented graph, for any pair of vertices  $a, b$ , either  $a$  beats  $b$ ,  $b$  beats  $a$  or  $a$  ties  $b$ . So, there are  $3^{\binom{n}{2}}$  different oriented graphs on  $n$  vertices. With no other restrictions, it is difficult to produce theorems about them. Therefore we decided to focus on another special case of oriented graph, which is called “quasi-transitive oriented graph”:

**Definition 5.1.** *a quasi-transitive oriented graph  $G$  is an oriented graph such that, for all vertices  $a, b, c \in V(G)$ , if  $a \rightarrow b \rightarrow c$ , then  $a$  is adjacent to  $c$ . See Fig. 5.1*

Because every vertex is adjacent to every other vertex in a tournament, tournaments is a special case of quasi-transitive oriented graph.

Although tournaments are just a small subset of quasi-transitive oriented graphs, and many quasi-transitive oriented graph are very different from tournaments (for ex-



Figure 5.1: if  $a \rightarrow b \rightarrow c$  then  $a$  is adjacent to  $c$ .

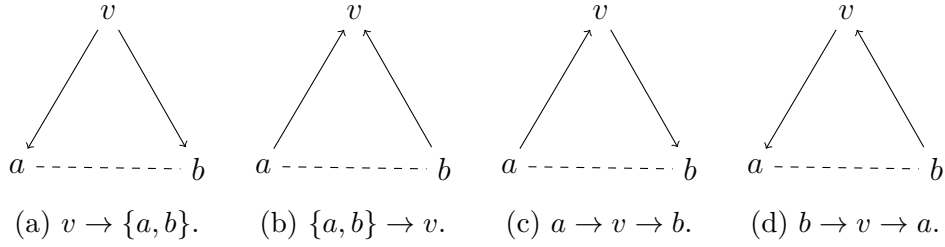


Figure 5.2: all the possible beating relations between vertex  $v$  and vertices  $a, b$

ample, every graph with no edge is a quasi-transitive oriented graph), quasi-transitive oriented graphs inherit many properties of tournaments [3].

## 5.2 Ties and Tie Paths

Properties of ties in quasi-transitive digraphs are vastly important in this chapter.

**Lemma 5.1.** *In a quasi-transitive oriented graph  $G$ , if vertex  $a$  ties vertex  $b$ , then for each vertex  $v$  that is adjacent to both  $a$  and  $b$ , either  $v$  beats both  $a$  and  $b$ , or  $v$  is beaten by both  $a$  and  $b$  ( $\{a, b\} \rightarrow v$  or  $v \rightarrow \{a, b\}$ ).*

*Proof.* Because  $v$  is adjacent to both  $a$  and  $b$ , we show all the possible beating relationships between  $v$  and  $a, b$  in Fig. 5.2. Because  $G$  is a quasi-transitive oriented graph, if  $a \rightarrow v \rightarrow b$  or  $b \rightarrow v \rightarrow a$  then  $a, b$  have to be adjacent. Therefore, the only possible beating relationships are  $\{a, b\} \rightarrow v$  or  $v \rightarrow \{a, b\}$ .  $\square$

Fig. 5.3a shows a nice visualization of Lemma 5.1. Consider the set of vertices that are adjacent to both  $a$  and  $b$ . In this figure the set is  $\{d_1, d_2, d_3, s_1, s_2, s_3, s_4\}$ . This set is partitioned into 2 parts: those vertices that dominate both  $a$  and  $b$ :  $\{d_1, d_2, d_3\}$ , and those vertices that are beaten by both  $a$  and  $b$ :  $\{s_1, s_2, s_3, s_4\}$  The behavior of  $\{a, b\}$  is somewhat like a vertex, see Fig. 5.3b.

However, this nice visualization in Fig. 5.3 has a very strong prerequisite, that is, all the vertices need to be adjacent to both  $a$  and  $b$ . What will happen when we add another vertex that ties either  $a$  or  $b$  and is adjacent to  $v$ ?

In Fig. 5.4a we add a vertex  $c$  that ties  $b$  and is adjacent to  $v$  (we use an edge without arrow to denote adjacency). Because  $b, c$  are both adjacent to  $v$ , and  $c$  ties



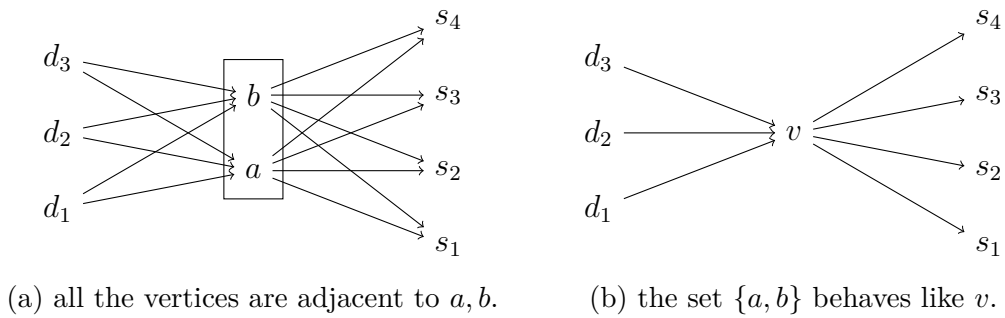


Figure 5.3: only look at all the vertices adjacent to both  $a$  and  $b$ , then  $\{a, b\}$  behaves like a vertex.

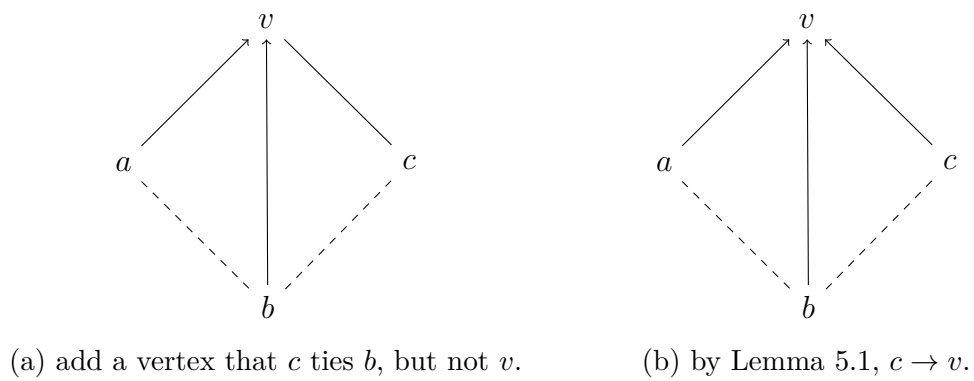


Figure 5.4: ties are “transmitting” arrow directions.

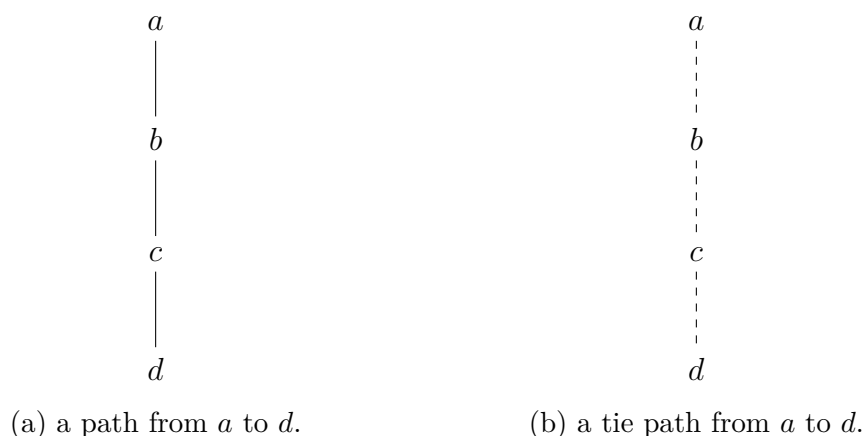


Figure 5.5: tie paths are just like paths.

$b$ , therefore either  $\{b, c\} \rightarrow v$  or  $v \rightarrow \{b, c\}$ . However, in the graph,  $b \rightarrow v$ , therefore  $c \rightarrow v$ . Thus, we get Fig. 5.4b.

We find out that because of the ties between  $a, b, c$ , the arrow direction between  $v$  and  $a$  and the arrow direction between  $v$  and  $b$ , got “transmitted” to be the arrow direction between  $v$  and  $c$  via Lemma 5.1. We formalize this “arrow transmission” idea:

**Definition 5.2.** *In a digraph, a **tie path** from vertex  $a_0$  to vertex  $a_n$ , or a tie path between vertices  $a_0$  and  $a_n$ , is a sequence of vertices  $[a_0, a_1, a_2, \dots, a_{n-1}, a_n]$ , such that for all  $0 \leq k < n$ ,  $a_k$  ties  $a_{k+1}$ . If there are  $n + 1$  vertices in that sequence, we will say **the length of the tie path** is  $n$ . Note that all the  $a_i$ ’s do not have to be distinct.*

Notice that every vertex  $a$  has a tie path  $[a]$  to itself with length 0. A tie path is similar to a path (called a “walk” in [8]) in an undirected graph; just substitute all the edges in a path with ties. See Fig. 5.5.

Tie paths are useful in analyzing quasi-transitive oriented graphs:

**Lemma 5.2.** *In a digraph, if there exists a tie path from vertex  $a$  to vertex  $b$  and there exists a tie path from vertex  $b$  to vertex  $c$ , then there exists a tie path from  $a$  to  $c$ .*

*Proof.* Write the tie path from  $a$  to  $b$  as:  $[a, a_0, a_1, \dots, a_n, b]$ , and the tie path from  $b$  to  $c$  as:  $[b, b_0, b_1, \dots, b_m, c]$ . Then there exists a tie path  $[a, a_0, a_1, \dots, a_n, b, b_0, b_1, \dots, b_m, c]$  from  $a$  to  $c$ . See Fig. 5.6. □

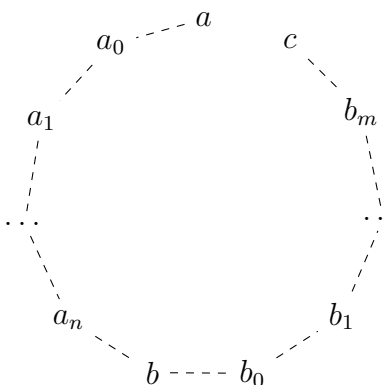


Figure 5.6: tie path from  $a$  to  $b$ , and from  $b$  to  $c$ .

