The state complexity of language operations on XNFA-succinct unary regular languages

Laurette Marais
Meraka Institute and Stellenbosch University
South Africa
LPretorius@csir.co.za

Lynette van Zijl
Stellenbosch University
South Africa
lvzijl@sun.ac.za

ABSTRACT

Given two unary languages accepted by symmetric difference nondeterministic finite automata, we establish bounds on the state complexity of their union, intersection, relative complement and symmetric difference. For languages $\mathcal{L}_1$ and $\mathcal{L}_2$ accepted by minimal symmetric difference nondeterministic finite automata of size m and n respectively, we show that the state complexity of their union, intersection and relative complement has an upper bound of $(2^m - 1)(2^n - 1)$.

CCS CONCEPTS

• **Theory of computation → Formal languages and automata theory**;

KEYWORDS

Descriptional complexity, nondeterminism, language operations

ACM Reference Format:
Laurette Marais and Lynette van Zijl. 2018. The state complexity of language operations on XNFA-succinct unary regular languages. In *Proceedings of SAICSIT conference (SAICSIT'18).* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Symmetric difference nondeterministic finite automata (XNFA) are automata that provide a favourable model for cyclic behaviour [1] such as, for example, long sequences in random number generation. In terms of the state complexity of XNFA, it has been shown that there are regular languages that have no succinct representation with NFA, but do have with XNFA. Indeed, in the case of unary regular languages, there is a tight upper bound of $2^n - 1$ states for the conversion of an n-state XNFA to a deterministic finite automaton (DFA). This stands in contrast to the upper bound for the conversion of unary nondeterministic finite automaton (NFA) to DFA, which is known to be $e^{\sqrt{n \log n}}$ [2].

An important question in the study of regular languages concerns the cost of applying an operation on a regular language. The state complexity (sc) of a regular language is given by the number of states of its minimal DFA. So if an operation is applied to one or more regular languages, the cost of this operation is given by the state complexity of the resulting language. It is also possible to consider the *nondeterministic* state complexity (nsc) of a language operation, and here one considers the number of states of a minimal NFA accepting the language. Many results are known for this topic, starting with the early work of Yu [3] and followed by others such as [4].

In this paper, we approach the problem by looking at the symmetric difference nondeterministic state complexity (xnsc) of unary regular languages. It is already known that there are interesting differences between xnsc and nsc. For example, the nsc for the complement of a regular language has a tight upper bound of 2^n [5], while the xnsc is simply $n + 1$ [6]. Here, we prove results on the xnsc of union, intersection, relative complement, and symmetric difference. Our witness languages are those unary regular languages which can be succinctly described by an XNFA; that is, the languages accepted by an n-state XNFA with an equivalent minimal DFA with $2^n - 1$ states. For these languages, we encode the minimal DFA as a primitive word over the alphabet $\{0, 1\}$, such that a 0 corresponds to a nonfinal state and a 1 corresponds to a final state. We note that these binary primitive words correspond to the cycles produced by primitive polynomials over the Galois field GF(2), which are used in random number generators [7], and thus display certain equidistribution properties [8]. We then consider the language operations as boolean operations on two binary primitive words, which may or may not lead to another primitive word. To establish whether the resulting word is primitive or not, we base our arguments on various properties of binary strings, such as slices and displacements, and the equidistribution properties of the resulting word.

In Sect. 2 we recall basic definitions for XNFA, primitive words, and the equidistribution properties of binary primitive words. Sect. 3 defines the properties of operations on primitive binary words, and in Sect. 4 we then establish the state complexity bounds.

2 PRELIMINARIES

We recap briefly some basic defintions, but assume that the reader is familiar with automata theory, such as in [9].

Let Σ be a finite alphabet, and Σ^* be the set of all strings over Σ, including the empty string ϵ. A DFA is a 5-tuple $(Q, \Sigma, \delta, q_0, F)$, where Q is a finite nonempty set of states, Σ is a finite alphabet, $q_0 \in Q$ is a start state, $F \subseteq Q$ is a set of final (accepting) states, and $\delta : Q \times \Sigma \to Q$ is the transition function. The transition function can be extended to strings, by $\delta(q, wa) = \delta(\delta(q, w), a)$ for $q \in Q$, $w \in \Sigma^*$ and $a \in \Sigma$. Then a string w (called a word) is accepted by

SAICSIT'18, September 2018, Port Elizabeth, South Africa

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

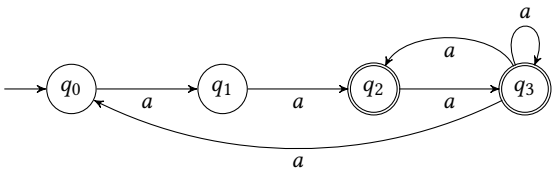
the DFA if $\delta(q_0, w) \in F$. The language accepted by the DFA is the set of all words accepted by the DFA.

Let 2^Q indicate the power set of a set Q. An NFA is similarly defined as a 5-tuple $(Q, \Sigma, \delta, Q_0, F)$, but may have more than one initial state, and the transition function is defined over sets of states: $\delta : 2^Q \times \Sigma \to 2^Q$. Again, δ can be extended to strings, and now acceptance of a word $w \in \Sigma^*$ follows if $\delta(Q_0, w) \cap F \neq \emptyset$. Any NFA $N = (Q, \Sigma, \delta, Q_0, F)$ can be converted to an equivalent DFA $D = (2^Q, \Sigma, \delta^D, Q_0^D, F^D)$ that accepts the same language, by using the subset construction to establish the transition function δ^D of the DFA: $\delta^D(A, \sigma) = \bigcup_{q \in A} \delta(q, a)$, where $A \subseteq Q$ and $\sigma \in \Sigma$.

An XNFA is a parity automaton, in the sense that it accepts a word from Σ^* only if there is an odd number of paths leading to an accept state. It is therefore defined exactly as an NFA, but acceptance of a word $w \in \Sigma$ follows if $|\delta(Q_0, w) \cap F| \bmod 2 \neq 0$. Similarly, the subset construction to find the DFA equivalent to an XNFA is formalized as $\delta^D(A, \sigma) = \bigoplus_{q \in A} \delta(q, a)$, where $A \subseteq Q$ and $\sigma \in \Sigma$. Here, the symbol $\oplus$ indicates symmetric difference in the normal set-theoretic sense, where $A \oplus B = (A \cup B) \setminus (A \cap B)$ for any two sets A and B.

An example of the difference between the subset construction for an NFA and an XNFA is given below.

Example 2.1. Let N be the unary NFA below.



The transition function that results from applying the subset construction to N is δ_{DFA}, shown below on the left. The initial state is $[q_0]$ and the final states are indicated with a $*$. However, if N is an XNFA, the transition function that results from applying the subset construction is δ_{XDFA}, shown on the right below. Compare for example the transitions from state $[q_0, q_2, q_3]$ below. In the NFA case, $\delta([q_0, q_2, q_3], a) = [q_0, q_1, q_2, q_3]$, since the union of the sets of transitions is calculated. But, in the XNFA case, the symmetric difference is calculated, and hence $\delta([q_0, q_2, q_3], a) = [q_0, q_1, q_2]$. Also, note that states $[q_0, q_2, q_3]$ and $[q_1, q_2, q_3]$ are not final states in the XNFA case, due to parity acceptance.

δ_{DFA}	a
$[q_0]$	$[q_1]$
$[q_1]$	$[q_2]$
$[q_2]*$	$[q_3]$
$[q_3]*$	$[q_0, q_2, q_3]$
$[q_0, q_2, q_3]*$	$[q_0, q_1, q_2, q_3]$
$[q_0, q_1, q_2, q_3]*$	$[q_0, q_1, q_2, q_3]$

δ_{XDFA}	a
$[q_0]$	$[q_1]$
$[q_1]$	$[q_2]$
$[q_2]*$	$[q_3]$
$[q_3]*$	$[q_0, q_2, q_3]$
$[q_0, q_2, q_3]$	$[q_0, q_1, q_2]$
$[q_0, q_1, q_2]*$	$[q_1, q_2, q_3]$
$[q_1, q_2, q_3]$	$[q_0]$

□

From [10] recall that the transition table of a unary XNFA can be written as a binary matrix, and that matrix multiplication over the Galois field GF(2) represents the behaviour of the XNFA. Moreover, for a unary XNFA, the matrix M has a so-called characteristic polynomial $c(X) = \det(XI - M)$. Also, M is said to be the companion matrix of $c(X)$.

