ZERO-DIVISORS AND THEIR GRAPH LANGUAGES

HARLEY D. EADES III

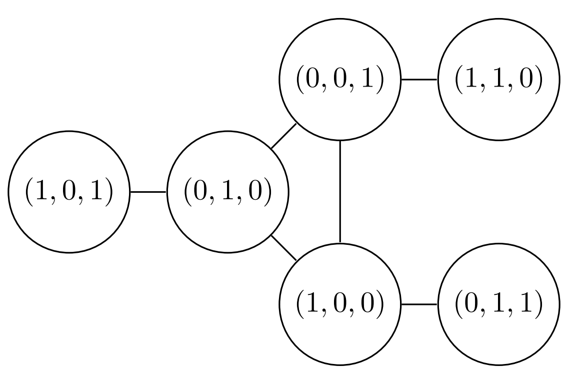
ABSTRACT. We introduce the use of formal languages in place of zero-divisor graphs used to study theoretic properties of commutative rings. We show that a regular language called a graph language can be constructed from the set of zero-divisors of a commutative ring. We then prove that graph languages are equivalent to their associated graphs. We go on to define several properties of graph languages.

1. INTRODUCTION

This article introduces the use of formal languages in place of zero-divisor graphs of a commutative ring with unity. We are interested mainly in whether we can define formal languages which provide the same information as zero-divisor graphs and what type of formal languages we obtain. In section two we primarily discuss how to obtain a formal language from the set of nonzero zero-divisors, and we show that it is regular. The remaining section deals with some interesting properties of graph languages.

We define several basic notions from commutative ring theory and formal language theory that are used throughout this article. Let R be a commutative ring with unity and $Z(R)$ be the set of zero-divisors of R. We define a simple graph $\Gamma(R)$, called the zero-divisor graph of R, with vertex set $Z(R)^* = Z(R) - 0$, and distinct $z_1, z_2 \in Z(R)^*$ are adjacent if and only if $z_1 z_2 = 0$. For example, $\Gamma(R)$ is the empty graph if and only if R is an integral domain [5]. Lets consider a short example.

Example 1.1. *Let $R = \mathbb{Z}_2 \times \mathbb{Z}_2 \times \mathbb{Z}_2$. Then $Z(R)^* = \{(1,0,1), (0,0,1), (0,1,0), (1,0,0), (1,1,0), (0,1,1)\}$. Using this set of zero-divisors we obtain the zero-divisor graph of R depicted below.*

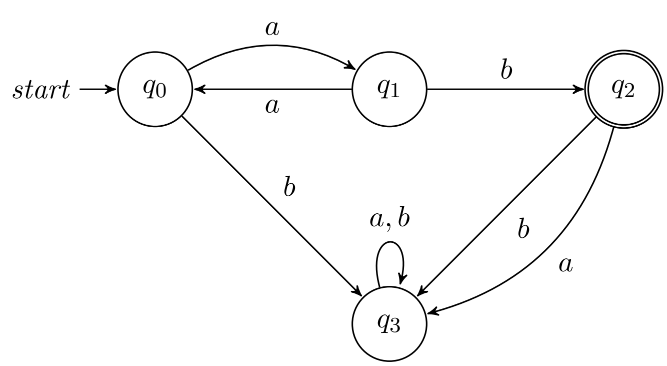


Let Σ be a set of symbols, called an alphabet, and Σ^* be the set of all possible concatenations of Σ. A *formal language* is a subset of Σ^*. A *deterministic finite accepter (DFA)* is defined by the quintuple, $M = (Q, \Sigma, \delta, q_0, F)$, where Q is a finite set of *internal states*, Σ is a finite set of symbols called the *input alphabet*, $\delta : Q \times \Sigma \to Q$ is the total function, called the *transition function*, $q_0 \in Q$ is the *initial state*, and $F \subseteq Q$ is a set of *final states*. Likewise, a *nondeterministic finite accepter (NFA)* is defined by the quintuple, $M = (Q, \Sigma, \delta, q_0, F)$ where Q, Σ, q_0, and F are as defined for deterministic finite accepters, but $\delta : Q \times (\Sigma \cup \lambda) \to 2^Q$. We define $S(M) = Q_M$ as the set of states of the finite automata M. Finally, $DFA(M_{NFA}(R, z))$ and $DFA(M_{NFA}(R))$ are the DFA's constructed using the subset construction algorithm found in [15, 20]. A language is *regular* when there exists a finite acceptor for it [15]. Before moving on we will now consider two examples illustrating the concepts discussed above.