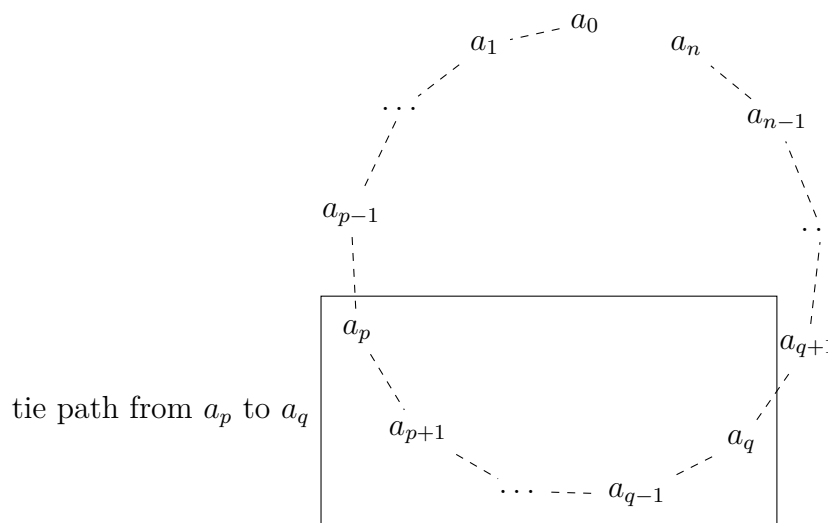


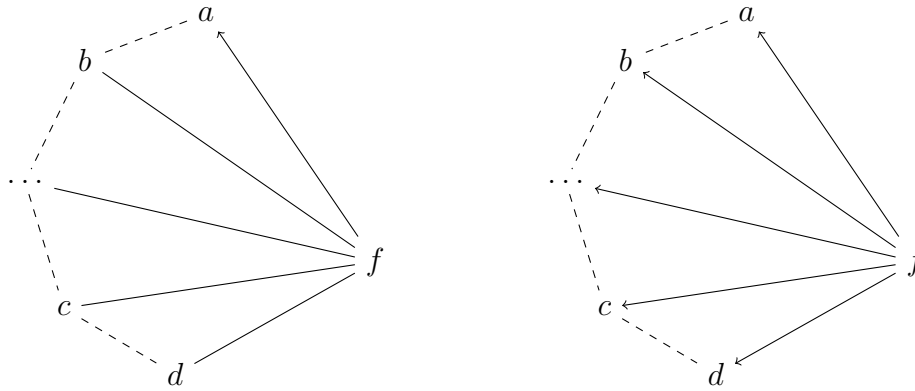
Figure 5.7: the tie path from  $a_0$  to  $a_n$ .

**Lemma 5.3.** *In a digraph, if there exists a tie path  $[a_0, a_1, \dots, a_{n-1}, a_n]$ , then for all  $p \neq q$   $0 \leq p \leq n$  and  $0 \leq q \leq n$ , there exists a tie path between  $p$  and  $q$ .*

*Proof.* Without loss of generality, assume  $p < q$ . Then we can find a tie path  $[a_p, a_{p+1}, \dots, a_{q-1}, a_q]$ , see Fig. 5.7. □

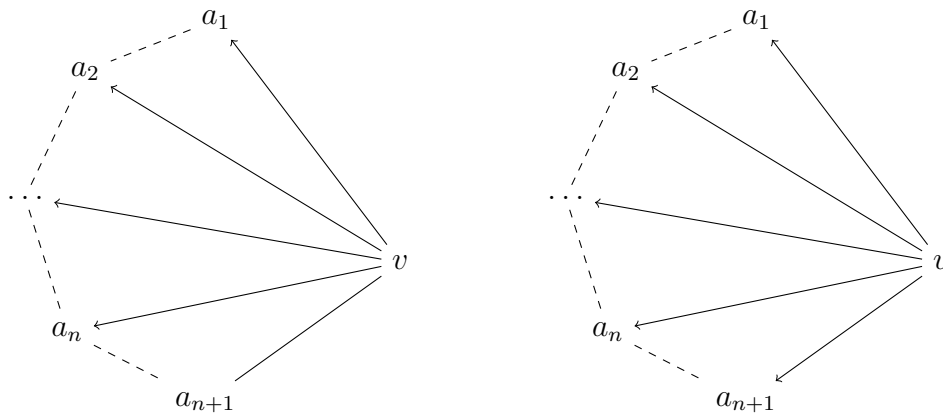
After looking at Fig. 5.4, we can hypothesize that tie paths can transmit arrow directions across the whole path in quasi-transitive oriented graphs. See Fig. 5.8.

**Lemma 5.4.** *For every tie path in a quasi-transitive oriented graph, if vertex  $v$  is adjacent to all the vertices on the tie path, then  $v$  beats every vertex on the tie path, or  $v$  is beaten by every vertex on the tie path.*



(a)  $a$  is beaten by  $f$ , and other vertices adjacent to  $f$       (b) the arrow direction is “transmitted” through the tie path

Figure 5.8: tie path transmits the arrow direction.



(a)  $a_{n+1}$  is adjacent to  $v$ .      (b) then  $a_{n+1}$  has to be beaten by  $v$ .

Figure 5.9: the induction step of the proof of Lemma 5.4

*Proof.* We can prove this lemma via induction, following the construction in Fig. 5.4.

Start with a tie path of length 1, then we have  $a_0$  ties  $a_1$  in this tie path, and  $v$  is adjacent to both of them. Lemma 5.1 proves the result for path of length 1.

Assume the property holds for any tie path of length  $n$  and vertex  $v$ , then we need to prove this property holds for any tie path of length  $n + 1$ . See Fig. 5.9, take a tie path  $[a_0, a_1, \dots, a_n, a_{n+1}]$ . we can apply the induction hypothesis to the tie path  $[a_0, a_1, \dots, a_n]$ . Because  $a_n$  ties  $a_{n+1}$ , and  $v$  is adjacent to both of them. By Lemma 5.1

- Case 1,  $v \rightarrow \{a_0, \dots, a_n\}$ , then  $v \rightarrow a_{n+1}$ , then  $v$  beats the whole tie path;
- Case 2,  $\{a_0, \dots, a_n\} \rightarrow v$ , then  $a_{n+1} \rightarrow v$ , then  $v$  is beaten by the whole tie path.

□

Lemma 5.4 generalizes Lemma 5.1, however, it is still far from elegant, because we still require that vertex  $v$  be adjacent to the whole tie path.

### 5.3 Tie Components

We continue to explore the relationship between tie paths and arrow direction transmitted by ties. We define the following structure with the intuition of “path” and “connected component” in an undirected graph [8].

**Definition 5.3.** *In a digraph  $G$ , a **tie component of vertex  $a$** , denoted  $C(a)$ , is the set of vertices  $v$  such that there exists a tie path between  $a$  and  $v$ .*

**Definition 5.4.** *In a digraph  $G$ , a tie component  $C(a)$  is a **trivial tie component** if  $C(a)$  only contains  $a$  itself and no other vertex.*

In Fig. 5.10, we first change every tie in Fig. 5.10a to an undirected edge and then remove all the directed edges to obtain Fig. 5.10b, and tie components in Fig. 5.10a exactly correspond to connected components in Fig. 5.10b. Therefore, just like a tie path is similar to a path in an undirected graph, a tie component is very similar to a connected component (called a “component” in [8]) in an undirected graph, and we can bring all the properties and intuitions of connected components to tie components.

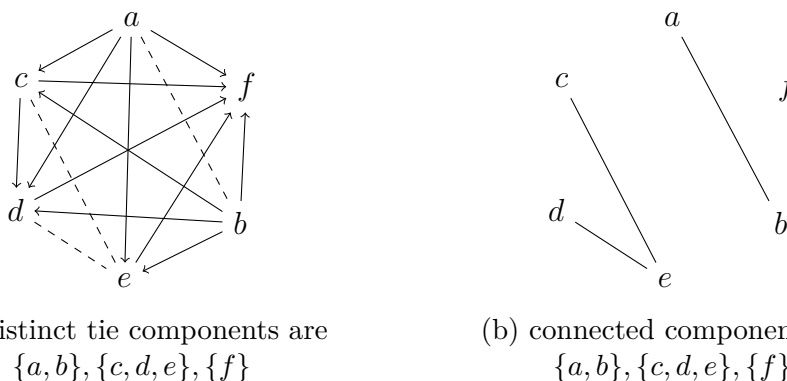


Figure 5.10: the tie components in (a) are the connected components in (b).

We prove results about tie components in digraphs. Notice that these results work in all digraphs, not just in quasi-transitive oriented graphs.

**Theorem 5.5.** *For a digraph  $G$ , we define a relation  $R$  on  $V(G)$  for vertices  $a, b \in V(G)$ ,  $a R b$  if and only if there exists a tie path between  $a$  and  $b$ .  $R$  is an equivalence relation.*

*Proof.* Reflexive: given a vertex  $a$ ,  $[a]$  is a tie path of length 0. therefore there exists a tie path from  $a$  to itself.

Symmetry: if there exists a tie path from  $a_0$  to  $a_n$ :  $[a_0, a_1, a_2, \dots, a_n]$ , then because ties do not have directions,  $[a_n, a_{n-1}, \dots, a_1, a_0]$  is a tie path from  $a_n$  to  $a_0$ .