Each $c(X)$ over GF(2) is associated with a certain cycle structure. Specifically, the properties of the characteristic polynomial of a unary XNFA N allow conclusions about the possible length of the cycle of states of its equivalent DFA N_D (see [11] in particular, as well as for example [10, 12, 13]). The choice of initial states for an XNFA determines which cycle in its polynomial cycle structure is the equivalent DFA.

In the rest of this paper, we consider only unary XNFA with non-singular matrices, whose cycle structures do not include transient heads; that is, states that are only reached once before a cycle is reached. By Lemma 1 of [11], this means that only matrices with a characteristic polynomial $c(X)$ that does not have X as a factor, are considered. For a given n-state XNFA N, if $c(X)$ is a primitive polynomial of order n over GF(2), then the minimal DFA equivalent to N has $2^n - 1$ states [10].

Example 2.2. In Example 2.1 above, the GF(2) matrix corresponding to the XNFA is given by

$$M = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix}.$$

The corresponding characteristic polynomial of M is $c(X) = X^4 + X^3 + X^2 + 1 = (X^3 + X + 1)(X + 1)$. Hence $c(X)$ is not primitive, and the cycle length of the DFA is only 7 (and not 15). □

Recall that a primitive word w over an alphabet Σ is a word that cannot be written in the form $w = u^k$, for any substring u of w and $k \geq 0$. Hence, the length of a primitive word is equal to its period—if u is the shortest string such that $w = u^k$, and u has length p, then w has period p [14].

Unary DFA cycles behave similarly to linear feedback shift registers (LFSRs) [1], and consequently, XNFA whose characteristic polynomials are primitive polynomials or products of primitive polynomials have good equidistribution properties [7, 8]. Specifically, Wang and Compagner [7] refer to three randomness properties, namely, **R1**, **R2** and **R3**, which were first introduced by Golomb in [15]. They show that the difference in these properties between maximal length cycles derived from primitive polynomials and those derived from products of primitive polynomials are negligible. A detailed discussion of these properties is beyond the scope of this work, but it suffices to say that they guarantee that in the applicable DFA cycles, the number of final states are only one more than the number of non-final states, and they limit the length of so-called runs of final and non-final states, leading to a uniform distribution of final states in the cycles.

As noted previously, the language accepted by an XNFA can be represented by a binary string s which corresponds to the cycle of the equivalent minimal DFA. For example, the string 10111 represents the language $\mathcal{L} = a^{5i} + j$, for $j \in \{0, 2, 3, 4\}$ and $i \geq 0$. Since the DFA is minimal, s must be a primitive string, and in the case presented here, it has good equidistribution properties.

3 PROPERTIES OF BINARY STRINGS

Note that the strings s and s^y have the same period, and any circular shift of a string s has the same period as s.

The following definition introduces the notion of q-slices of strings, which will be useful when reasoning about the periods of certain strings.

Definition 3.1. Let $s = s_0 s_1 \cdots s_{n-1}$ be a string of length n and let $dk = n$. Now, let

$$\chi_d(s,q) = s_{0+(q \bmod d)} s_{d+(q \bmod d)} s_{2d+(q \bmod d)} \cdots s_{(k-1)d+(q \bmod d)} .$$

We say that $\chi_d(s,q)$ is the q-slice of s with respect to d. The length of $\chi_d(s,q)$ is $k = \frac{n}{d}$. □

Note that if $q < d$, one can write

$$\chi_d(s,q) = s_{0+q} s_{d+q} s_{2d+q} \cdots s_{(k-1)d+q} .$$

This observation leads to the following lemma.

Lemma 3.2. *Let $s = s_0 s_1 \cdots s_{n-1}$ be a string of length n. Then for any q and $q' = q \bmod d$,*

$$\chi_d(s,q) = \chi_d(s,q') .$$

Proof. One can simply note that

$$\begin{aligned}\chi_d(s,q) &= s_{0+(q \bmod d)} s_{d+(q \bmod d)} s_{2d+(q \bmod d)} \cdots s_{(k-1)d+(q \bmod d)} \\ &= s_{0+q'} s_{d+q'} s_{2d+q'} \cdots s_{(k-1)d+q'} \\ &= \chi_d(s,q') .\end{aligned}$$

□

Essentially, a q-slice with respect to d divides the string into equal parts of length d and constructs a new string by taking the q-th element from each part.

Example 3.3. Let $s_1 = 101101101$, $s_2 = 101101011$ and let $s_3 = 110111101111011$, so that $n_1 = n_2 = 9$ and $n_3 = 15$. Some examples of q-slices are as follows:

$$\chi_3(s_1,1) = 000 \qquad \chi_3(s_3,1) = 11011$$
$$\chi_3(s_2,1) = 001 \qquad \chi_5(s_3,2) = 000$$

Figure 1 illustrates how $\chi_3(s_1,1)$ is found. The positions of the dashed lines that divide the strings into sections of equal length are determined by the value of d, while the position of the blocks within each section is a consequence of the choice of q.

□

The next lemmas establish relationships that exist between the periods of q-slices of related strings. For the proofs, please refer to the appendix.

Lemma 3.4. *Let t be a string of length p and let $df = p$. Then,*

$$\chi_d(t^r,q) = \chi_d(t,q)^r .$$

Proof. Since $\chi_d(t^r,q) = \chi_d(t^r,q')$ for some $q' < d$ by Lemma 3.2, one may assume, without loss of generality, that $q < d$. Let $t = t_0 t_1 \cdots t_{p-1}$. Then

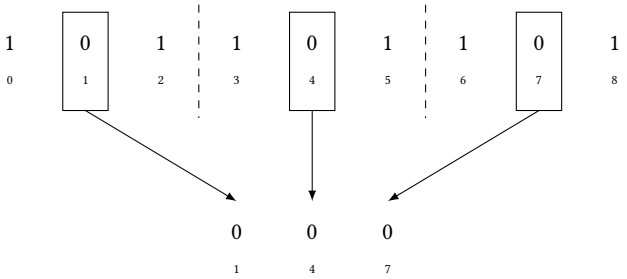


Figure 1: Example 3.3: $\chi_3(s_1,1)$, with $d = 3$ and $q = 1$

$$\begin{aligned}t^r &= (t_0 t_1 \cdots t_{p-1})^r \\ &= \underbrace{(t_0 t_1 \cdots t_{p-1})(t_0 t_1 \cdots t_{p-1}) \cdots (t_0 t_1 \cdots t_{p-1})}_{r \text{ times}} .\end{aligned}$$

Also,

$$\chi_d(t,q) = (t_{0+q} t_{d+q} t_{2d+q} \cdots t_{(f-1)d+q}) .$$

Let n be the length of t^r, and hence $pr = n$. From Definition 3.1, it is known that the length of $\chi_d(t,q)$ is $\frac{p}{d} = f$. Then

$$\begin{aligned}\chi_d(t^r,q) &= \underbrace{(t_{0+q} t_{d+q} t_{2d+q} \cdots t_{(f-1)d+q}) \cdots (t_{0+q} t_{d+q} t_{2d+q} \cdots t_{(f-1)d+q})}_{r \text{ times}} \\ &= (t_{0+q} t_{d+q} t_{2d+q} \cdots t_{(f-1)d+q})^r \\ &= \chi_d(t,q)^r .\end{aligned}$$

□

Lemma 3.5. *If a string s has period p, then for any q, the q-slice of s with respect to p has period 1. Conversely, if any q-slice with respect to d of a string s has period 1, then s has period p such that $p \mid d$.*

Proof. Let s have period p. Then there is some string $t = t_0 t_1 \cdots t_{p-1}$ of length p such that $s = t^x$ for some x. Let $0 \le q \le p-1$, then

$$\begin{aligned}\chi_p(s,q) &= s_q s_{p+q} s_{2p+q} \cdots s_{(x-1)p+q} \\ &= t_{q \bmod p} t_{(p+q) \bmod p} t_{(2p+q) \bmod p} \cdots t_{[(x-1)p+q] \bmod p} \\ &= \underbrace{t_q t_q t_q \cdots t_q}_{x \text{ times}} ,\end{aligned}$$

which clearly has period 1.