Example 1.2. *Let $\Sigma = \{a, b, c\}$ and $L = \{w \mid w \in \Sigma^*$ and $w = a^r b$, where r is an odd integer$\}$. To show that L is regular we have to construct a finite automata which accepts all the words in L. There in fact does exist a DFA that accepts L. Let $M = (\{q_0, q_1, q_2, q_\}, \Sigma, \delta, q_0, \{q_2\})$, where δ is defined in the table below.*

δ	a	b
q_0	q_1	q_3
q_1	q_0	q_2
q_2	q_3	q_3
q_3	q_3	q_3

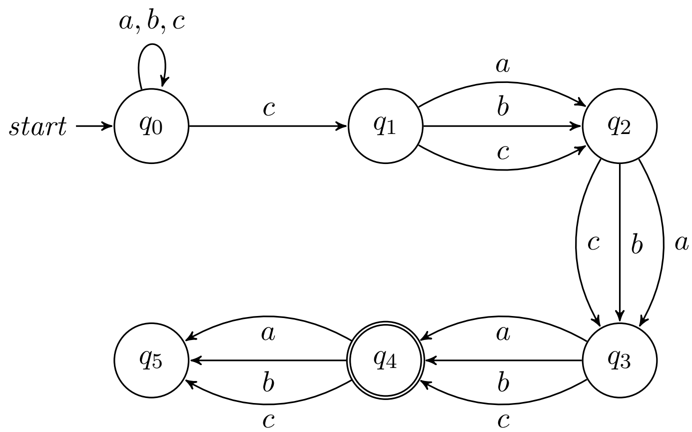
It is common to depict a finite automata as a transition diagram. The transition diagram for M is the following.



Example 1.3. *Let $\Sigma = \{a, b, c\}$ and $L = \{w \mid w \in \Sigma^*$ and the $fourth$ $symbol$ $from$ the $last$ $symbol$ in w is $c\}$. We construct a NFA accepting L. Let $M = (\{q_0, q_1, q_2, q_3, q_4, q_5\}, \Sigma, \delta, q_0, \{q_4\})$, where δ is defined in the table below.*

δ	a	b	c
q_0	$\{q_0\}$	$\{q_0\}$	$\{q_0, q_1\}$
q_1	$\{q_2\}$	$\{q_2\}$	$\{q_2\}$
q_2	$\{q_3\}$	$\{q_3\}$	$\{q_3\}$
q_3	$\{q_4\}$	$\{q_4\}$	$\{q_4\}$
q_4	$\{q_5\}$	$\{q_5\}$	$\{q_5\}$
q_5	$\{q_5\}$	$\{q_5\}$	$\{q_5\}$

The transition diagram for M is the following.



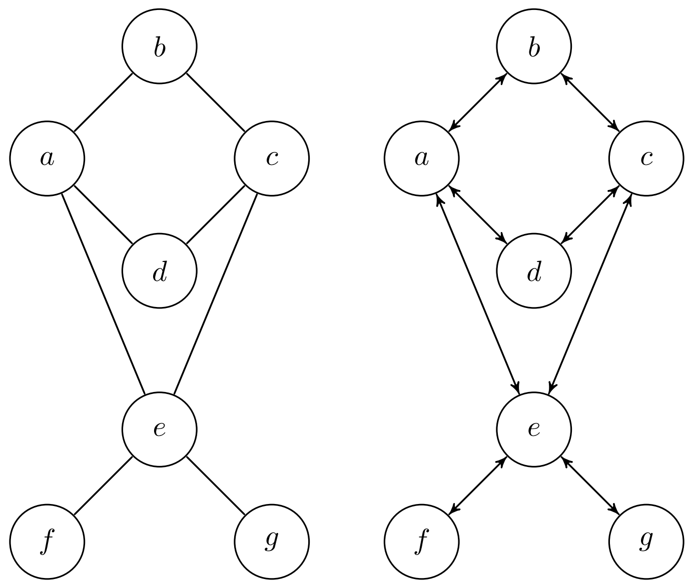
A grammar G is defined as a quadruple $G = (V, T, S, P)$, where V is a finite set of objects, called variables, T is a finite set of objects, called terminal symbols, $S \in V$ is a special symbol, called the start variable, and P is a finite set of productions. A grammar G is said to be *regular* if all productions are of the form $A \to xB$, $A \to x$ or $A \to Bx$, $A \to x$ where $A, B \in V$, and $x \in T^*$ [15]. A basic introduction to formal language theory can be found in [15, 20, 21], and an introduction to commutative ring theory and zero-divisor graphs can be found in [2, 3, 5, 6, 7, 8].

A word on notation is in order. The term ring should be read as commutative ring with unity; hence, we will implicitly assume that all rings are indeed commutative rings with unity which are not integral domains. We also abbreviate a formal language as simply a language. By subword we mean, that if u is a word in some language, then u is a *subword* if and only if there exists a word $w = vuz$, where v and z are words in the language [21].

2. Obtaining A Regular Graph Language

By definition, zero-divisor graphs of commutative rings are undirected. However, we will be converting directed graphs to formal languages, so we will need to be able to convert an undirected graph to a directed graph. To convert an undirected graph to a directed graph, simply replace all edges in the undirected graph with a directed edge. Then connect each vertex to all its adjacent vertices with directed edges. For example, in example 2.1 the graph on the left is an undirected graph, and its equivalent directed graph obtained by following the previous algorithm is on the right.

Example 2.1.



Throughout the remainder of this section, we will only consider directed graphs.

We state next an algorithm called *The Graph Conversion Algorithm (TGCA)* which converts any directed graph into a grammar which generates a language whose words are all the paths of a directed graph starting from a particular start vertex. We call such a language a *start-vertex graph language*.