Transitive: by Lemma 5.2. □

**Corollary 5.6.** *For a digraph  $G$ , a tie component is an equivalence class on  $V(G)$ .*

*Proof.* By Theorem 5.5, and the definition of tie component. □

**Corollary 5.7.** *For digraph  $G$ , tie components form a unique partition of  $V(G)$ .*

*Proof.* By Corollary 5.6, tie components form equivalence classes on  $V(G)$ , and equivalence classes form a unique partition of the set  $V(G)$  [4]. □

**Corollary 5.8.** *In a digraph, if vertex  $v$  is not in tie component  $C(a)$ , then  $v$  is adjacent to  $C(a)$ .*

*Proof.* Suppose  $v$  is not in  $C(a)$ . Because  $C(a)$  is an equivalence class on the relation  $R$  such that  $a R b$  if there exists a tie path between  $a$  and  $b$ . two equivalent class  $C(a)$

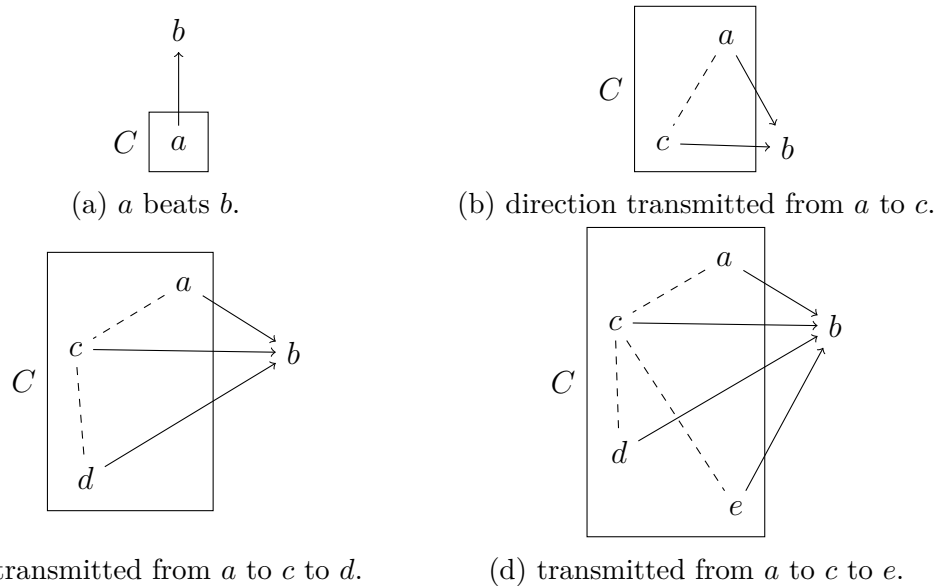


Figure 5.11: the arrow direction to  $b$  was transmitted in tie component  $C$ .

and  $C(v)$  disjoint [4]. Therefore there is no tie path between  $v$  and  $a$ , thus  $v$  does not tie any vertex in  $C(a)$ .

Therefore,  $v$  is adjacent to every vertex in  $C(a)$ . □

Our previous results that ties “transmit” arrow directions in quasi-transitive oriented graphs, had us see that for a tie component  $C$  and a vertex  $v \notin C$ ,  $C \rightarrow v$  or  $v \rightarrow C$ . See Fig. 5.11, for a visualization.

We then formalize and generalize this idea into two theorems:

**Lemma 5.9.** *In a quasi-transitive oriented graph, for any tie component  $C(a)$  and any vertex  $v$  such that  $v \notin C(a)$ , then either  $C(a) \rightarrow v$  or  $v \rightarrow C(a)$ .*

*Proof.* By Corollary 5.8, we know that  $v$  is adjacent to  $C(a)$ . Let  $x, y$  be two vertices in  $C(a)$ . There exists a tie path between them, so both  $x$  and  $y$  beat  $v$  or  $v$  beats both  $x, y$ . Since this is true for all pairs of vertices on  $C(a)$ , we get  $v \rightarrow C(a)$  or  $C(a) \rightarrow v$ . □

**Corollary 5.10.** *In a quasi-transitive oriented graph, for any tie component  $C(a)$ , If there exists one vertex  $v \in C(a)$  that beats (is beaten by) a vertex  $v' \notin C(a)$ . Then  $C(a) \rightarrow v'$  ( $v' \rightarrow C(a)$ ). See Fig. 5.12*

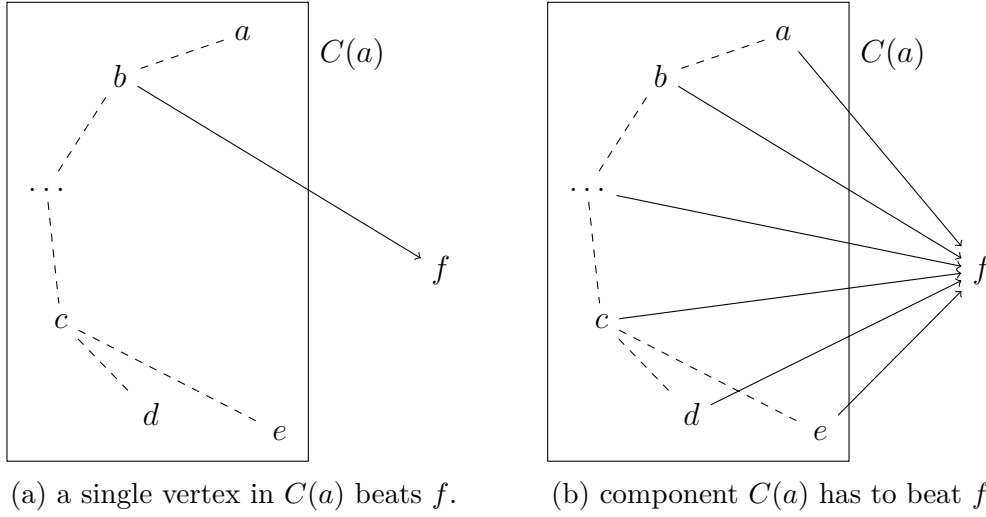


Figure 5.12: a vertex  $b$  beats  $f$  will force the component  $C(a)$  to beats  $f$ .

*Proof.* Because  $v' \notin C(a)$ , then either  $v' \rightarrow C(a)$  or  $C(a) \rightarrow v'$  by Lemma 5.9.

Case 1,  $v \rightarrow v'$ : because  $v \in C(a)$ , then  $v'$  cannot beat any vertex in  $C(a)$ . Therefore  $C(a) \rightarrow v'$ .

Case 2,  $v' \rightarrow v$ : because  $v \in C(a)$ , then  $v'$  cannot be beaten by all of  $C(a)$ . Therefore  $v' \rightarrow C(a)$ . □

To further generalize Lemma 5.9, we prove the following theorem:

**Theorem 5.11.** *For any two distinct tie components  $C(a)$  and  $C(b)$  in a quasi-transitive oriented graph,  $C(a) \rightarrow C(b)$  or  $C(b) \rightarrow C(a)$ .*

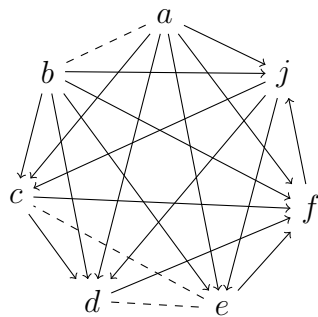
*Proof.* Take any vertex  $a'$  in  $C(a)$ . By Corollary 5.7,  $a'$  is not in  $C(b)$ . Then because of Lemma 5.9,  $a' \rightarrow C(b)$  or  $C(b) \rightarrow a'$ .

- Case 1,  $a' \rightarrow C(b)$ : take any element  $b' \in C(b)$ . By Corollary 5.10,  $C(a) \rightarrow b'$ . Therefore  $C(a)$  beats every vertex in  $C(b)$ , then  $C(a) \rightarrow C(b)$ .
- Case 2,  $C(b) \rightarrow a'$ : by the same reasoning,  $C(b) \rightarrow C(a)$ .

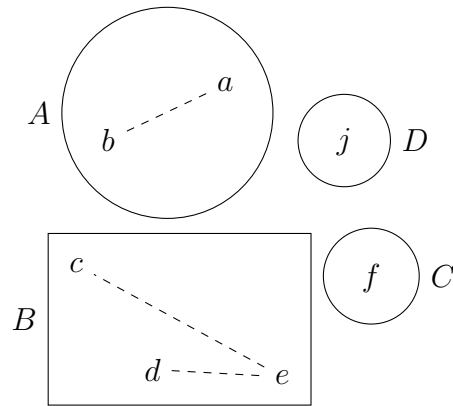
□

After several pages of theorems, we can finally take a pause and understand what we are saying here. We combine Corollary 5.7 and Theorem 5.11, and try to understand it visually.

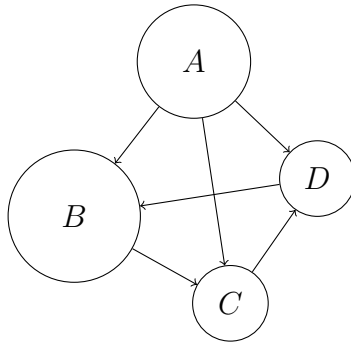




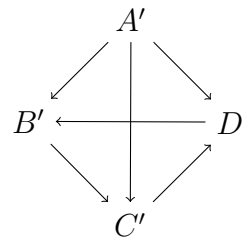
(a) a quasi-transitive oriented graph.



(b) find its tie components.



(c) components beat each other.



(d) components are just like vertices.

Figure 5.13: condense the tie components and get a tournament.

See Fig. 5.13. For every quasi-transitive oriented digraph, we can split it into tie components, and every tie component either beats another tie component or is beaten by another tie component. Thus, tie components behave like vertices. Because there are no ties between tie components, The graph formed by tie components is a tournament, which is the “most well understood class of directed graphs” [1].

## 5.4 Graph Condensations

**Definition 5.5.** A **condensation** is a function  $f : G \rightarrow H$ , where  $G$  and  $H$  are oriented graphs, and  $f$  maps  $V(G)$  to  $V(H)$  surjectively, such that

- for any two vertices  $a, b \in V(G)$ , if  $f(a) \rightarrow f(b)$  in  $H$  then  $a \rightarrow b$  in  $G$ .
- for any two vertices  $a, b \in V(G)$ , if  $f(a)$  ties  $f(b)$  in  $H$  then  $a$  ties  $b$  in  $G$ .