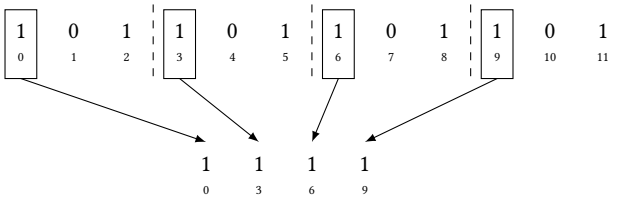
Conversely, let $\chi_d(s,q)$ have period 1 for any q. By Lemma 3.2, it is only necessary to consider $0 \le q \le d-1$. Then it follows that, for any $0 \le q \le d-1$,

$$s_q = s_{d+q} = s_{2d+q} = \cdots = s_{(\frac{n}{d}-1)d+q} .$$

Let $w = (s_0 s_1 \ldots s_{d-1})$, then $s = (s_0 s_1 \ldots s_{d-1})^{\frac{n}{d}} = w^{\frac{n}{d}}$. Let w have period p, then $p \mid d$, and since a repetition of a string has the same period as the original string, s has period p. □

Lemma 3.5 is illustrated in the next example.

Example 3.6. Let $s = 101101101101$ so that $n = 12$ and $p = 3$; that is $s = (101)^4$. Then, for example, one has $\chi_3(s, 0) = 1111$. Note that the period of $\chi_3(s, q)$ is 1. The figure below illustrates how $\chi_3(s, 0)$ is found:



□

Lemma 3.7. *If a string s of length n has period p and if $df = p$, then for any q, the q-slice of s with respect to d has period f' such that $f' \mid f$.*

Proof. Since s has period p, there is some primitive string $t = t_0 t_1 \cdots t_{p-1}$ of length p such that $s = t^{\frac{n}{p}}$. Let $dk = n$ and let $0 \leq q \leq d-1$, then

$$\chi_d(s, q) = s_{0+q} s_{d+q} s_{2d+q} \cdots s_{(k-1)d+q},$$

which has length $\frac{n}{d} = k$. But $s = t^{\frac{n}{p}}$, so by Lemma 3.4,

$$\begin{aligned}\chi_d(s, q) &= \chi_d(t^{\frac{n}{p}}, q) \\ &= \chi_d(t, q)^{\frac{n}{p}} .\end{aligned}$$

Since a repetition of a string has the same period as the original string, the period of $\chi_d(s, q)$ is equal to the period of $\chi_d(t, q)$. We know that the length of $\chi_d(s, q)$ is $\frac{n}{d} = k$, and hence the length of $\chi_d(t, q)^x$ must also be k. Let the length of $\chi_d(t, q)$ be y, then $(\frac{n}{p})(y) = k$, and so

$$y = \frac{pk}{n} = \frac{p}{d} = f .$$

Therefore, the length of $\chi_d(t, q)$ is f, and hence the period of $\chi_d(t, q)$ – and therefore the period of $\chi_d(s, q)$ – must be some f' such that $f' \mid f$. □

Example 3.8. Let $s = (110000111011)^3$, so that $n = 36$ and $p = 12$. Let $d = 2$ and $f = 6$, so that $df = p$. Then

$$\begin{aligned}\chi_2(s, 0) &= (100111)^3 \\ \chi_2(s, 1) &= (100101)^3 .\end{aligned}$$

In both cases, the period is 6, which divides $f = 6$. □

The notion of k-displacements of a string is now introduced. A k-displacement is a specific kind of permutation of a string which results in an ordered rearrangement of the string.

Definition 3.9. Let $\gcd(k, n) = 1$ and $\kappa(i, k) = i \cdot k \mod n$, and let s be a string of length n. A k-displacement of s is a permutation s' of s where the value at index i in s' is the value at index $\kappa(i, k)$ in s.

Example 3.10. Let $s = 011100011$, so that $n = 9$. Represent the indices of s as (0 1 2 3 4 5 6 7 8). Now let $k = 4$, and let s' be the 4-displacement of s, so that $\kappa(i, 4) = (i \cdot 4) \mod 9$. The value for s' at each index is computed by determining the corresponding index of s. For example, $\kappa(4, 4) = (4 \cdot 4) \mod 9 = 7$.

It follows that s' corresponds to (0 4 8 3 7 2 6 1 5), which is an ordered arrangement of the indices of s, and so $s' = 001111010$.

Figure 2 illustrates the calculation of the k-displacement. Here κ is computed modulo 9. At the top of the figure, therefore, one has repeated instances of s, separated by dashed lines, with the corresponding index indicated below each value. In the representation of s' below, the difference modulo 9 between adjacent indices is 4. The k-displacements can be seen as slices of extended strings, where d has the value of k and q is fixed at 0. The terminology k-displacements aims to emphasise the fact that the resulting string is a permutation of the original, with the indices displaced so that the difference modulo n between adjacent indices is k.

□

A shifted k-displacement is similar to a k-displacement, but the original string is shifted before the ordered rearrangement is computed.

Definition 3.11. Let s be a string of length n. A shifted k-displacement of s is a permutation s_k of s where the value at index i in s_k is the value at index $\kappa_q(i, k)$ in s, where $\gcd(k, n) = 1$ and $\kappa_q(i, k) = (i \cdot k + q) \mod n$ for some q.

Since $(i \cdot k + q) \mod n = [i \cdot k \mod n + q \mod n] \mod n$, a shifted k-displacement is a shift of a k-displacement.

The following lemma is concerned with the relation between indices of a string s and the indices of any repeating substring w of s.

Lemma 3.12. *Let $q \mid n$ and let s be a string of length n and w a string of length q such that $s = w^{\frac{n}{q}}$. Then for any $0 \leq i, j \leq n-1$, if $i \bmod q = j \bmod q$, then $s_i = s_j$. Conversely, let s be a string of length n, where for some q, it is true that $s_i = s_j$ for any $0 \leq i, j \leq n-1$ such that $i \bmod q = j \bmod q$. Then s has a substring w of length q such that $s = w^{\frac{n}{q}}$.*

Proof. First, let $s = w^{\frac{n}{q}}$, and let $w = w_0 w_1 \cdots w_{q-1}$. Then $s = (w_0 w_1 \cdots w_{q-1})^{\frac{n}{q}}$, and hence $s_i = w_{i \bmod q}$ and $s_j = w_{j \bmod q}$. Hence, for any $0 \leq i, j \leq n-1$, if $i \bmod q = j \bmod q$, then $s_i = w_{i \bmod q} = w_{j \bmod q} = s_j$.

Conversely, let s be a string of length n, where for some q, it is true that $s_i = s_j$ for any $0 \leq i, j \leq n-1$ such that $i \bmod q = j \bmod q$. Then, for $0 \leq k \leq q-1$, it follows that $s_k = s_{q+k} = s_{2q+k} = \cdots = s_{(\frac{n}{q}-1)q+k}$, and hence $s = (s_0 s_1 \cdots s_{q-1})^{\frac{n}{q}}$. Let $w = (s_0 s_1 \cdots s_{q-1})$, then $s = w^{\frac{n}{q}}$. □

The following lemma relates to the period of k-displacements and shifted k-displacements.