Definition 2.1.

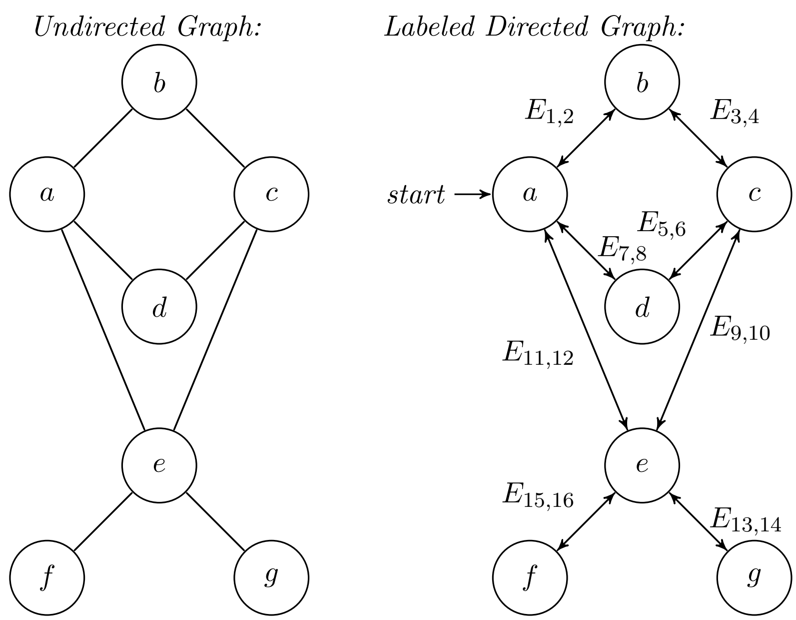
INPUT: Digraph $D = (V, E)$.

1. *Give each vertex of D some label. The vertex labels represent the terminals of the grammar. It is important that the vertex labels are distinct from each other and distinct from the edge labels.*
2. *Label each edge of D with E_i where $0 \leq i \leq |E|$, and all the E_i are distinct. If the edges of D are already labeled, replace those labels with the E_i labels. These labels represent the nonterminals of the grammar.*
3. *Choose a start vertex. The start vertex designates the location in the graph where* all *the paths start. Let $a \in V$ be the start vertex. Then for each outgoing edge E_i of a, write rules $S \to aE_i$. Then write the rule $S \to a$.*

4. *For each of the remaining vertices and for each outgoing edge E_j of each vertex, write rules $E_i \to bE_j$. Then write the rule $E_i \to b$, where $b \in V$ and E_i is an incoming edge of b.*
5. *For each of the incoming edges, make sure* not *to introduce duplicates. If the left side of E_i matches another nonterminals left side, replace all occurrences of E_i with the matched nonterminal.*
   OUTPUT: A grammar G in TGCA form.

Before we move on we consider an example using TGCA.

Example 2.2. *The graph on the left is the zero-divisor graph of $\mathbb{Z}_{12}$, and the graph on the right is the graph on the left after the labeling scheme defined by TGCA is applied (steps 1 - 3).*



Next we follow steps 4 and 5 of the algorithm, and we obtain the following grammar in TGCA form.

$$
\begin{array}{lclclclclcl}
S & \rightarrow & aE_1 & \mid & aE_7 & \mid & aE_{11} & \mid & a & & \\
E_1 & \rightarrow & bS & \mid & bE_3 & \mid & b & & & & \\
E_7 & \rightarrow & dS & \mid & dE_3 & \mid & d & & & & \\
E_{11} & \rightarrow & eS & \mid & eE_3 & \mid & eE_{15} & \mid & eE_{13} & \mid & e \\
E_3 & \rightarrow & cE_1 & \mid & cE_7 & \mid & cE_{11} & \mid & c & & \\
E_{15} & \rightarrow & fE_{11} & \mid & f & & & & & & \\
E_{13} & \rightarrow & gE_{11} & \mid & g. & & & & & &
\end{array}
$$

This grammar generates the language whose words are every path starting from the vertex labeled a *of the graph above.*

We now prove that the output grammar is regular and has the smallest number of productions needed to generate a start-vertex graph language.

Theorem 2.1. *Every digraph $G = (V, E)$ can be converted into a minimal regular grammar, where the number of production rules $= |V| + |E|$.*

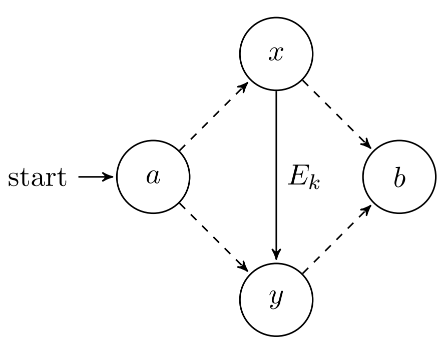
Proof. By induction on the number of edges, take a digraph of two vertices with a single edge $(a, b) = E_1$ as our base case. Then by TGCA the grammar is

$$
\begin{array}{lclcl}
S & \rightarrow & aE_1 & \mid & a \\
E_1 & \rightarrow & b. & &
\end{array}
$$

The number of productions is $|E| + |V|$, and this satisfies our base case.