We call  $G$  the **uncondensed graph**, and  $H$  the **condensed graph**.

Two examples of condensations are identity condensation and trivial condensation.

Identity condensation  $i$  maps an oriented graph  $G$  to itself, such that  $i$  maps every vertex to itself, and  $f(a) \rightarrow f(b)$  if and only if  $a \rightarrow b$ . The identity condensation preserves all the edges and ties, and does not change the graph at all. We can see that this map is a condensation by definition.

Another example is trivial condensation  $t$ , which maps any oriented graph  $G$  into a single point. Because for any two vertices  $a, b \in G$ ,  $t(a) = t(b)$ , therefore by definition,  $t(a)$  does not tie or beat  $t(b)$ . Therefore, this mapping also satisfies the definition of condensation.

**Definition 5.6.** Given oriented graph  $G$  as the uncondensed graph of condensation  $f$ , A **component** of vertex  $a \in V(G)$  is the pre-image of  $f(a)$ , in other words, the component of  $a$  is  $\{x \in V(G) \mid f(x) = f(a)\}$ .

A component of vertex  $a$  is a set of vertices that are condensed into the same vertex as  $a$ . For example, the component of vertex  $a$  in oriented graph  $G$  with identity condensation is a set that only contains  $a$  itself, and the component of a vertex  $a$  in

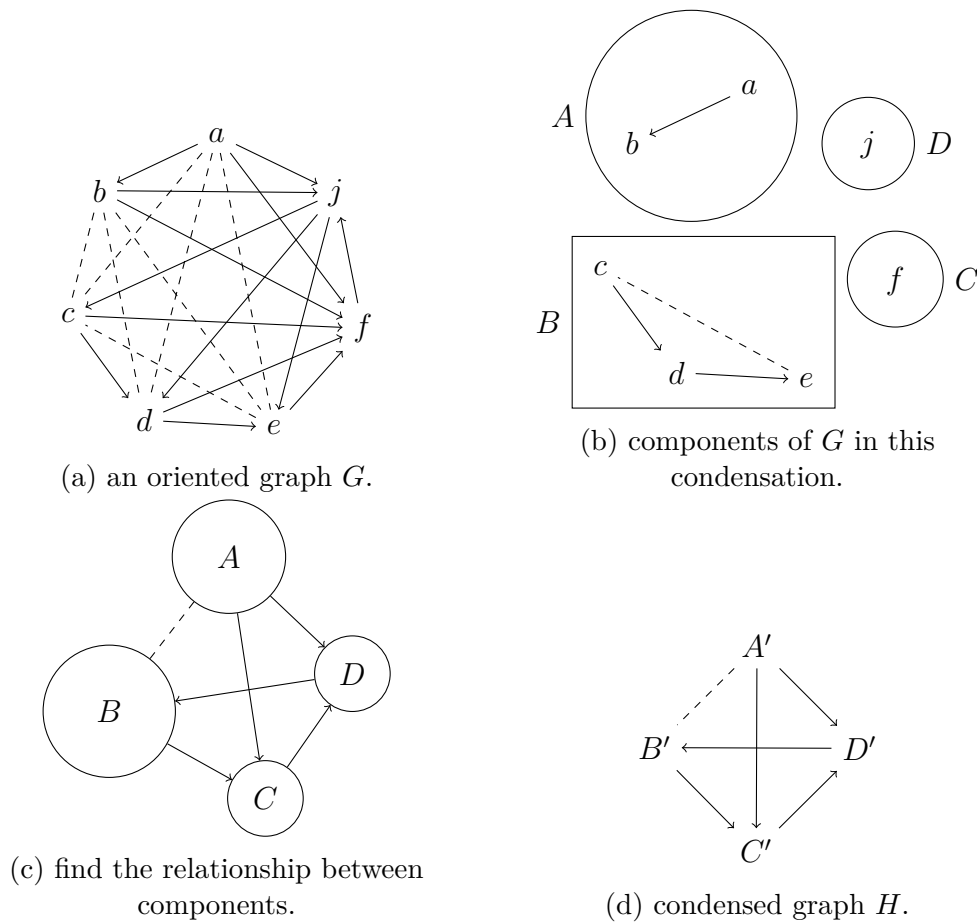


Figure 5.14: an example of graph condensation.

oriented graph  $G$  with the trivial condensation is the set of all the vertices, because all vertices in  $G$  are mapped onto a single vertex in  $H$ , therefore all the vertices in  $G$  are condensed into the same vertex as  $a$ .

In Fig. 5.14, we show an example of a condensation. In this example, the distinct components in  $G$  are  $A$ ,  $B$ ,  $C$  and  $D$ , and they are condensed into vertices  $A'$ ,  $B'$ ,  $C'$  and  $D'$  respectively. The component of  $a$  is  $A$ , the component of  $d$  is  $B$ , and the component of  $c$  is also  $B$ , because both  $c$  and  $d$  are mapped into vertex  $B'$  in oriented graph  $H$ .

We prove several theorems involving condensation and components to help us with our goal of identifying kings in quasi-transitive oriented graphs.

**Theorem 5.12.** *Given any oriented graph  $G$  and condensation  $f$ , distinct components in  $G$  partition  $V(G)$ .*

*Proof.* We show distinct components disjoint: Assume there exists two vertex  $a$  and  $b$ , such that the component of  $a$  is  $A$ ; component of  $b$  is  $B$ , and  $A \neq B$ ;  $A \cap B \neq \emptyset$ . Therefore, there exists  $v \in A \cap B$ . By definition of components,  $f(v) = f(a)$ , and  $f(v) = f(b)$ , therefore  $f(b) = f(a)$ , and  $B = \{x \in V(G) \mid f(x) = f(b)\} = \{x \in V(G) \mid f(x) = f(a)\} = A$ .  $A = B$ , contradiction.

We show every vertex is in a component: for every vertex  $v \in V(G)$ ,  $v$  is in the component of  $v$  because  $f(v) = f(v)$ .  $\square$

**Lemma 5.13.** *Given any oriented graph  $G$  and condensation  $f$ , for any two distinct components  $A$  and  $B$ ,*

- *for all  $a \in A, b \in B$ , if  $a$  ties  $b$ , then  $f(a)$  ties  $f(b)$ .*
- *for all  $a \in A, b \in B$ , if  $a \rightarrow b$ , then  $f(a) \rightarrow f(b)$*

*Proof.* Because  $A$  and  $B$  are distinct components, then by Theorem 5.12,  $A$  and  $B$  are disjoint, therefore for any vertex  $a \in A$ , and any vertex  $b \in B$ ,  $f(a) \neq f(b)$ .

Given vertex  $a \in A$  ties vertex  $b \in B$ . Then  $f(a)$  cannot beat  $f(b)$ , otherwise  $a$  should beat  $b$ , by definition of condensation. Also,  $f(b)$  cannot beat  $f(a)$ , otherwise  $b$  should beat  $a$ , therefore  $f(a)$  ties  $f(b)$ .

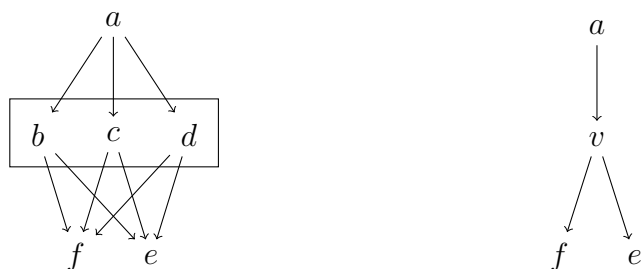
Given vertex  $a \in A$  beats vertex  $b \in B$ . Then  $f(a)$  cannot tie  $f(b)$ , otherwise  $a$  should tie  $b$ . Also  $f(b)$  cannot beat  $f(a)$ , otherwise  $b$  should beat  $a$ . Therefore  $f(a) \rightarrow f(b)$ .  $\square$

**Theorem 5.14.** *Given any oriented graph  $G$  and condensation  $f$ , for any two distinct components  $A$  and  $B$ ,*

- *if any vertex  $a \in A$  ties any vertex  $b \in B$ , then  $A$  ties  $B$ .*
- *if any vertex  $a \in A$  beats any vertex  $b \in B$ , then  $A \rightarrow B$ .*

*Proof.* Because  $A$  and  $B$  are distinct components, then by Theorem 5.12,  $A$  and  $B$  are disjoint, therefore for any vertex  $a \in A$ , and any vertex  $b \in B$ ,  $f(a) \neq f(b)$ .

Given vertex  $a \in A$  ties vertex  $b \in B$ , by Lemma 5.13,  $f(a)$  ties  $f(b)$ . Thus, for every vertex  $a' \in A$ , and every vertex  $b' \in B$ , because  $f(a') = f(a)$  and  $f(b') = f(b)$ ,



(a) every vertex either beats  $\{b, c, d\}$  or is beaten by them. (b) vertices  $\{b, c, d\}$  in Fig. 5.15a behaves exactly like  $v$  in this graph.

Figure 5.15:  $\{b, c, d\}$  behaves like a vertex.

and  $f(a)$  ties  $f(b)$ , therefore  $f(a')$  ties  $f(b')$ . By the definition of condensation,  $a'$  ties  $b'$ . Therefore,  $A$  ties  $B$ .

Given vertex  $a \in A$  beats vertex  $b \in B$ . by Lemma 5.13,  $f(a) \rightarrow f(b)$ . Thus for every vertex  $a' \in A$ , and every vertex  $b' \in B$ , because  $f(a') = f(a)$  and  $f(b') = f(b)$ , and  $f(a) \rightarrow f(b)$ , therefore  $f(a') \rightarrow f(b')$ . By the definition of condensation,  $a' \rightarrow b'$ . Therefore,  $A$  ties  $B$ .  $\square$

**Corollary 5.15.** *Given any oriented graph  $G$  and condensation  $f$ , for any two distinct components  $A$  and  $B$ , either  $A \rightarrow B$ , or  $B \rightarrow A$ , or  $A$  ties  $B$ .*

*Proof.* Because components are not empty (component of  $v$  always contains  $v$  itself), we can take  $a \in A$ , and  $b \in B$ . Because  $G$  is an oriented graph, then either  $a \rightarrow b$ ,  $b \rightarrow a$ , or  $a$  ties  $b$ .