Lemma 3.13. *The period of a string is equal to the period of any k-displacement or shifted k-displacement of that string.*

Proof. Let $s = s_0 s_1 \cdots s_{n-1}$ be a string of length n and let p be its period. Then $s = t^{\frac{n}{p}}$, where t is a string of length p. Hence, by Lemma 3.12, the values at any indices i and j of s are equal if $i \bmod p = j \bmod p$. The value at any index r of a k-displacement s' of s is equal to the value of s at $\kappa(r, k)$. Without loss of generality,

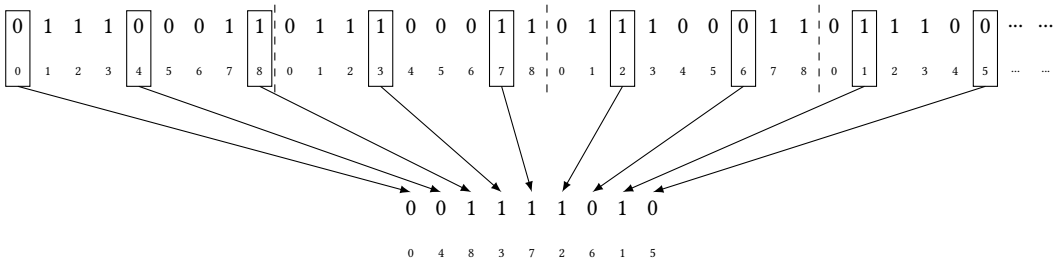


Figure 2: Example 3.10: $\kappa(i, 4) = (i \cdot 4) \mod 9$

one can assume that $k < n$. Now, since $p \mid n$, it follows that $i \bmod n = xp + (i \bmod p)$ for some x. Hence, for $0 \le i, j \le n-1$, if $i \bmod p = j \bmod p$, then

$$\begin{aligned}
\kappa(i,k) \bmod p &= (i \cdot k \bmod n) \bmod p \\
&= ([(i \bmod n)(k \bmod n)] \bmod n) \bmod p \\
&= ([(xp + (i \bmod p))(k \bmod n)] \bmod n) \bmod p \\
&= ([(xp + (j \bmod p))(k \bmod n)] \bmod n) \bmod p \\
&= ([(j \bmod n)(k \bmod n)] \bmod n) \bmod p \\
&= (j \cdot k \bmod n) \bmod p \\
&= \kappa(j,k) \bmod p\ .
\end{aligned}$$

That is, if $i \bmod p = j \bmod p$, then $\kappa(i,k) \bmod p = \kappa(j,k) \bmod p$ as long as $\gcd(k,n) = 1$. But $\kappa(i,k)$ and $\kappa(j,k)$ are indices for which the values of s are equal to the values of s' at i and j, respectively. Hence, for $0 \le i, j \le n-1$, if $i \bmod p = j \bmod p$, then $s'_i = s'_j$.

Therefore, by Lemma 3.12, $s' = w^{\frac{n}{p}}$, where w has length p. Since the repetition of a string has the same period as the original string, the period of s' is equal to the period p' of w, where $p' \mid p$. Stated generally, the period of a k-displacement of a string divides the period of the string.

Recall that k is chosen so that $\gcd(n,k) = 1$, and therefore it has a modular multiplicative inverse[1] k^{-1} so that $(k \cdot k^{-1}) \bmod n = 1$. Let s'' be a k^{-1}-displacement of s'. That is, given some index r of s'', its value is equal to the value of s' at $\kappa(r, k^{-1})$. Furthermore, given any index $\kappa(r, k^{-1})$ of s', its value is equal to the value of s at $\kappa(\kappa(r, k^{-1}), k)$. However,

$$\begin{aligned}
\kappa(\kappa(r,k^{-1}),k) &= \kappa([r \cdot k^{-1} \bmod n], k) \\
&= (r \cdot k^{-1} \bmod n)k \bmod n \\
&= (r \cdot k^{-1} \bmod n)(k \bmod n) \bmod n \\
&= (r \cdot k^{-1} \cdot k) \bmod n \\
&= (r \bmod n)(k^{-1} \cdot k \bmod n) \bmod n \\
&= r \bmod n \\
&= r
\end{aligned}$$

Hence, $s'' = s$, and so s is a k^{-1}-displacement of s'. Therefore, the period p of s must divide p'. Since $p' \mid p$ and $p \mid p'$, it follows that $p' = p$, and hence the period of s' is p. Furthermore, shifting a string does not affect its period, therefore the same is true for any s' that is a shifted k-displacement of s. □

Example 3.14. Let $s = 011100011$. Then its 4-displacement $s' = 001111010$, and its 3-shifted 4-displacement $s'' = 010001111$. Note that s, s' and s'' all have period 9.

Let $r = 110110110 = (110)^3$ with length 9 and period 3. Let $k = 2$ and $q = 5$, so that the 2-displacement of r is $r' = 101101101 = (101)^3$, and its 5-shifted 2-displacement is $r'' = 011011011 = (011)^3$. Then r, r' and r'' all have equal period 3. For example, for r', one can compute $\kappa(3,2) = (3 \cdot 2) \bmod 9 = 6$, and for r'' it holds that $\kappa_5(3,2) = (3 \cdot 2 + 5) \bmod 9 = 2$. □

Finally, the following lemma shows a specific relationship between q-slices and k-displacements.

Lemma 3.15. *Let $s = t^k$ be a string of length pk with period p, where $\gcd(p,k) = 1$. Let $p = p_\alpha p_\beta \neq 1$ and $k = k_\alpha k_\beta \neq 1$. Then*

$$\chi_{p_\alpha k_\alpha}(s,q) = (t')^{k_\beta}$$

where t' is a k_α-displacement of $\chi_{p_\alpha}(t,q)$.

Proof. We have

$$\chi_{p_\alpha}(t,q) = t_q t_{p_\alpha+q} t_{2p_\alpha+q} \cdots t_{(p_\beta-1)p_\alpha+q}\ .$$

Then the following is a k_α-displacement of $\chi_{p_\alpha}(t,q)$, since every index has the form $i \cdot k \bmod n$:

$$t' = t_{q \bmod p} t_{(p_\alpha k_\alpha+q) \bmod p} t_{(2p_\alpha k_\alpha+q) \bmod p} \cdots t_{[(p_\beta-1)p_\alpha k_\alpha+q] \bmod p}\ .$$

Furthermore, it holds that

$$\begin{aligned}
&[(p_\beta - 1)p_\alpha k_\alpha + q] \bmod p \\
=&([(p_\beta - 1)p_\alpha k_\alpha] \bmod p + [q \bmod p]) \bmod p \\
=&([p - p_\alpha k_\alpha] \bmod p + [q \bmod p]) \bmod p \\
=&[(-p_\alpha k_\alpha) \bmod p + (q \bmod p)] \bmod p \\
=&(-p_\alpha k_\alpha + q) \bmod p\ ,
\end{aligned}$$

and also

$$\begin{aligned}
&[(rp_\beta - 1)p_\alpha k_\alpha + q] \bmod p \\
=&([(rp_\beta - 1)p_\alpha k_\alpha] \bmod p + [q \bmod p]) \bmod p \\
=&([rp - p_\alpha k_\alpha] \bmod p + [q \bmod p]) \bmod p \\
=&[(-p_\alpha k_\alpha) \bmod p + (q \bmod p)] \bmod p \\
=&(-p_\alpha k_\alpha + q) \bmod p\ .
\end{aligned}$$

[1] Given $\mathbb{Z}_n$, namely the integers modulo n, any k has a multiplicative inverse k^{-1} if and only if $\gcd(n,k) = 1$. Then $(k \cdot k^{-1}) \bmod n = 1$.