Now assume the theorem and the algorithm are true up to $|E| = n$ edges. Consider a graph with $n + 1$ edges. We break it into two cases.

Case 1: We let the graph be connected with an arbitrary edge E_k which when removed the graph remains connected. The graph and the grammar in TGCA form are



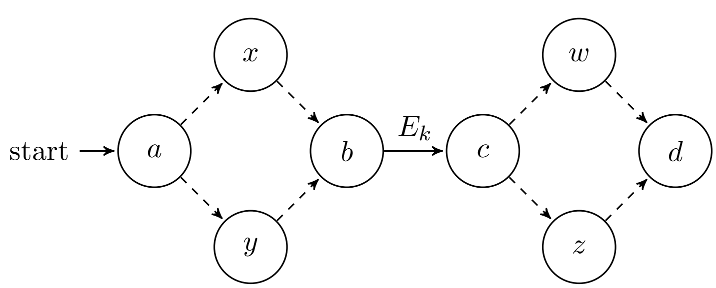
$$
\begin{array}{l}
S \rightarrow aE_1 | \cdots | a \\
\vdots \\
E_i \rightarrow xE_k | \cdots | x \\
\vdots \\
E_{n+1} \rightarrow \cdots | b.
\end{array}
$$

Consider Case 1. If we remove the edge labeled E_k from the graph, we obtain a graph of n edges; hence the inductive hypothesis applies. So, if we add the production $E_i \to xE_k$ to the grammar in the inductive hypothesis, we obtain

$$\begin{aligned}
S &\to aE_1|\cdots|a \\
&\ \ \vdots \\
E_i &\to xE_k|\cdots|x \\
&\ \ \vdots \\
E_{n+1} &\to \cdots|b.
\end{aligned}$$

This grammar remains regular by definition. This is a minimal regular grammar because our addition amounts to adding an additional edge labeled E_k to the graph in the inductive hypothesis, hence, the number of production rules $= |V|+|E|$, and it corresponds to the grammar obtained from the graph before the edge removal.

Case 2: We consider a graph that when an arbitrary edge E_k is removed we obtain a disconnected graph. This graph has the following form



and its associated grammar in TGCA form is then

$$
\begin{aligned}
S &\rightarrow aE_1|\cdots|a \\
&\vdots \\
E_i &\rightarrow bE_k|\cdots|b \\
&\vdots \\
E_k &\rightarrow \cdots|c \\
&\vdots \\
E_{n+1} &\rightarrow \cdots|d.
\end{aligned}
$$

Now consider Case 2. If we remove the edge labeled E_k, we obtain two disjoint graphs with n or fewer edges. Thus, by the inductive hypothesis, we obtain two minimal regular grammars which correspond to the two disjoint graphs of n or fewer edges. If we add the production rule $E_i \rightarrow bE_k$ to the first grammar, we obtain the grammar of the graph where the disjoint graphs are connected by the edge E_k. This grammar is regular by definition. Since our addition is only a single edge, the regular grammar remains minimal where the number of production rules $= |V| + |E|$.

□

At this point the regular grammars we are defining only generate start-vertex graph languages. Since we are working with regular grammars, and since regular grammars generate regular languages which are both closed under union ([15, 20]), we can use TGCA on every vertex of the graph and union the grammars together. Therefore, we obtain the regular language whose words are all the paths of a graph, called the *graph language*.

Up until now, we have discussed obtaining a graph language for any graph. Our research is mainly concerned with zero-divisor graphs, and our ultimate goal is to be able to move away from the zero-divisor graph completely and talk about languages. TGCA works for any graph, but it requires we work directly with the graph. So, we ask the question, is it possible to obtain the graph language of a zero-divisor graph without using the graph at all?

3. Properties of graph languages

To answer this question let R be a ring, and define a function $f : R \times R \rightarrow \{0, 1\}$ by

$$
f(a, b) = \begin{cases} 1 & : \quad ab = 0 \\ 0 & : \quad otherwise \end{cases}
$$

The function f determines if two nonzero elements of R multiply to zero. Using the previous function, we define $\phi : \Sigma^* \to L$, where $\Sigma = Z(R)^*$, by

$$\phi(a_0 \cdots a_n) = \begin{cases} a & : \quad n = 1 \\ a_0 \cdots a_n & : \quad f(a_0, a_1) = \cdots = f(a_{n-1}, a_n) = 1 \\ \lambda & : \quad \textit{otherwise.} \end{cases}$$

The regular language L contains as words all the paths of the zero-divisor graph of R, hence we denote this graph language, by L_R (the start-vertex graph language where every word starts with $z \in Z(R)^*$ is denoted $L_{R,z}$).