According to Theorem 5.14, if  $a \rightarrow b$ , then  $A \rightarrow B$ ; if  $b \rightarrow a$ , then  $B \rightarrow A$ ; if  $a$  ties  $b$ , then  $A$  ties  $B$ .  $\square$

In Fig. 5.15, we show an example of a set of vertices  $\{b, c, d\}$  that behaves like the vertex  $v$ . We then try to visualize Corollary 5.15, with the help of Fig. 5.14, we find out that graph condensation just takes the idea in Fig. 5.15 to the next level: all the components behave like vertices. A condensation looks for those sets of vertices that behave like a single vertex (components), and then condenses them into a single vertex.

Notice that all oriented graphs have the identity condensation and trivial condensation, but some oriented graphs may not have other condensations defined on them.

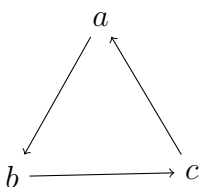
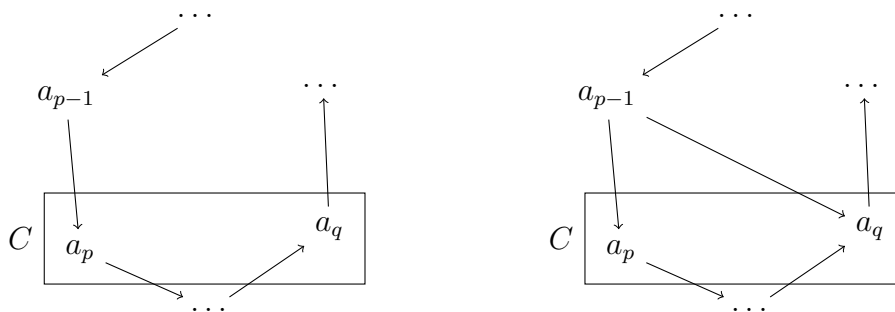


Figure 5.16: an oriented graph with only identity condensation and trivial condensation.



(a)  $a_p$  and  $a_q$  are in the same component. (b) by Theorem 5.14,  $a_{p-1} \rightarrow a_q$ .

Figure 5.17: the path took a “detour” at  $a_{p-1}$ .

For example, the graph in Fig. 5.16 cannot have any condensation defined on it except for identity condensation and trivial condensation, because we cannot find any component that is not the whole graph or a single vertex. For example, let’s look at set  $\{a, c\}$ . Vertex  $b$  beats a vertex in the set ( $b$  beats  $a$ ), and is beaten by a vertex in the set ( $c$  is beaten by  $b$ ). Therefore the set  $\{a, c\}$  is not a component.

Condensation is a relatively strong transformation, because it preserves many properties of the graph. We will prove some theorems that will be useful in later sections, when looking for kings

**Lemma 5.16.** *Let  $f : G \rightarrow H$  be a condensation of oriented graph  $G$  and let  $a_0, a_n$  be two vertices in  $G$  from different components. If  $P : a_0 \rightarrow a_1 \rightarrow \dots \rightarrow a_n$  is a shortest path from  $a_0$  to  $a_n$ , then vertices  $a_0, a_1, \dots, a_n$  are all in different components.*

*Proof.* Suppose not. Take the the smallest  $p$  such that there exist  $a_p$  and  $a_q (p < q \leq n)$  in the same component  $C$ .

Since  $a_p$  is the first vertex in  $P$  that belongs to the same component as another vertex in  $P$ .  $a_{p-1}$  cannot be in the same component as  $a_p$ , then  $a_{p-1} \notin C$ . Because  $a_{p-1} \rightarrow a_p$  and Theorem 5.14,  $a_{p-1} \rightarrow C$ . Thus  $a_{p-1} \rightarrow a_q$ , and path  $a_0 \rightarrow a_1 \rightarrow \dots \rightarrow$

$a_{p-2} \rightarrow a_{p-1} \rightarrow a_q \rightarrow a_{q+1} \rightarrow \cdots \rightarrow a_{n-1} \rightarrow a_n$  exists and is shorter than  $P$ . See Fig. 5.17. Contradiction.  $\square$

**Theorem 5.17.** *Given a condensation  $f : G \rightarrow H$ , on oriented graph  $G$ , for any  $a_0, a_n \in V(G)$ , such that  $a_0$  and  $a_n$  are in two distinct components. A shortest path from  $a_0$  to  $a_n$  is  $P_G : a_0 \rightarrow a_1 \rightarrow \cdots \rightarrow a_{n-1} \rightarrow a_n$  in  $G$  if and only if a shortest path from  $P_H : f(a_0)$  to  $f(a_n)$  is  $f(a_0) \rightarrow f(a_1) \rightarrow \cdots \rightarrow f(a_{n-1}) \rightarrow f(a_n)$  in  $H$ .*

*Proof.* ( $\Rightarrow$ ): Suppose  $P_G$  is a shortest path but  $P_H$  is not a shortest path in  $H$ . Then there exists a shorter path  $P'_H$  in  $H$  from  $f(a_0)$  to  $f(a_n)$ :  $f(a_0) \rightarrow f(b_1) \rightarrow f(b_2) \rightarrow \cdots \rightarrow f(b_{m-1}) \rightarrow f(b_m) \rightarrow f(a_n)$   $m < n - 1$ . Because  $a_0$  and  $a_n$  are in distinct components, the path  $P'_H$  does not have length 0. The path  $a_0 \rightarrow b_1 \rightarrow b_2 \rightarrow \cdots \rightarrow b_{m-1} \rightarrow b_m \rightarrow a_n$  exists (by definition of condensation) and is shorter than  $P_G$ , a contradiction.

( $\Leftarrow$ ): Suppose  $P_H$  is a shortest path in  $H$  but  $P_G$  is not a shortest path in  $G$ . Then there exists a shorter path  $P'_G$  in  $G$  from  $a_0$  to  $a_n$ :  $a_0 \rightarrow b_1 \rightarrow b_2 \rightarrow \cdots \rightarrow b_{m-1} \rightarrow b_m \rightarrow a_n$ ,  $m < n - 1$ . Because of Lemma 5.16,  $a_0, b_1, b_2, \dots, b_m, a_n$  are all in different components. Then by Lemma 5.13,  $f(a_0) \rightarrow f(b_1) \rightarrow f(b_2) \rightarrow \cdots \rightarrow f(b_m) \rightarrow f(a_n)$  exists and is shorter than  $P_H$ , contradiction.  $\square$

**Corollary 5.18.** *Let  $f : G \rightarrow H$  be a condensation of oriented graph  $G$ , for all  $a_0, a_n \in G$ , such that  $a_0$  and  $a_n$  are in two distinct components,  $a_0$  beats  $a_n$  by  $n$  steps in  $G$  if and only if  $f(a_0)$  beats  $f(a_n)$  by  $n$  steps in  $H$ .*

*Proof.* Result of Theorem 5.17 and definition of “beats by  $n$  steps”.  $\square$

**Corollary 5.19.** *Given a condensation  $f : G \rightarrow H$  of oriented graph  $G$ , vertex  $k$  is a king in  $G$  if and only if*

- $k$  is a king in the induced subgraph of the component of  $k$ .
- $f(k)$  is a king in  $H$ .

*Proof.* Denote component of  $k$  as  $C$ . Denote the induced subgraph of  $C$  as  $G_c$ .

Prove ( $\Rightarrow$ ): Suppose  $k$  is a king in  $G$ . Then  $k$  beats every vertex in  $V(G)$  by 1 or 2 steps. In particular,  $k$  beats every vertex outside of  $C$  by 1 or 2 steps. Hence, by

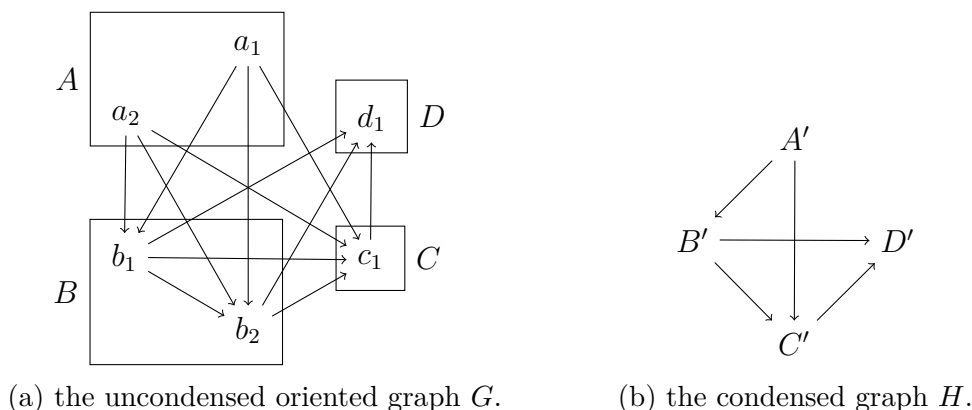


Figure 5.18: information lost during condensation.

Corollary 5.18,  $f(k)$  beats every vertex in  $H$  by 1 or 2 steps, Therefore  $f(k)$  is a king in  $H$ .

We know  $k$  beats every vertex in  $C$  by 1 or 2 steps in  $G$ . Let  $v \in C$  and  $v \neq k$ . We need to show  $k$  beats every vertex in  $G$  by 1 or 2 steps in  $G_C$ .

- Case 1:  $k$  beats  $v$  by 1 step, then  $k$  beats  $v$  by 1 step in  $G_c$ .
- Case 2:  $k$  beats  $v$  by 2 steps, then there exists  $a \in V(G)$  such that  $k \rightarrow a \rightarrow v$  in  $G$ . because  $v, k \in C$  and Theorem 5.14,  $a \in C$  ( $a \notin C$  causes a contradiction). Therefore,  $k$  beats  $v$  by 2 steps in  $G_c$ .

Thus,  $k$  is a king in  $G_c$ , and  $f(k)$  is a king in  $H$ .