Hence,

$$[(rp_\beta - 1)p_\alpha k_\alpha + q] \bmod p = [(p_\beta - 1)p_\alpha k_\alpha + q] \bmod p \ .$$

Consequently,

$$\begin{aligned}\chi_{p_\alpha k_\alpha}(s, q) &= s_q s_{p_\alpha k_\alpha + q} s_{2p_\alpha k_\alpha + q} \cdots s_{(p_\beta k_\beta - 1)p_\alpha k_\alpha + q} \\ &= (s_q s_{p_\alpha k_\alpha + q} s_{2p_\alpha k_\alpha + q} \cdots s_{(p_\beta - 1)p_\alpha k_\alpha + q})^{k_\beta} \\ &= (t_{q \bmod p} t_{(p_\alpha k_\alpha + q) \bmod p} t_{(2p_\alpha k_\alpha + q) \bmod p} \\ &\quad \cdots t_{[(p_\beta - 1)p_\alpha k_\alpha + q] \bmod p})^{k_\beta} \\ &= (t')^{k_\beta} \ .\end{aligned}$$

□

3.1 Bitwise binary operations on strings

Definition 3.16. Let $s^1 = s^1_0 s^1_1 \dots s^1_{n-1}$ and $s^2 = s^2_0 s^2_1 \dots s^2_{n-1}$ be two binary strings of length n. Then $s^- = s^-_0 s^-_1 \dots s^-_{n-1}$, where $s^-_i = 1$ if $s^1_i = 1$ and $s^2_i = 0$, and 0 otherwise. We say that s^- is the result of performing a bitwise DIFF operation on s^1 and s^2. □

The bitwise DIFF operation is analogous to the relative complement (or difference) set operation, which is defined as $A \setminus B$ for any sets A and B. Just as $A \setminus B$ contains only those elements of A that do not also occur in B, s^- contains only 1's in positions where a 1 occurs in s^1 but not also in s^2.

The following lemma will enable us to show that the proof of Theorem 3.18 is similar to those of Theorems 3.20 and 3.21.

Lemma 3.17. *Let s^1 and s^2 be two binary strings of length n. Let $s^\star = s^1 \star s^2$, where $\star$ is either the bitwise AND operator, the bitwise OR operator or the bitwise DIFF operator. Then*

- **AND.** *If $s^1 = 1^n$, $s^\wedge = 1^n$ if and only if $s^2 = 1^n$, and $s^\wedge = 0^n$ if and only if $s^2 = 0^n$. Furthermore, the period of $s^\wedge$ is equal to the period of s^2.*
- **OR.** *If $s^1 = 0^n$, $s^\vee = 0^n$ if and only if $s^2 = 0^n$, and $s^\vee = 1^n$ if and only if $s^2 = 1^n$. Furthermore, the period of $s^\vee$ is equal to the period of s^2.*
- **DIFF.** *If $s^1 = 1^n$, $s^- = 0^n$ if and only if $s^2 = 1^n$, and $s^- = 1^n$ if and only if $s^2 = 0^n$. Furthermore, the period of s^- is equal to the period of s^2.*

Proof. The lemma follows directly from the definitions of bitwise AND, bitwise OR and bitwise DIFF. □

The next theorem establishes that if the AND operation is applied to two primitive words, then the result is also a primitive word.

Theorem 3.18. *Let s^1 and s^2 be two primitive binary strings of length m and n, respectively, where $\gcd(m, n) = 1$ and hence $lcm(m, n) = mn$. Let $s^{1\prime} = (s^1)^n$ and $s^{2\prime} = (s^2)^m$, so that $s^{1\prime}$ and $s^{2\prime}$ are strings of length mn. Now, let $s^\wedge = s^{1\prime} \wedge s^{2\prime}$, i.e. $s^\wedge$ is the result of performing a bitwise AND operation on $s^{1\prime}$ and $s^{2\prime}$. Then $s^\wedge$ is a primitive string.*

Proof. Since the theorem concerns the bitwise AND operation between two strings, the positions of interest are those where a 1 occurs in both strings, since these are the only pairs of bits that will result in a 1 in $s^\wedge$.

The length of each of the strings $s^{1\prime}$, $s^{2\prime}$ and $s^\wedge$ is mn and hence the period of $s^\wedge$ must be such that it divides mn. We identify four exhaustive cases for possible divisors of mn and prove that $s^\wedge$ can only have period mn.

Case 1: $d = m$ (or $d = n$). Since the repetition of a string has the same period as the original string, it follows that $s^{1\prime}$ has period m. Then by Lemma 3.5 on page 3, $\chi_m(s^{1\prime}, q)$ has period 1 for any $0 \le q \le m - 1$. Let q be some index of s^1 such that $s^1_q = 1$. Then $\chi_m(s^{1\prime}, q) = 1^n$.
It also holds that

$$\begin{aligned}\chi_m(s^{2\prime}, q) &= s^{2\prime}_q s^{2\prime}_{m+q} s^{2\prime}_{2m+q} \cdots s^{2\prime}_{(n-1)m+q} \\ &= s^2_{q \bmod n} s^2_{(m+q) \bmod n} s^2_{(2m+q) \bmod n} \\ &\quad \cdots s^2_{[(n-1)m+q] \bmod n} \ .\end{aligned}$$

Since $\gcd(m, n) = 1$, $\chi_m(s^{2\prime}, q)$ is a shifted m-displacement of s^2. Therefore, by Lemma 3.13, the period of $\chi_m(s^{2\prime}, q)$ is equal to the period of s^2, which is n. Now, $\chi_m(s^\wedge, q) = \chi_m(s^{1\prime}, q) \wedge \chi_m(s^{2\prime}, q)$. Since $\chi_m(s^{1\prime}, q) = 1^n$ and $\chi_m(s^{2\prime}, q)$ has period n, by Lemma 3.17, the period of $\chi_m(s^\wedge, q)$ is n and not 1. Therefore, by Lemma 3.5, $s^\wedge$ cannot have period m. Similarly, $s^\wedge$ cannot have period n.

Case 2: d such that $d \mid m$ (or $d \mid n$). Let q be some index of s^1 such that $s^1_q = 1$.
If d is the period of $s^\wedge$, then $\chi_d(s^\wedge, q)$ has period 1. Since $m = dk$, $\chi_m(s^\wedge, q)$ is a substring of $\chi_d(s^\wedge, q)$ and hence must also have period 1. However, in **Case 1** we considered $\chi_m(s^\wedge, q)$ and showed that it must have period n, and not 1. This is a contradiction, and hence $s^\wedge$ does not have period d where $d \mid m$. Similarly, $s^\wedge$ does not have period d where $d \mid n$.

Case 3: $d = m_\alpha n_\alpha$ where $m_\alpha m_\beta = m$, $n_\alpha n_\beta = n$. If $m_\beta = n_\alpha = 1$, or if $m_\alpha = n_\beta = 1$, one has **Case 1**.
If $m_\alpha \ne 1$ and $m_\beta \ne 1$, while $n_\alpha = 1$, or if $n_\alpha \ne 1$ and $n_\beta \ne 1$, while $m_\alpha = 1$, one has **Case 2**.
Now consider the case where neither m_α, n_α nor n_β are equal to 1.
Assume that $s^\wedge$ has period $m_\alpha n_\alpha$. We first seek to establish that for any index of the form rm_α, one has $s^{1\prime}_{rm_\alpha} = 1$, in order to reason about the values of $s^{2\prime}$ at indices of the form rm_α.
Since $s^{1\prime}$ has period m and $s^{2\prime}$ has period n, there exists some q_0 so that $s^\wedge_{q_0} = s^{1\prime}_{q_0} \wedge s^{2\prime}_{q_0} = 1$, and hence $s^{1\prime}_{q_0} = s^{2\prime}_{q_0} = 1$. We can shift $s^{1\prime}$ and $s^{2\prime}$ by the same amount without affecting the period of the resulting $s^\wedge$, and so, without loss of generality, one may assume that $q_0 = 0$.
Now, by assumption, $\chi_{m_\alpha n_\alpha}(s^\wedge, q)$ has period 1, and specifically, $\chi_{m_\alpha n_\alpha}(s^\wedge, 0) = 1^{m_\beta n_\beta}$. Then, by Lemma 3.17,

$$\chi_{m_\alpha n_\alpha}(s^{1\prime}, 0) = \chi_{m_\alpha n_\alpha}(s^{2\prime}, 0) = 1^{m_\beta n_\beta} \ .$$