Since L_R is regular, it is possible to construct a finite automata which accepts the words of L_R. We construct such a machine directly from $Z(R)^*$. For all $z_i \in Z(R)^*$, write down a state labeled z_i. There exists an edge labeled z_i between two states z_i and z_j if and only if $f(z_i, z_j) = 1$. Every state of this machine is final. Choose z_i as the start state. This machine is an NFA which accepts all the words in L_{R,z_i}. We denote this machine by $M_{NFA}(R, z_i)$. Next write down a state labeled S. Connect an edge from S labeled z_i to its corresponding start state in $M(R, z_i)$ for all $z_i \in Z(R)$. The state S is our new start state. This new machine is the NFA which accepts all the words of L_R, and we denote said NFA by $M_{NFA}(R)$. In fact, $M_{NFA}(R) = \bigcup_{z \in Z(R)^*} M_{NFA}(R, z)$. Likewise, we denote $M_{NFA}(R, z_i)$'s and $M_{NFA}(R)$'s equivalent DFAs as $M_{DFA}(R, z_i)$ and $M_{DFA}(R)$, respectively.

To be able to completely move away from zero-divisor graphs, the graph languages must contain all the information obtainable from the graphs. In this section we show exactly where this information is hiding in the graph language.

Definition 3.1. *Let R be a ring. Then $Fact(L_R) = \{ a_0 b_1 \cdots b_i a_1 \in L_R \mid a_0$ and a_1 are not necessarily distinct, $a_0 \neq a_1 \neq \lambda$, $b_i's$ are unique, and $b_i \neq a_0 \ \wedge b_i \neq a_1 \}$ is a set of subwords of length at least 2 composed of mainly unique symbols of all the words in L_R.*

In the previous definition we assert that the subwords are composed of mainly unique symbols, because the first and the last symbol may be equivalent. $Fact(L_R)$ is easily constructed from the zero-divisor graph itself. Consider the zero-divisor graph associated with the paths in L_R. Starting from each vertex, follow each edge to every other vertex, writing down the path after crossing an edge. This set of paths is $Fact(L_R)$. We use this notion in the proof of the next theorem.

Definition 3.2. *A graph G is* isomorphic *to its associated graph language L_G if and only if there exists an one-to-one correspondence between the paths in G and the words in L_G.*

Theorem 3.1. *Let L be a graph language. Then L is isomorphic to its associated graph G.*

Proof. To prove Theorem 3.2 we construct $Fact(L_R)$ directly from some arbitrary graph. Let $G = (V, E)$ be a finite connected graph, where $V = \{a_0, ..., a_1\}$, and let L be its associated graph language. Starting from a_0, write down the paths to each adjacent vertex. Let a_1 be adjacent to a_0, then write down the paths starting with a_0a_1 and ending at each adjacent vertex of a_1, excluding the edge took to get to a_1. Continue this pattern until an end is reached or a_0 is reached forming a cycle. Do this for every vertex of G. This set of paths is exactly $Fact(L)$. Therefore, L is isomorphic to G. □

Theorem 3.2 implies that graphs and graph languages are interchangeable. This is a very pleasing result because it means among other things that all theorems, definitions, lemmas, corollaries, and so forth dealing with zero-divisor graphs ([1, 2, 3, 4, 5, 6, 8, 9, 10, 12, 13, 14, 16, 17, 18, 19]) have analogues in graph languages.

We now show that the NFA which accepts all the words in a zero-divisor graph language starting from a particular start-vertex is isomorphic to its associated zero-divisor graph.

Definition 3.3. *Let R be a ring and $z \in Z(R)^*$. Then a zero-divisor graph $\Gamma(R)$ is isomorphic to a $M_{NFA}(R, z)$ if and only if there exists a one-to-one correspondence ϕ between the vertices of $\Gamma(R)$ and the states of $M_{NFA}(R, z)$ where a and b are two adjacent vertices of $\Gamma(R)$ if and only if the state $\phi(a)$ is connected to $\phi(b)$ with an edge labeled a and $\phi(b)$ is connected to $\phi(a)$ with an edge labeled b in $M_{NFA}(R, z)$.*

Theorem 3.2. *Let R be a ring and let $\Gamma(R)$ be its zero-divisor graph. Then each $M_{NFA}(R, z)$ for each $z \in Z(R)^*$ is isomorphic to $\Gamma(R)$.*

Proof. Let $\phi : V(\Gamma(R)) \to S(M_{NFA}(R, z))$, where $z \in Z(R)^*$, be defined by $\phi(x) = x$. By definition $V(\Gamma(R)) = Z(R)^* = S(M_{NFA}(R, z))$. Clearly, ϕ is a one-to-one correspondence. Furthermore, z_i and z_j are adjacent in $\Gamma(R)$ if and only if $f(z_i, z_j) = 1$ and adjacent in $M_{NFA}(R, z)$ if and only if $f(z_i, z_j) = 1$. Hence, z_i and z_j are adjacent if and only if $\phi(z_i)$ and $\phi(z_j)$ are as well with appropriately labeled edges. Therefore, $\Gamma(R)$ and $M_{NFA}(R, z)$ are isomorphic. □

Another proof of their equivalence can be found in the construction of $M_{NFA}(R, z)$ directly from $\Gamma(R)$. We first convert $\Gamma(R)$ into a directed graph, and we choose z as the start state. For each vertex x we label each outgoing edge x. All vertices are final states. This NFA is exactly $M_{NFA}(R, z)$.