( $\Leftarrow$ ): Suppose  $f(k)$  is a king in  $H$ , and  $k$  is a king in  $G_C$ . Since  $f(k)$  is a king in  $H$ ,  $f(k)$  beats every other vertex  $f(v)$  in  $H$  by 1 or 2 steps. Thus,  $k$  beats every other vertex that is not in  $C$  by 1 or 2 steps, by Corollary 5.18. Since  $k$  is a king in  $G_c$ ,  $k$  beats every vertex in  $C$  by 1 or 2 steps. Therefore,  $k$  beats every vertex in  $V(G)$  by 1 or 2 steps, so  $k$  is a king in  $G$ .  $\square$

Although condensation is a very strong transformation, some information in the graph does get lost via this transformation. For example, the beating relationships between all the vertices in the same components are lost.

In Fig. 5.18, we show an example of a condensation, where the vertices  $a_1, a_2$  are condensed into  $A'$ ;  $b_1, b_2$  are condensed into  $B'$ ;  $c_1$  is condensed into  $C'$ ; and  $d_1$  is condensed into  $D$ . If we just look at the condensed graph, we notice that we cannot



recreate the beating relationship between  $a_1$  and  $a_2$ , therefore the information about the beating relationship is lost. This fact is true for all the vertices in the same component. Another example in this graph is that we cannot know the beating relationship between  $b_1$  and  $b_2$  just by the condensed graph.

Notice that the beating relationship between components is not lost. For example, if we want to know the beating relationship between  $b_1$  and  $a_2$ , we first observe that  $A' \rightarrow B'$  in the condensed graph, then by definition of a condensation,  $a_2$  has to beat  $b_1$  in the uncondensed graph. Another example is that we can know that  $a_1$  ties  $d_1$  because  $A'$  ties  $D'$  in the condensed graph.

Therefore, an “efficient condensation” should keep as many vertices as possible. One of the most “efficient” condensations is the identity condensation, because it does not lose any information about this graph. However, the identity condensation does not mutate the graph at all, therefore it is not very practical.

**Definition 5.7.** *Given a set of condensations  $F = \{f_0, f_1, \dots, f_n\}$  where  $f_k : G_k \rightarrow H_k$  and all the  $V(G_k)$  are equal for  $0 \leq k \leq n$ , an **efficient condensation**  $f : G \rightarrow H$  in  $F$  is a condensation such that  $H$  has largest vertex set in  $\{H_0, H_1, H_2, \dots, H_n\}$*

Notice in this definition, all the  $G_k$ 's where  $0 \leq k \leq n$  are not necessarily distinct. One example of efficient condensation is given a graph  $G$  and all the condensation defined on  $G$ , that is  $\{f_0, f_1, \dots, f_n\}$  where  $f_k : G \rightarrow H_k$ . Then the identity condensation, is the only efficient condensation in this set, since it keeps all the vertices.

Later, we will restrict the types of graphs  $H$  can be. In so doing, the identity condensation will no longer be allowed.

## 5.5 Tie Component Condensations

In the previous section we talked about the general property of graph condensations. In this section, we will focus on a specific type of condensation mentioned in previous sections and Fig. 5.13. Recall a tie component is a set of vertices

**Definition 5.8.** *A **tie component condensation** is a function  $f : Q \rightarrow T$ , where  $Q$*

is a quasi-transitive oriented graph,  $T$  is a directed graph, and  $f$  maps  $V(Q)$  to  $V(T)$  surjectively, such that:

- if  $a, b \in V(Q)$  are in the same tie component, then  $f(a) = f(b)$ .
- if  $a, b \in V(Q)$  are in 2 distinct tie components  $A, B$  respectively, then
  - if  $A \rightarrow B$ , then  $a \rightarrow b$ .
  - if  $B \rightarrow A$ , then  $b \rightarrow a$ .

**Corollary 5.20.** *All tie component condensations are graph condensations, where the components of the uncondensed graphs are the tie components of the uncondensed graphs.*

**Corollary 5.21.** *For any quasi-transitive orientated graph  $Q$ , the tie component condensation  $f : Q \rightarrow T$  will always exist and  $T$  will always be a tournament.*

*Proof.* By Theorem 5.11 and the definition of tie component condensation, the tie component condensation always exists.

Also by Theorem 5.11 and the definition of tie component condensation, for any two distinct vertices  $f(a), f(b) \in T$ , either  $f(a) \rightarrow f(b)$  or  $f(b) \rightarrow f(a)$ . Therefore,  $T$  is a tournament.  $\square$

Fig. 5.13 is a great example of tie component condensation. In this condensation, the function  $g$  maps  $a, b$  to  $A'$ , maps  $c, d, e$  to  $B'$ , maps  $f$  to  $C'$ , and maps  $j$  to  $D'$ .

It is also helpful to note that the tie components  $C$  and  $D$  are trivial tie components, since they contain only vertex  $g$  and  $j$  respectively.

**Corollary 5.22.** *Given a quasi-transitive oriented graph  $Q$ , the tie component condensation  $f : Q \rightarrow T$  is unique.*

*Proof.* Tie component condensation exists by Corollary 5.21; tie component condensation is unique by Corollary 5.7.  $\square$

**Definition 5.9.** *For a given quasi-transitive oriented graph  $Q$ , we call the result of the tie component condensation of  $Q$ , the **underlying tournament** of  $Q$ .*

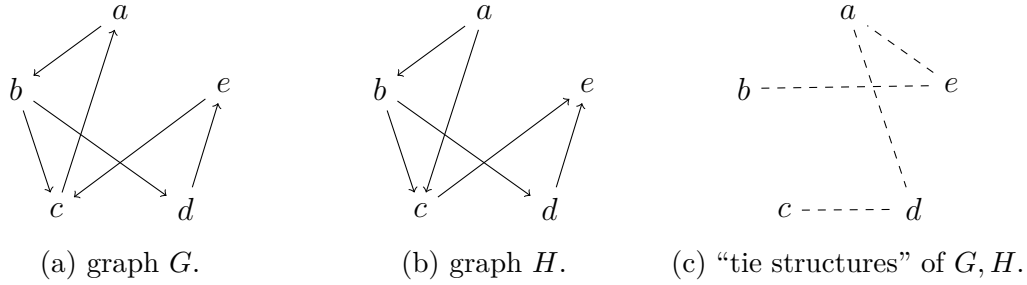


Figure 5.19:  $G$  and  $H$  has the same tie.

It is a nice result that every quasi-transitive oriented graph can be condensed into a tournament, which is one of the most understood families of digraphs. But just to say that a condensation to tournament exists is not helpful enough, since for every graph, there exists a trivial condensation, which produces a tournament with only one vertex (a single vertex is, by definition, a tournament). Therefore, we need to investigate what we call the “efficiency” of the tie component condensation.

**Definition 5.10.** *For two oriented graphs  $G, H$ , such that*

- $V(G) = V(H)$ ,
- $a$  ties  $b$  in  $H$  if and only if  $a$  ties  $b$  in  $G$ ,

*then we say  $G, H$  have the same **tie structure**.*

Graphs with the same tie structure means that if we draw only the ties from the two graphs, the graphs formed by the ties are the same. In other words, these two graphs may differ by the orientation of their edges, but not their ties.

In Fig. 5.19a and Fig. 5.19b, we show two graphs  $G$  and  $H$  with the same tie structure, In Fig. 5.19c, we draw only the ties from the previous two graphs, and discover that they are the same. The only differences between these two graphs are that  $c \rightarrow a, e \rightarrow c$  in  $G$ , however  $a \rightarrow c, c \rightarrow e$  in  $H$ , that is, the only differences between  $G$  and  $H$  are the orientations of these two edges.

**Corollary 5.23.** *For oriented graphs  $G$  and  $H$  with the same tie structure, if there exists a tie path between  $a, b$  in  $G$ , then there exists a tie path between  $a, b$  in  $H$ .*

*Proof.* Since given any two vertices  $p, q \in G$ , if  $p$  ties  $q$  in  $G$  then  $p$  ties  $q$  in  $H$ . Therefore if there exists a tie path  $[a, a_0, a_1, \dots, a_n, b]$  in  $G$ , then the same tie path exists in  $H$   $\square$

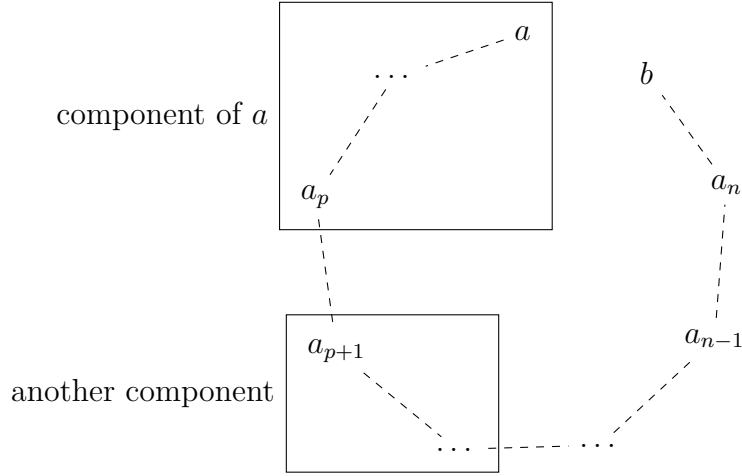


Figure 5.20: vertices  $a_p$  and  $a_{p+1}$  crosses components.

**Theorem 5.24.** *Given a quasi-transitive graph  $Q$  and its tie component condensation  $f$ , consider  $F = \{f_0, f_1, \dots, f_{n-1}, f_n\}$ , the set of all the condensations  $f_k : G_k \rightarrow T_k$ , where  $G_k$  has the same tie structure as  $Q$  and  $T_k$  is a tournament.  $f$  is an efficient condensation in  $F$ .*

*Proof.* Because all the  $G_k$  have the same tie structure as  $Q$ , all  $G_k$ 's have the same vertices.