Furthermore, by Lemma 3.15, $\chi_{m_\alpha n_\alpha}(s^{1\prime}, 0) = (t^1)^{n_\beta}$, where t^1 is a n_α-displacement of $\chi_{m_\alpha}(s^1, 0)$. Consequently, since $\chi_{m_\alpha n_\alpha}(s^{1\prime}, 0) = 1^{m_\beta n_\beta}$ and consists only of the elements of $\chi_{m_\alpha}(s^1, 0)$, it follows that $\chi_{m_\alpha}(s^1, 0)$ consists entirely of 1's, and hence

$$s^1_0 = s^1_{m_\alpha} = s^1_{2m_\alpha} = \cdots = s^1_{(m_\beta - 1)m_\alpha} = 1 \ .$$

Now, $s^{1\prime} = (s^1)^n$, and hence

$$s_0^{1\prime} = s_{m_\alpha}^{1\prime} = s_{2m_\alpha}^{1\prime} = \cdots = s_{(nm_\beta-1)m_\alpha}^{1\prime} = 1\ .$$

Therefore, any q-slice of $s^{1\prime}$ with respect to $m_\alpha n_\alpha$ where $q = rm_\alpha$ is a string consisting of 1's:

$$\chi_{m_\alpha n_\alpha}(s^{1\prime}, rm_\alpha) = 1^{m_\beta n_\beta} \quad \text{for } 0 \le rm_\alpha \le m_\alpha n_\alpha - 1$$
$$\therefore \chi_{m_\alpha n_\alpha}(s^{1\prime}, rm_\alpha) = 1^{m_\beta n_\beta} \quad \text{for } 0 \le r \le n_\alpha - 1\ .$$

By assumption $\chi_{m_\alpha n_\alpha}(s^\wedge, rm_\alpha)$ has period 1, and hence from Lemma 3.17 it follows that $\chi_{m_\alpha n_\alpha}(s^{2\prime}, rm_\alpha)$ must have period 1 for $0 \le r \le n_\alpha - 1$.
But by Lemma 3.15, $\chi_{m_\alpha n_\alpha}(s^{2\prime}, rm_\alpha) = (t^2)^{m_\beta}$, where t^2 is an m_α-displacement of $\chi_{n_\alpha}(s^2, rm_\alpha)$,. Since the repetition of a string has the same period as the original string, it follows that t^2 has period 1, and so by Lemma 3.13, $\chi_{n_\alpha}(s^2, rm_\alpha)$ has period 1. Since by Lemma 3.2, $\chi_{n_\alpha}(s^2, rm_\alpha) = \chi_{n_\alpha}(s^2, rm_\alpha \bmod n_\alpha)$, it follows that $\chi_{n_\alpha}(s^2, rm_\alpha \bmod n_\alpha)$ must also have period 1. Recall that $0 \le r \le n_\alpha - 1$.
Now, $\gcd(m_\alpha, n_\alpha) = 1$ and so the following are unique:

$$\begin{aligned} &0,\\ &m_\alpha \bmod n_\alpha,\\ &2m_\alpha \bmod n_\alpha,\\ &\vdots\\ &(n_\alpha - 2)m_\alpha \bmod n_\alpha\\ \text{and } &(n_\alpha - 1)m_\alpha \bmod n_\alpha. \end{aligned}$$

Since there are n_α unique values of $rm_\alpha \bmod n_\alpha$, it follows that $\chi_{n_\alpha}(s^2, q)$ has period 1 for any $0 \le q \le n_\alpha - 1$, and hence by Lemma 3.2, for any q. But then, by Lemma 3.5, s^2 has period p such that $p \mid n_\alpha$. This is a contradiction, and therefore $s^\wedge$ does not have period $m_\alpha n_\alpha$.
Similarly, if neither n_α, m_α nor m_β are equal to 1, $s^\wedge$ does not have period $m_\alpha n_\alpha$.

Case 4: $d = mn$ This is the only remaining case, hence $s^\wedge$ has period mn.

□

Example 3.19. Let s^1 and s^2 be two primitive strings of length $m = 15$ and $n = 14$ respectively, with $s_0^1 = s_0^2 = 1$. Let $m_\alpha = 3$, $m_\beta = 5$, $n_\alpha = 7$ and $n_\beta = 2$. Furthermore, let $s^{1\prime} = (s^1)^n$ and $s^{2\prime} = (s^2)^m$, so that both $s^{1\prime}$ and $s^{2\prime}$ have length $mn = 210$.

Now, let us suppose that $s^\wedge = s^{1\prime} \wedge s^{2\prime}$ has period $m_\alpha n_\alpha = 21$. By Lemma 3.5, it follows that $\chi_{21}(s^\wedge, 0) = 1^{10}$. Now, $s_0^1 = s_0^2 = 1$, and so by Lemma 3.17, $\chi_{21}(s^{1\prime}, 0) = \chi_{21}(s^{2\prime}, 0) = 1^{10}$.

However, by Lemma 3.15, $\chi_{21}(s^{1\prime}, 0) = (t^1)^2$, where t^1 is a 7-displacement of $\chi_3(s^1, 0)$. Now, $\chi_3(s^1, 0) = s_0^1 s_3^1 s_6^1 s_9^1 s_{12}^1$. Its 7-displacement is therefore $s_0^1 s_6^1 s_{12}^1 s_3^1 s_9^1$. Consequently, $\chi_{21}(s^{1\prime}, 0) = (s_0^1 s_6^1 s_{12}^1 s_3^1 s_9^1)^2 = 1^{10}$. Hence, $s_0^1 = s_3^1 = s_6^1 = s_9^1 = s_{12}^1 = 1$.
Since $s^{1\prime} = (s^1)^n$, it follows that

$$s_0^{1\prime} = s_3^{1\prime} = s_6^{1\prime} = \cdots = s_{207}^{1\prime}\ .$$

Therefore, specifically for $0 \le r < 7$, it follows that $\chi_{21}(s^{1\prime}, 3r) = 1^{10}$. Hence, by Lemma 3.17, $\chi_{21}(s^{2\prime}, 3r)$ has period 1 for $0 \le r < 7$.

However, by Lemma 3.15, $\chi_{21}(s^{2\prime}, 3r) = (t^2)^5$, where t^2 is a 3-displacement of $\chi_7(s^2, 3r)$. Since s and s^y have the same period for any string s, the period of $\chi_{21}(s^{2\prime}, 3r)$ is equal to the period of t^2, which is equal to the period of $\chi_7(s^2, 3r)$ by Lemma 3.13, and hence equal to the period of $\chi_7(s^2, 3r \bmod 7)$ by Lemma 3.2. Then, $\chi_7(s^2, 3r \bmod 7)$ has period 1 for 7 unique values of $q = 3r \bmod 7$, and hence for any $0 \le q < 7$, and consequently by Lemma 3.2 for any q. Then, by Lemma 3.5, s^2 has some period p such that $p \mid 7$. But s^2 has period 14, and hence this is a contradiction. □

The same result holds for the union and the relative complement. The proofs are given in the appendix.