Theorem 3.2 allows us to deduce some interesting properties of $M_{NFA}(R, z)$. For example, the length of the longest cycle in $M_{NFA}(R, z)$ is less than or equal to 4, and the longest distance from any two states without taking a detour or cycle is less than or equal to 3 [2, 6, 16]. We now move away from the NFA's of graph languages toward their DFA's.

We next find an upper bound on the number of states of any $M_{DFA}(R, z)$.

Theorem 3.3. *The number of states in any $M_{DFA}(R, z)$ where $z \in Z(R)^*$ is less than or equal to $|Z(R)^*| + 1$.*

Proof. By subset construction, we can identify the states of $M_{DFA}(R, z)$. The subsets, and hence the set of states of $DFA(M_{NFA}(R, z))$, are $Q = \{Q_z \mid Q_z = \{x \mid f(z, x) = 1 \text{ where } x \in Z(R)^*\} \text{ for all } z \in Z(R)^*\}$. Now suppose $|Q| > |Z(R)^*|$. Then there must exist at least two elements $Q_i, Q_j \in Q$ where $Q_i \neq Q_j$ for a particular element of $Z(R)^*$. This is a contradiction. Therefore, $|S| \leq |Z(R)^*| + 1$. □

In Theorem 3.3 the NFA to DFA algorithm will obtain an extra state if and only if the start state is an end in $\Gamma(R)$. The subsets used in the proof can be constructed directly from $\Gamma(R)$. For each vertex v, Q_i is the set of vertices adjacent to v. The distinct Q_i's are the states of $M_{DFA}(R, z)$. We now turn to an example of when the number of states of a DFA is less than or equal to $|Z(R)^*| + 1$.

Example 3.1. *Consider the zero-divisor graph $\Gamma(\mathbb{Z}_{12})$. $\Gamma(\mathbb{Z}_{12})$ has several symmetries, so, the NFA to DFA algorithm will output less states than there are vertices in $\Gamma(\mathbb{Z}_{12})$, because several vertices have the same set of neighbors. In fact, $M_{DFA}(\mathbb{Z}_{12}, z)$ will have at most five states while $Z(\mathbb{Z}_{12})^*$ has only seven vertices. Now consider the NFA of the graph language associated with the zero-divisor graph of the ring $\mathbb{Z}_4[X]/(X^2+X+1)$. The zero-divisor graph of this ring only has three vertices, but for any particular start vertex the DFA has exactly four states.*

Next we show exactly when the number of states of any $M_{DFA}(R, z)$ is less than $|Z(R)^*| + 1$ and we find the upper bound on the number of states in any $M_{DFA}(R)$.

Theorem 3.4. *Let R be a ring. The number of states in $M_{DFA}(R, z)$ for some $z \in Z(R)^*$ is less than $|Z(R)^*|+1$ if and only if there exists at least two distinct elements of $x, y \in Z(R)^*$ such that $Q_x = \{r \mid f(x, r) = 1 \text{ where } r \in Z(R)^*\} = \{s \mid f(y, s) = 1 \text{ where } s \in Z(R)^*\} = Q_y$.*

Proof. ($\Leftarrow$) By definition.
($\Rightarrow$) Let R be a ring. Suppose for some $z \in Z(R)^*$ we have $|S(M_{DFA}(R, z))| < |Z(R)^*| + 1$. Then $|S(M_{DFA}(R, z)) - \{z\}| < |Z(R)^*|$. By subset construction $|S(M_{DFA}(R, z)) - \{z\}| \geq |Q|$, where $Q = \{Q_y \mid Q_y = \{x \mid f(y, x) = 1 \text{ where } x \in Z(R)^*\} \ \forall y \in Z(R)^*\}$. Hence, $|Q| < |Z(R)^*|$. Let $\phi : Z(R)^* \to Q$, where $\phi(a) = Q_a$. By definition ϕ is onto; thus, there exists an element of $Z(R)^*$ for every element of Q, but $|Q| < |Z(R)^*|$; hence, ϕ is not one-to-one. However, by the definition of Q there exists an element of $Z(R)^*$ for every element of Q. Therefore, there must exist at least two elements $a, b \in Z(R)^*$ such that $\phi(a) = \phi(b)$ and $a \neq b$. □

Corollary 3.5. *The number of states in any $M_{NFA}(R)$ is less than or equal to $|Z(R)^*|^2 + |Z(R)^*| + 1$.*

Proof. Corollary 3.4 can be established easily by the notion that $M_{NFA}(R) = \bigcup_{z \in Z(R)^*} M_{DFA}(R, z)$. If each $M_{DFA}(R, z)$ has the largest possible number of states, we will have a $M_{DFA}(R, z)$ for each zero-divisor, and we obtain $|Z(R)^*|$ DFA's with $|Z(R)^*| + 1$ states. We link all these DFA's together with a new start state. Hence, we obtain the value in the corollary. □

In the last couple of theorems and corollaries, we established an upper bound on the number of states of the DFA's of zero-divisor graph languages. Next we establish a specific number of states for the DFA's of zero-divisor graphs languages when the NFA of a zero-divisor graph language is isomorphic to a star graph.