If there exists another condensation  $f' : G' \rightarrow T'$  such that  $T'$  has more vertices than the underlying tournament  $T$  of  $Q$ , then there exists two vertices  $a, b$  in the same tie component in  $Q$ , and in different components in  $G'$ , because the number of components in the uncondensed graph is the same as the number of vertices in the condensed graph.

Because  $a, b$  are in the same component in  $Q$ , there exists a tie path between  $a$  and  $b$  in  $Q$ . By Corollary 5.23, there exists a tie path between  $a$  and  $b$  in  $G'$ . Because  $a$  and  $b$  are in different components in  $G'$ , there exists a point on the tie path between  $a$  and  $b$  that “crosses components” in  $G'$ .

Formally, denote the tie path between  $a, b$  as:  $[a, a_0, a_1, \dots, a_n, b]$ , then there exists  $a_p$  such that  $a_p$  and  $a_{p+1}$  are not in the same component. Because  $a_p$  ties  $a_{p+1}$  and they are in distinct components, then by Lemma 5.13,  $f'(a)$  ties  $f'(a')$ . Therefore  $T'$  is not a tournament. Contradiction.  $\square$

Theorem 5.24 states that the tie component condensation is not only a most efficient

condensation of quasi-transitive oriented graphs to tournaments, the tie component condensation is a most efficient condensation of oriented graphs with the same structure to tournaments.

To put it in an other way, for all the oriented graphs with the same tie structure, quasi-transitive oriented graphs are the ones that can be condensed into tournaments most efficiently.

## 5.6 Kings

The definition of king states that it can beat every vertex by one or two steps, and the definition of quasi-transitive oriented graph states that if there exists a path of length two (implies beats by 1 or 2 steps) from one vertex to another vertex, then they are adjacent.

Both definitions are related to “beats by 1 or 2 steps”, therefore kings in quasi-transitive oriented graphs have nice properties.

**Lemma 5.25.** *In a quasi-transitive oriented graph, a king is adjacent to every vertex.*

*Proof.* Given a king  $k$ , and another vertex  $v$ , then there are two possibilities:

- Case 1,  $k$  beats  $v$  by one step: therefore  $k \rightarrow v$ ,  $k$  and  $v$  are adjacent.
- Case 2,  $k$  beats  $v$  by two steps: there exists vertex  $a$ , such that  $k \rightarrow a \rightarrow v$ . Then by definition of quasi-transitive oriented graph,  $k$  needs to be adjacent to  $v$

□

**Lemma 5.26.** *In a quasi-transitive oriented graph, for any king  $k$ ,  $\{D_k, S_k, \{k\}\}$  partition the vertices of the graph.*

*Proof.*  $D_k, S_k$  disjoint because we are working in an oriented graph.  $D_k, S_k$  disjoint with  $\{k\}$  because a vertex cannot beat itself. Thus the three sets are disjoint.

The union of the three sets is the whole vertex set; by Lemma 5.25. □

**Lemma 5.27.** *In a quasi-transitive oriented graph, for any king  $k$ ,  $D_k, S_k$  are adjacent.*

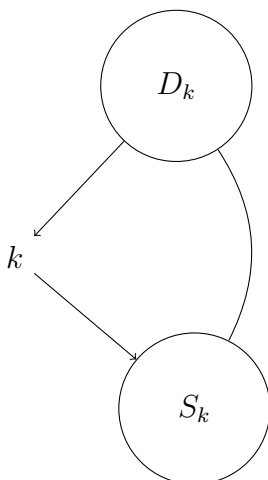


Figure 5.21: the rich structure of a king in a quasi-transitive oriented graph.

*Proof.* By definition of quasi-transitive oriented graph, because every vertex in  $D_k$  beats  $k$  and then beats every vertex in  $S_k$ , every vertex in  $D_k$  is adjacent to every vertex in  $S_k$ .  $\square$

**Theorem 5.28.** *In a quasi-transitive oriented graph, if we have a king  $k$ , then*

- $D_k, S_k, \{k\}$  partitions the vertex set.
- $D_k$  and  $S_k$  are adjacent.

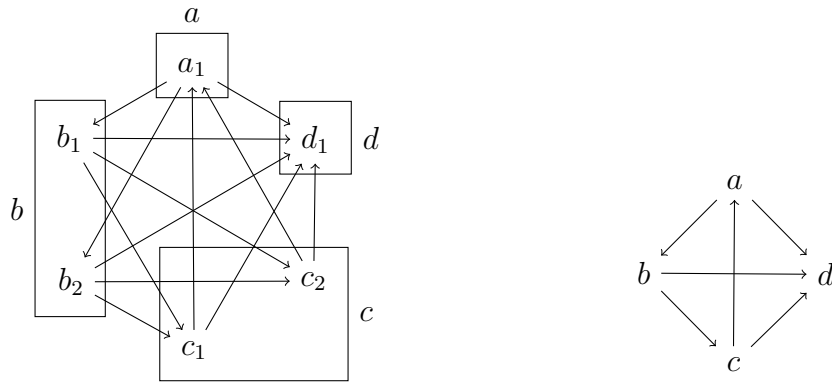
See Fig. 5.21, this figure shows the rich structure of a king in a quasi-transitive oriented graph. We were able to see a similar partition structure mentioned in Chapter 3, and Lemma 3.1 (the partition structure in Fig. 5.21 is the same as  $\{\{v\}, D_v, S_v\}$  for any vertex  $v$  in a tournament [6]).

$D_k$  cannot tie with anything outside of  $D_k$ , and similarly for  $S_k$ . If there is a king in a quasi-transitive oriented graph, then this graph becomes “very connected”, the only places ties can appear are inside the induced subgraphs of  $D_k$  and  $S_k$ .

Since we have deduced nice structures about ties in quasi-transitive oriented graphs, it is only logical to combine the property of king with the property of tie (tie component).

**Theorem 5.29.** *A vertex  $k$  is a king in a quasi-transitive oriented graph if and only if*

- $k$  is in a trivial tie component.



(a) quasi-transitive oriented graph  $G$ . (b) the underlying tournament of  $G$ .

Figure 5.22: a quasi-transitive digraph and its underlying tournament.

- $k$  is a king in the underlying tournament.

*Proof.* By Corollary 5.19,  $k$  is a king if and only if

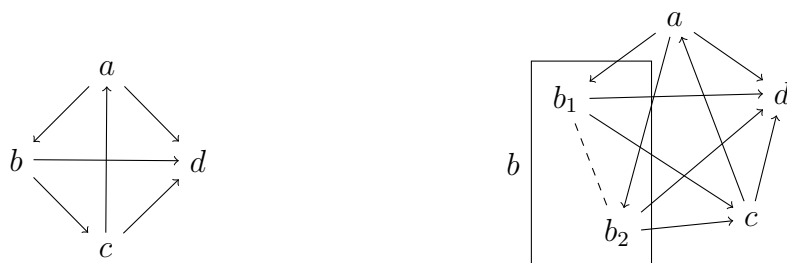
- $k$  is a king in the induced subgraph of its tie component.
- the tie component containing  $k$  is a king in the underlying tournament.

So we need to show  $k$  is a king in the induced subgraph of its tie component if and only if  $k$  is in a trivial tie component.

Suppose  $k$  is a king in the induced graph of its tie component, and the fact that the induced subgraph is a quasi-transitive oriented graph. Because of Lemma 5.25,  $k$  is a king in its tie component implies  $k$  is adjacent to all vertices in the tie component. There can be no tie path in this tie component. Therefore  $k$  has to be in a trivial tie component.

Suppose  $k$  is in a trivial tie component, then  $k$  is a king in the induced graph of its tie component. Recall that the definition of king says that in a digraph with one vertex, that vertex is a king in the digraph.  $\square$

In Fig. 5.22, we show a quasi-transitive oriented graph  $G$  and its underlying tournament. In the underlying tournament the kings are  $a, b, c$ , but only  $a_1$  can be a king of  $G$ , since vertices  $b$  and  $c$  do not correspond to trivial tie components. (tie component  $\{b_1, b_2\}$  is condensed into  $b$ , and tie component  $\{c_1, c_2\}$  is condensed into  $c$ ). Notice,



(a) start with a tournament with 3 kings. (b) then change king  $b$  to  $\{b_1, b_2\}$ .

Figure 5.23: construct a quasi-transitive oriented graph with 2 kings.

although vertex  $d_1$  is in a trivial tie component,  $d_1$  is not a king in  $G$ , because in the underlying tournament,  $d$  (the image of  $d_1$  under tie condensation) is not a king.

This theorem not only gives us a way to identify kings in a quasi-transitive oriented graph, but it helps us to construct a quasi-transitive oriented graph with a certain number of kings.

For example, if we want to construct a quasi-transitive oriented graph with  $k$  kings, we first start with a quasi-transitive graph with  $k'$  kings, where  $k' > k$ , and then we change  $k' - k$  kings into “non-trivial tie components”. This method enables us to construct a quasi-transitive oriented graphs with arbitrary number of kings. See Fig. 5.23, we start with a tournament with 3 kings  $\{a, b, c\}$ , then we change the king  $b$  into a non-trivial tie component  $\{b_1, b_2\}$ . Then the only kings left are  $a$  and  $c$ , and we get a quasi-transitive oriented graph with 2 kings.

Another significant usage of Theorem 5.29 is to use the properties of kings in tournaments to prove properties of kings in quasi-transitive oriented graphs.

**Definition 5.11.**  $k$  is a **great king** in a digraph  $G$ , if and only if  $k$  is the only king in digraph  $G$ , and  $k$  is not an emperor.

By Theorem 2.1, Corollary 3.5, a great king does not exist in semi-complete digraphs and tournaments.

**Corollary 5.30.** *If there exists a great king in a quasi-transitive oriented graph  $G$ , then there exists at least two ties in  $G$ .*

*Proof.* Assume  $k$  is the great king in  $G$ . Then there exists a vertex  $v$  such that  $v \rightarrow k$ .



Denote the underlying tournament of  $G$  as  $T$ , and the tie component condensation of  $G$  as  $f$ .