Theorem 3.20. *Let s^1 and s^2 be two primitive binary strings of length m and n, where $\gcd(m, n) = 1$ and hence $lcm(m, n) = mn$. Let $s^{1\prime} = (s^1)^n$ and $s^{2\prime} = (s^2)^m$, so that s_1' and s_2' are strings of length mn. Now, let $s^\vee = s^{1\prime} \vee s^{2\prime}$, i.e. $s^\vee$ is the result of performing a bitwise OR operation on $s^{1\prime}$ and $s^{2\prime}$. Then $s^\vee$ is a primitive string.*

Proof. Since the theorem concerns the bitwise OR operation between two strings, one need only consider those positions where a 0 occurs in both strings, since these are the only pairs of bits that will result in a 0 in $s^\vee$. In the proof for Theorem 3.18, the selection of q-slices was limited to those where 1's occur in $s^{1\prime}$ in order to reason about 1's occurring in $s^{2\prime}$. A proof for the bitwise OR case would be identical, except that the selection of q-slices would be limited to those where 0's occur in $s^{1\prime}$ in order to reason about 0's occurring in $s^{2\prime}$. In particular, the proof relies similarly on Lemma 3.17, but simply invokes a different case. □

Theorem 3.21. *Let s^1 and s^2 be two primitive binary strings of length m and n, where $\gcd(m, n) = 1$ and hence $lcm(m, n) = mn$. Let $s^{1\prime} = (s^1)^n$ and $s^{2\prime} = (s^2)^m$, so that s_1' and s_2' are strings of length mn. Now, let $s^- = s^{1\prime} - s^{2\prime}$, i.e. s^- is the result of performing a bitwise DIFF operation on $s^{1\prime}$ and $s^{2\prime}$. Then s^- is a primitive string.*

Proof. Since the theorem concerns the bitwise DIFF operation between two strings, the positions of interest are those where a 1 occurs in $s^{1\prime}$ and a 0 occurs in $s^{2\prime}$, since these are the only pairs of bits that will result in a 1 in s^-. As in the case of Theorem 3.20, the proof is similar to that of Theorem 3.18, relying on Lemma 3.17 but invoking a different case. □

Having established the previous three theorems, one can now apply them to operations on languages represented by XNFA.

4 STATE COMPLEXITY OF LANGUAGE OPERATIONS OF XNFA

Unary DFA cycles have similar behaviour to linear feedback shift registers (LFSRs) [1], and consequently, XNFA whose characteristic polynomials are primitive polynomials or products of primitive polynomials have good equidistribution properties [7, 8]. Specifically, Wang and Compagner [7] refer to three randomness properties, **R1**, **R2** and **R3**, which were first introduced by Golomb in [15]. They show that the difference in these properties between maximal length cycles derived from primitive polynomials and those derived from products of primitive polynomials are negligible. A detailed discussion of these properties is beyond the scope of this work, but

it suffices to say that they guarantee that in the applicable DFA cycles, the number of final states are only one more than the number of non-final states, and they limit the length of so-called runs of final and non-final states, leading to a uniform distribution of final states in the cycles.

As noted before, the language accepted by an XNFA can be represented by a binary string s which corresponds to the cycle of the equivalent minimal DFA. Since the DFA is minimal, s must be a primitive string, and in the case presented here, it has good equidistribution properties. These characteristics can be used to show that $m + n$, a lower bound on the descriptional complexity of the intersection, union and relative complement language operations for unary XNFA, cannot be reached for all m and n.

First, it is shown that $(2^m - 1)(2^n - 1)$ is the upper bound on the operation of intersection applied to languages represented minimally by two XNFA with m and n states, respectively.

Theorem 4.1. *Let $N^1 = (Q^1, \Sigma, \delta^1, Q_0^1, F^1)$ be an m-state and $N^2 = (Q^2, \Sigma, \delta^2, Q_0^2, F^2)$ be an n-state unary XNFA, respectively, and let $2^m - 1$ and $2^n - 1$ be relatively prime. Furthermore, let $Q_0^1 = Q_0^2 = F^1 = F^2 = \{q_0\}$. Let $\mathcal{L}^1$ and $\mathcal{L}^2$ be the languages recognised by N^1 and N^2, respectively. Then the minimal DFA that accepts $\mathcal{L} = \mathcal{L}^1 \cap \mathcal{L}^2$ has $(2^m - 1)(2^n - 1)$ states.*

Proof. First, let s^1 be the binary string of length $2^m - 1$ representing $\mathcal{L}^1$, corresponding to its DFA cycle of length $2^m - 1$, and let s^2 be the binary string of length $2^n - 1$ representing $\mathcal{L}^2$, corresponding to its DFA cycle of length $2^n - 1$. Now, let $s^{1\prime} = (s^1)^{2^n - 1}$ and let $s^{2\prime} = (s^2)^{2^m - 1}$. Then $s^\wedge = s^{1\prime} \wedge s^{2\prime}$, which has length $(2^m - 1)(2^n - 1)$, would represent $\mathcal{L} = \mathcal{L}^1 \cap \mathcal{L}^2$. Note that since $(2^m - 1)$ and $(2^n - 1)$ are relatively prime, $\text{lcm}(2^m - 1, 2^n - 1) = (2^m - 1)(2^n - 1)$, and hence no larger DFA cycle for the intersection of $\mathcal{L}^1$ and $\mathcal{L}^2$ would be minimal. Since $\gcd(2^m - 1, 2^n - 1) = 1$, by Theorem 3.18, $s^\wedge$ is primitive, and hence a DFA cycle recognising $\mathcal{L}$ must have exactly $(2^m - 1)(2^n - 1)$ states. Hence, $(2^m - 1)(2^n - 1)$ is an upper bound. □

The following lemma allows us to conclude that for any language that requires a cycle of length $(2^m - 1)(2^n - 1)$ in its minimal DFA, $m + n$ is a lower bound on the descriptional complexity of the equivalent XNFA representing that language.

Lemma 4.2. *For $m, n \geq 2$, a minimal DFA cycle of length $(2^m - 1)(2^n - 1)$ requires at least $m + n$ states in an equivalent XNFA.*

Proof. Let $r = m+n-1$. Then the largest cycle length achievable by an XNFA with r states is $2^r - 1$. However,

$$\begin{aligned}(2^m - 1)(2^n - 1) &= 2^{m+n} - 2^m - 2^n + 1 \\ &> 2^{m+n} - (2^m + 2^n) \\ &\geq 2^{m+n} - 2^{m+n-1} \\ &= 2^{m+n-1} \\ &> 2^{m+n-1} - 1 \\ &= 2^r - 1\,.\end{aligned}$$

Hence, at least $m + n$ XNFA states are necessary to achieve a cycle of length $(2^m - 1)(2^n - 1)$. □

The following theorem states that for certain XNFA with m states and n states, respectively, an XNFA with $m + n$ states is insufficient for representing the intersection of the two languages represented by the XNFA.

Theorem 4.3. *Let $N^1 = (Q^1, \Sigma, \delta^1, Q_0^1, F^1)$ be an m-state and $N^2 = (Q^2, \Sigma, \delta^2, Q_0^2, F^2)$ be an n-state unary XNFA, respectively, and let $2^m - 1$ and $2^n - 1$ be relatively prime. Let M_1 and M_2, their respective transition matrices, be the non-singular normal form matrices of primitive polynomials $c_m(X)$ and $c_n(X)$. Furthermore, let $Q_0^1 = Q_0^2 = F^1 = F^2 = \{q_0\}$. Let $\mathcal{L}^1$ and $\mathcal{L}^2$ be the languages recognised by N^1 and N^2, respectively. Then the smallest XNFA that accepts $\mathcal{L} = \mathcal{L}^1 \cap \mathcal{L}^2$ must have more than $m + n$ states.*

Proof. As shown in the proof of Theorem 4.1, the equivalent DFA that accepts $\mathcal{L}$ has $(2^m - 1)(2^n - 1)$ states. By Lemma 4.2, the cycle length $(2^m - 1)(2^n - 1)$ in an DFA requires at least $m + n$ states in an equivalent XNFA.

Any XNFA with exactly $m + n$ states that leads to an equivalent DFA that is a cycle of length $(2^m - 1)(2^n - 1)$ must have a characteristic polynomial that is reducible and has two primitive polynomial factors $c_m(X)$ and $c_n(X)$ of degree m and n respectively [12]. Such an XNFA is equivalent to the XNFA whose transition matrix contains two blocks that are the normal form matrices of $c_m(X)$ and $c_n(X)$, respectively. Let $c(X) = c_m(X)c_n(X)$, which is the polynomial of least degree with factors $c_m(X)$ and $c_n(X)$, and let N be an XNFA with characteristic polynomial $c(X)$ whose transition matrix is the block diagonal matrix of $c(X)$. One may assume that the initial states of N are chosen in such a way that its equivalent DFA is the appropriate cycle of length $(2^m - 1)(2^n - 1)$.