Theorem 3.6. *Let R be a ring and let $z \in R$. Then $M_{NFA}(R, z)$ is isomorphic to a star graph if and only if $DFA(M_{NFA}(R, z))$ has two states if (z is the center of $M_{NFA}(R, z)$) or has three states (if z is an end-state of $M_{NFA}(R, z)$).*

Proof. ($\Rightarrow$) Let R be a ring and $M_{NFA}(R, z)$ be the NFA which accepts all the words in the zero-divisor graph language of R starting with z. Suppose $M_{NFA}(R, z)$ is isomorphic to a star-graph and let z be the center of $M_{NFA}(R, z)$. Then by subset construction let $Q = \{Q_y \mid Q_y = \{x \mid f(y, x) = 1 \; where \; x \in Z(R)^*\} \; for \; all \; y \in Z(R)^*\}$ be the set of states of $DFA(M_{NFA}(R, z))$. Since the center of $M_{NFA}(R, z)$ is z, then $Q = \{Q_z, Q_x \mid Q_z = Z(R)^* - \{z\} \; and \; Q_x = \{z\} \; for \; all \; x \in Z(R)^* - \{z\}\}$. Thus, $|Q| = 2$. Next assume z is an end-state of $M_{NFA}(R, z)$. Then Q is defined as before, but every word must start with z; therefore, the set of states of $DFA(M_{NFA}(R, z))$ is $Q \cup \{z\}$. Therefore $|Q \cup \{z\}| = 3$.
($\Leftarrow$) Let $S(DFA(M_{NFA}(R, q_0))) = \{q_0, q_1\}$. If $S(M_{NFA}(R, q_0)) = \{q_0, q_1\}$, where $q_0, q_1 \in Z(R)^*$, then clearly $M_{NFA}(R, q_0)$ is isomorphic to a star graph. If $S(M_{NFA}(R, q_0)) = \{q_0, p_0, \ldots, p_n\}$, then $q_1 \subseteq S(M_{NFA}(R, q_0))$ where $q_1 \in Q = \{Q_x \mid Q_x = \{z \mid f(x, z) = 1 \; for \; all \; z \in Z(R)^*\} \; for \; all \; x \in Z(R)^*\}$ thus, $Q_{p_i} = q_1, i = 0, \ldots, n$. Therefore, $M_{NFA}(R, q_0)$ is isomorphic to a star graph.

We now consider three states. Let $DFA(M_{NFA}(R, q_i))$ where $i \in \{0, 1, 2\}$ have three states q_0, q_1 and q_2. According to the NFA to DFA algorithm q_0, q_1 and q_2 fall into the following categories: 1. $q_0 \in Z(R)^*$ and $q_1, q_2 \subseteq Z(R)^*$, 2. $q_1 \in Z(R)^*$ and $q_0, q_2 \subseteq Z(R)^*$, 3. $q_2 \in Z(R)^*$ and $q_0, q_1 \subseteq Z(R)^*$, 4. $q_0, q_1 \in Z(R)^*$ and $q_2 \subseteq Z(R)^*$, 5. $q_0, q_2 \in Z(R)^*$ and $q_1 \subseteq Z(R)^*$, 6. $q_1, q_2 \in Z(R)^*$ and $q_0 \subseteq Z(R)^*$, and 7. $q_0, q_1, q_2 \in Z(R)^*$.

It is possible to rule out several categories from being DFAs of a zero-divisor graph. Consider category 7. All the states being zero-divisors implies that $M_{NFA}(R, q_i)$, where $i \in \{0, 1, 2\}$, is a DFA, and each q_i is adjacent to only one other vertex. This is impossible by the property that all zero-divisor graphs are undirected and Theorem 3.2.

Now consider categories 4 through 6. Without loss of generality, let $q_0, q_1 \in Z(R)^*$ and $q_2 \subseteq Z(R)^*$, Then q_2 is not adjacent to either q_0 or

q_1, because by the NFA to DFA algorithm q_2 is adjacent to a subset of $Z(R)^*$. Hence, no DFAs have states in these categories.

Consider categories 1 through 3. Without loss of generality let $q_0 \in Z(R)^*$ and $q_1, q_2 \subseteq Z(R)^*$. Then q_0 is adjacent to only one of q_1 or q_2, and neither q_1 nor q_2 is adjacent to q_0. Hence, q_0 is the start state. If q_0 is adjacent to q_1, then $|q_1| = 1$ since q_0 is an end in $\Gamma(R)$. Likewise, if q_0 is adjacent to q_2, then $|q_2| = 1$. Now if q_0 is adjacent to q_1 and q_1 is adjacent to q_2, then q_2 is adjacent to q_1 because $\Gamma(R)$ is always undirected. Similar cases are true for when q_0 and q_2 are adjacent and when q_2 and q_1 are adjacent. One of the cases just stated must occur because $\Gamma(R)$ is always connected [5, Theorem 2.3].