By Lemma 5.13,  $f(v) \rightarrow f(k)$ . Therefore, by Theorem 5.29  $f(k)$  is a king, but not an emperor in  $T$ . Because a tournament never has exactly 2 kings and exactly 1 king must be an emperor, there exists at least 3 kings in  $T$ . Because  $k$  is the only king in  $G$ , there has to exist at least 2 non-trivial tie components. Therefore,  $G$  have at least 2 ties.  $\square$

Theorem 5.29 provides us a way to detect, construct, and understand kings in quasi-transitive oriented graph using the properties and constructions of kings in tournaments.

# Chapter 6

## Further Problems

In Chapter 3 and Chapter 4, we see that because of the addition of two types of ties, we can construct many more  $(n, k)$  semi-complete digraphs or  $(n, k)$  oriented graphs than  $(n, k)$  tournaments. Then what can we construct if we limit the number of ties or double ties?

**Definition 6.1.** *an  $(n, k, t)$  digraph is a digraph with  $n$  vertices,  $k$  kings and  $t$  ties (or double ties).*

In Chapter 3, and Chapter 4, we explore the construction of  $(n, k)$  semi-complete digraphs and  $(n, k)$  oriented graphs with 1 tie. In other words, we have solved the problem of constructing  $(n, k, 1)$  oriented graphs and semi-complete digraphs. What happens when  $t$  gets larger?

One of the useful things to note when approaching the problem of the existence of  $(n, k, t)$  digraphs (or oriented graphs, semi-complete digraphs, etc.) is the construction of a quasi-transitive oriented graph with given number of kings, which appears at the end of Section 5.6. This way of constructing quasi-transitive oriented graphs may be very useful, since we can manipulate the number of ties in the graph without changing the number of vertices and kings. For a tie component with  $n$  vertices, the number of ties in the component can be any number between  $n - 1$  (ties form a spanning tree of the component) and  $\frac{n(n+1)}{2}$  (there exists a tie between every two vertices).

In Section 2.4, we showed that there exist kings with very low out-degrees. What

are the properties of a king with very low out-degree? What are the properties of a king with only out-degree 1? What are the properties of a king that has the lowest out-degree in the graph? What are the properties of a king that has the lowest out-degree among all of the kings in the graph?

In Section 5.6, we showed that a king has a very rich structure. However, Lemma 5.26 and Lemma 5.27 are not used in this thesis. Can we use these 2 properties to deduce any property of quasi-transitive oriented graphs with kings? A interesting path is that Lemma 5.27 implies that in a quasi-transitive oriented graph with king  $k$ , all the tie components are subsets of  $D_k$  or  $S_k$ .

**Definition 6.2.** *An infinite graph is a graph with an infinite number of vertices.*

Can we extend these results to an infinite graphs? Can we keep the definition of tie components when we expand the context to infinite graphs? How will the property of infinite path and infinite tie path change the results in this thesis?

In [2] and [3], the author uses the idea of strong digraph and strong components (see the definition of strong digraph and strong components in [8]) to deduce the properties of quasi-transitive oriented graphs. We can show that given any *non-trivial* condensation, the uncondensed graph is strong if and only if its underlying tournament is strong (*spoiler alert!*: Lemma 5.16 will be useful). Can we combine our results with the result in [2] and [3]?

In Section 5.4, we showed that graph condensation is a very useful transformation. and many graphs have only the identity condensation and the trivial condensation defined on them. How hard is a condensation? How many graphs have only those two condensations on them? What can be said about this class of graphs?

How useful are graph condensations? We showed that they preserve the shortest path (Theorem 5.17), and we can show that they preserve the “strong” property of the graph. What are other properties that graph condensations preserve?

# List of Figures

1.1	the relationship between tournaments, semi-complete digraphs, oriented graphs, and quasi-transitive oriented graphs. . . . .	2
2.1	example of a directed graph. . . . .	4
2.2	we can draw graph (a) as graph (b). . . . .	6
2.3	we can draw graph (a) as graph (b). . . . .	7
2.4	$D_v$ is the dominant set of $v$ ; $S_v$ is the submissive set of $v$ . . . . .	7
2.5	the process to create an induced subgraph. . . . .	8
2.6	example of an oriented graph. . . . .	9
2.7	example of a tournament. . . . .	9
2.8	an example with kings $e$ and $d$ . . . . .	10
2.9	the vertex with smallest out-degree is the only king. . . . .	11
2.10	vertex $e$ is the emperor of this graph. . . . .	11
3.1	an example of semi-complete digraph. . . . .	13
3.2	illustration of Lemma 3.1. . . . .	14
3.3	every vertex has the same out-degree. . . . .	15
3.4	both vertices $a$ and $b$ are emperors and kings. . . . .	16
3.5	an $(n + 1, k)$ semi-complete digraph by adding a new vertex $v$ . . . . .	16
3.6	an $(n + 1, k + 1)$ semi-complete digraph by adding a new king $a$ . . . . .	17
3.7	to construct a $(4, 2)$ semi-complete digraph. . . . .	17
3.8	a $(4, 4)$ semi-complete digraph with only one double tie . . . . .	18
4.1	all the oriented graph with 2 vertices. . . . .	21

4.2	all the oriented graphs with 3 vertices and maximum out-degree 1. . . . .	22
4.3	only $a$ and $b$ are kings for $n \geq 4$ . . . . .	23
4.4	the constructive proof for Lemma 4.11 . . . . .	24
4.5	$(2, 0)$ oriented graph. . . . .	24
4.6	construct $(n + 1, 0)$ oriented graph from $(n, 0)$ oriented graph. . . . .	25
5.1	if $a \rightarrow b \rightarrow c$ then $a$ is adjacent to $c$ . . . . .	26
5.2	all the possible beating relations between vertex $v$ and vertices $a, b$ . . . . .	27
5.3	only look at all the vertices adjacent to both $a$ and $b$ , then $\{a, b\}$ behaves like a vertex. . . . .	28
5.4	ties are “transmitting” arrow directions. . . . .	28
5.5	tie paths are just like paths. . . . .	29
5.6	tie path from $a$ to $b$ , and from $b$ to $c$ . . . . .	30
5.7	the tie path from $a_0$ to $a_n$ . . . . .	30
5.8	tie path transmits the arrow direction. . . . .	31
5.9	the induction step of the proof of Lemma 5.4 . . . . .	31
5.10	the tie components in (a) are the connected components in (b). . . . .	33
5.11	the arrow direction to $b$ was transmitted in tie component $C$ . . . . .	34
5.12	a vertex $b$ beats $f$ will force the component $C(a)$ to beats $f$ . . . . .	35
5.13	condense the tie components and get a tournament. . . . .	36
5.14	an example of graph condensation. . . . .	38
5.15	$\{b, c, d\}$ behaves like a vertex. . . . .	40
5.16	an oriented graph with only identity condensation and trivial condensation. . . . .	41
5.17	the path took a “detour” at $a_{p-1}$ . . . . .	41
5.18	information lost during condensation. . . . .	43
5.19	$G$ and $H$ has the same tie. . . . .	46
5.20	vertices $a_p$ and $a_{p+1}$ crosses components. . . . .	47
5.21	the rich structure of a king in a quasi-transitive oriented graph. . . . .	49
5.22	a quasi-transitive digraph and its underlying tournament. . . . .	50
5.23	construct a quasi-transitive oriented graph with 2 kings. . . . .	51

# Bibliography

- [1] Jørgen Bang-Jensen and Gregory Gutin. “Generalizations of tournaments: A survey”. In: *Journal of Graph Theory* 28.4 (1998), pp. 171–202.
- [2] Jørgen Bang-Jensen and Jing Huang. “Kings in quasi-transitive digraphs”. In: *Discrete mathematics* 185.1 (1998), pp. 19–27.
- [3] Jørgen Bang-Jensen and Jing Huang. “Quasitransitive digraphs”. In: *Journal of Graph Theory* 20.2 (Oct. 1, 1995), pp. 141–161. ISSN: 1097-0118. DOI: 10.1002/jgt.3190200205.
- [4] Susanna S. Epp. *Discrete mathematics with applications*. 4th ed. OCLC: ocn496962601. Boston, MA: Brooks/Cole, Cengage Learning, 2011. 1 p. ISBN: 978-0-495-39132-6.
- [5] Hortensia Galeana-Sánchez, César Hernández-Cruz, and Manuel Alejandro Juárez-Camacho. “On the existence and number of  $(k+1)$ -kings in  $k$ -quasi-transitive digraphs”. In: *Discrete Mathematics* 313.22 (Nov. 28, 2013), pp. 2582–2591. ISSN: 0012-365X. DOI: 10.1016/j.disc.2013.08.007.
- [6] Stephen B. Maurer. “The King Chicken Theorems”. In: *Mathematics Magazine* 53.2 (Mar. 1, 1980), p. 67. ISSN: 0025570X. DOI: 10.2307/2689952.
- [7] K. B. Reid. “Every vertex a king”. In: *Discrete Mathematics* 38.1 (1982), pp. 93–98.
- [8] Douglas Brent West. *Introduction to graph theory*. 2nd ed. Upper Saddle River, N.J: Prentice Hall, 2001. 588 pp. ISBN: 978-0-13-014400-3.

# Index

- out-degree, 7
- $(n, k)$  digraph, 11
- $(n, k)$  oriented graph, 11
- $(n, k)$  tournament, 11
- adjacent, 5
- beat by  $n$  steps, 5
- beat, 5
- component, 37
- condensation, 37
- condensed graph, 37
- digraph, 4
- directed graph, 4
- dominant set, 7
- dominate, 5
- double tie set of vertex, 13
- double tie, 12
- efficient condensation, 44
- emperor, 11
- graph, 4
- great king, 51
- in-degree, 7
- induced subgraph, 8
- king, 10
- length of path, 5
- oriented graph, 8
- path, 5
- quasi-transitive oriented graph, 26
- semi-complete digraph, 12
- strictly beat, 12
- submissive set, 7
- the length of the tie path, 29
- tie component condensation, 44
- tie component of a vertex, 32
- tie path, 29
- tie structure, 46
- tie, 5
- tournament, 9
- trivial tie component, 32
- uncondensed graph, 37
- underlying tournament, 45
- vertex-induced subgraph, 8
- walk, 5