Given any choice of final states, the cycle of length $(2^m - 1)(2^n - 1)$ in the cycle structure of $c(X)$ has good equidistribution properties [8], and specifically the property **R1**, which is that the number of non-final states is only one less than the number of final states [7]. That is, the language $\mathcal{L}_N$ accepted by N is equidistributed.

However, according to [8] and [7], the languages $\mathcal{L}^1$ and $\mathcal{L}^2$ are similarly equidistributed. This means that, given strings $s^{1\prime}$ and $s^{2\prime}$ and considering pairs of the form $(s_i^{1\prime}, s_i^{2\prime})$, the pairs $(0, 0)$, $(0, 1)$, $(1, 0)$ and $(1, 1)$ are equidistributed over the length of the strings. Therefore, $s^\wedge$, the result of the bitwise AND operation on $s^{1\prime}$ and $s^{2\prime}$, does not have the property **R1**, since occurrences of the pairs $(0, 0)$, $(0, 1)$, $(1, 0)$ result in a 0 in $s^\wedge$, while only occurrences of the pair $(1, 1)$ result in a 1 in $s^\wedge$. Hence, $\mathcal{L}_N \neq \mathcal{L}$, and consequently, no XNFA with $m + n$ states, which leads to a cycle of length $(2^m - 1)(2^n - 1)$ with good equidistribution properties, will accept $\mathcal{L}$. □

The results for union and relative complement follow in the same manner:

Theorem 4.4. *Let $N^1 = (Q^1, \Sigma, \delta^1, Q_0^1, F^1)$ be an m-state and $N^2 = (Q^2, \Sigma, \delta^2, Q_0^2, F^2)$ be an n-state unary XNFA, respectively, and let $2^m - 1$ and $2^n - 1$ be relatively prime. Let M_1 and M_2, their respective transition matrices, be the non-singular normal form matrices of primitive polynomials $c_m(X)$ and $c_n(X)$. Furthermore, let $Q_0^1 = Q_0^2 = F^1 = F^2 = \{q_0\}$. Let $\mathcal{L}^1$ and $\mathcal{L}^2$ be the languages recognised by N^1 and N^2, respectively. Then the minimal DFA that accepts*

$\mathcal{L} = \mathcal{L}^1 \cup \mathcal{L}^2$ has $(2^m - 1)(2^n - 1)$ states, while the smallest XNFA requires more than $m + n$ states.

Proof. The proof is similar to the proof for Theorem 4.3, except that it is proven that the minimal DFA requires $(2^m - 1)(2^n - 1)$ on the basis of Theorem 3.20. As for the intersection case, $\mathcal{L} = \mathcal{L}^1 \cup \mathcal{L}^2$ is not an equidistributed language, since occurrences of the pairs $(0, 1)$, $(1, 0)$ and $(1, 1)$ result in a 1 in $s^{\vee}$, while only occurrences of the pair $(0, 0)$ result in a 0. Hence, no XNFA with $m + n$ states, which leads to a cycle of length $(2^m - 1)(2^n - 1)$ with good equidistribution properties, will accept $\mathcal{L}$. □

Theorem 4.5. *Let $N^1 = (Q^1, \Sigma, \delta^1, Q_0^1, F^1)$ be an m-state and $N^2 = (Q^2, \Sigma, \delta^2, Q_0^2, F^2)$ be an n-state unary XNFA, respectively, and let $2^m - 1$ and $2^n - 1$ be relatively prime. Let M_1 and M_2, their respective transition matrices, be the non-singular normal form matrices of primitive polynomials $c_m(X)$ and $c_n(X)$. Furthermore, let $Q_0^1 = Q_0^2 = F^1 = F^2 = \{q_0\}$. Let $\mathcal{L}^1$ and $\mathcal{L}^2$ be the languages recognised by N^1 and N^2, respectively. Then the minimal DFA that accepts $\mathcal{L} = \mathcal{L}^1 \setminus \mathcal{L}^2$ has $(2^m - 1)(2^n - 1)$ states, while the smallest XNFA requires more than $m + n$ states.*

Proof. The proof is similar to the proof for Theorem 4.3, except that it is proven that the minimal DFA requires $(2^m - 1)(2^n - 1)$ on the basis of Theorem 3.21. As for the intersection case, $\mathcal{L} = \mathcal{L}^1 \setminus \mathcal{L}^2$ is not an equidistributed language, since occurrences of the pairs $(0, 1)$, $(0, 0)$ and $(1, 1)$ result in a 0 in s^{-}, while only occurrences of the pair $(1, 0)$ result in a 1. Hence, no XNFA with $m+n$ states, which leads to a cycle of length $(2^m - 1)(2^n - 1)$ with good equidistribution properties, will accept $\mathcal{L}$. □

5 CONCLUSION

We have given an upper bound of $(2^m - 1)(2^n - 1)$ for the descriptional complexity of the operations of intersection, union and relative complement for two unary XNFA, with m and n states, respectively. We have also shown that $m + n$ states are insufficient for an XNFA to represent the intersection, union or relative complement of some pairs of languages that can be represented minimally by an m-state XNFA and an n-state XNFA, respectively.

ACKNOWLEDGEMENT

This work is based on the research supported partly by the National Research Foundation of South Africa (grant number 111732).

REFERENCES

[1] Van Zijl, L.: Random number generation with ⊕-NFAs. In Watson, B.W., Wood, D., eds.: Implementation and Application of Automata: 6th International Conference, CIAA 2001 Pretoria, South Africa, July 23–25, 2001 Revised Papers. Volume 2494 of LNCS. Springer, Berlin, Heidelberg (2002) 263–273
[2] Chrobak, M.: Finite automata and unary languages. Theoretical Computer Science **47** (1986) 149–158
[3] Yu, S., Zhuang, Q., Salomaa, K.: The state complexities of some basic operations on regular languages. Theoretical Computer Science **125**(2) (1994) 315–328
[4] Holzer, M., Kutrib, M.: Nondeterministic descriptional complexity of regular languages. International Journal of Foundations of Computer Science **14**(6) (2003) 1087–1102
[5] Birget, J.C.: Partial orders on words, minimal elements of regular languages, and state complexity. Theoretical Computer Science **119**(2) (1993) 267–291
[6] Van der Merwe, A., Van Zijl, L., Geldenhuys, J.: Ambiguity and structural ambiguity of symmetric difference NFAs. Theoretical Computer Science **537** (2014) 97–104
[7] Wang, D.K., Compagner, A.: On the use of reducible polynomials as random number generators. Mathematics of Computation **60** (1993) 363–374
[8] L'Ecuyer, P.: Tables of maximally equidistributed combined LFSR generators. Mathematics of Computation **68**(225) (1999) 261–269
[9] Hopcroft, J.E., Ullman, J.D.: Introduction to Automata Theory, Languages, and Computation. 1st edn. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA (1990)
[10] Van Zijl, L.: Nondeterminism and succinctly representable regular languages. In: Proceedings of the 2002 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists. SAICSIT '02. ACM, Republic of South Africa (2002) 212–223
[11] Marais, L., Van Zijl, L.: Unary self-verifying symmetric difference automata. In Câmpeanu, C., Manea, F., Shallit, J., eds.: Proceedings of the 18th International Conference on the Descriptional Complexity of Formal Systems, DCFS 2016, Bucharest, Romania, July 5-8, 2016. Volume 9777 of LNCS. Springer, Cham (2016) 180–191
[12] Stone, H.S.: Discrete Mathematical Structures and their Applications. Science Research Associates Chicago (1973)
[13] Dornhoff, L.L., Hohn, F.E.: Applied Modern Algebra. Macmillan Publishing Co., Inc. (1978)
[14] Lothaire, M.: Algebraic Combinatorics on Words. Volume 90 of Encyclopedia of mathematics and its applications. Cambridge University Press (2002)
[15] Golomb, S.W.: Shift Register Sequences. Aegean Park Press (1981)