In any case there exists a subset $x \subseteq Z(R)^*$ such that $|x| = 1$ and all elements of $Z(R)^* - x$ are adjacent to $z \in x$. Hence, z is adjacent to all elements of $Z(R)^* - x$. Now since every element of $Z(R)^* - x$ is adjacent to only one element of $Z(R)^*$, every element of $Z(R)^* - x$ is an end. Therefore, $M_{NFA}(R, q_i)$ is adjacent to a star graph. Thus, category 1 through 3 is the only set that makes up an actual DFA, and the NFA of said DFA is isomorphic to a star graph. $\square$

Anderson and Livingston in [5] showed that if $\Gamma(R)$ has greater then or equal to 4 vertices then R is isomorphic to $\mathbb{Z}_2 \times F$, where F is a field, if and only if $\Gamma(R)$ is a star graph. Thus, we obtain yet another corollary.

Corollary 3.7. *Let R be a ring with $|Z(R)^*| \geq 4$. Then $M_{DFA}(R, z)$ for any $z \in Z(R)^*$ has two or three states if and only if $R \cong \mathbb{Z}_2 \times F$, where F is a field.*

Proof. By [5, Theorem 2.13] $\Gamma(R)$ is a star graph if and only if $R \cong \mathbb{Z}_2 \times F$. Then by Theorem 3.2 $\Gamma(R)$ is isomorphic to each $M_{NFA}(R, z)$ for each $z \in Z(R)^*$. Therefore, by Theorem 3.7 $DFA(M_{NFA}(R, z))$ for each $z \in Z(R)^*$ has two or three states if and only if $M_{NFA}(R, z)$ is isomorphic to a star graph. $\square$

4. Acknowledgments

I would like to thank Dr. Joe Stickles and Dr. James Rauff of Millikin University's Mathematics department for their exceptional feedback and suggestions. Without their help and interest this research would never have happened.

References

[1] S. Akbari, H.R. Maimani, and S. Yassemi. *When a zero-divisor graph is planar or a complete r-partite graph. J. Algebra*, **270**(**1**):169–180, 2003.

[2] D.F. Anderson. *On the diameter and girth of a zero-divisor graph, II. Houston J. Math.*, **34**(**2**):361–371, 2008.

[3] D.F. Anderson, A. Frazier, A. Lauve, and P.S. Livingston. *The zero-divisor graph of a commutative ring, II. Lectures Notes in Pure and Appl. Math.*, **220**:67–72, 2001.

[4] D.F. Anderson, R. Levy, and J. Shapiro. *Zero-divisor graphs, von Neumann regular rings, and Boolean algebras. J. Pure Appl. Algebra*, **180(3)**:221–241, 2003.

[5] D.F. Anderson and P.S. Livingston. *The zero-divisor graph of a commutative ring. J. Algebra*, **217(2)**:434–447, 1999.

[6] D.F. Anderson and S.B. Mulay. *On the diameter and girth of a zero-divisor graph. J. Pure Appl. Algebra*, **210(2)**:543–550, 2007.

[7] M.F. Atiyah and I.G. MacDonald. Introduction to Commutative Algebra. Perseus Book, Cambridge, Massachusetts, 1969.

[8] M. Axtell, J. Coykendall, and J. Stickles. *Zero-divisor graphs of polynomial and power series over commutative rings. Comm. Algebra*, **33(6)**:2043–2050, 2005.

[9] M. Axtell, J. Stickles, and W. Trampbachls. *Zero-divisor ideals and realizable zero-divisor graphs. Involve*, to appear.

[10] R. Belshoff and J. Chapman. *Planar zero-divisor graphs. J. Algebra*, **316(1)**:471–480, 2007.

[11] G. Chartrand. Introductory Graph Theory. Dover Publications, New York, Mineola, 1985.

[12] F. DeMeyer and K. Schneider. *Automorphisms and zero-divisor graphs of commutative rings. Int. J. Commut. Rings*, **1(3)**:93–106, 2002.

[13] J.D. LaGrange. *Complemented zero-divisor graphs and Boolean rings. J. Algebra*, **315(2)**:600–611, 2007.

[14] J.D. LaGrange. *On realizing zero-divisor graphs. Comm. Algebra*, **36(12)**:4509–4520, 2008.

[15] Peter Linz. Formal Languages and Automata. Jones and Bartlett Publishers, Sudbury, Massachusetts, 2006.

[16] T.G. Lucas. *The diameter of a zero-divisor graph. J. Algebra*, **301(1)**:174–193, 2006.

[17] S.B. Mulay. *Cycles and symmetries of zero-divisors. Comm. Algebra*, **30(7)**:3533–3558, 2002.

[18] S.P. Redmond. *An ideal-based zero-divisor graph of a commutative ring. Comm. Algebra*, **31(9)**:4425–4443, 2003.

[19] S.P. Redmond. *On zero-divisor graphs of small finite commutative rings. Discrete Math.*, **307(9)**:1155–1166, 2007.

[20] G. Rozenberg and A. Eds. Salomaa. Handbook of Formal Languages, volume 1. Springer-Verlang, Berlin, Heidelberg, 1997.

[21] A. Solomaa. Jewels of Formal Language Theory. Computer Science Press, Rockville, Maryland, 1981.

Department of Computer Science, University of Iowa, 14 MacLean Hall, Iowa City, IA 52242-1419

E-mail address: harley-eades@uiowa.edu