Descriptional Complexity of Non-Unary Self-Verifying Symmetric Difference Automata

Laurette Marais[1,2] and Lynette van Zijl[1]

[1] Department of Computer Science, Stellenbosch University, South Africa
[2] Meraka Institute, CSIR, South Africa

Abstract. We show that, for every $n \geq 2$, there is a regular language $\mathcal{L}_n$ accepted by a non-unary self-verifying symmetric difference nondeterministic automaton with n states, such that its equivalent minimal deterministic finite automaton has 2^{n-1} states. Also, given any SV-XNFA with n states, it is possible, up to isomorphism, to find at most another $|GL(n, \mathbb{Z}_2)| - 1$ equivalent SV-XNFA.

1 Introduction

In [1], we extended the notion of self-verification (SV) to symmetric difference nondeterministic finite automata (XNFA). In contrast to XNFA that have a tight bound of $O(2^n)$ on state complexity for any alphabet [2], we established an upper bound of $2^n - 1$ for SV-XNFA in the case of a unary alphabet, and showed that the bound is not tight. A lower bound of $2^{n-1} - 1$ was also established for the unary case. We showed this to be a tight bound in [3], where we also introduced a more general notion of acceptance for SV-XNFA. In this paper, we consider SV-XNFA with non-unary alphabets. For SV-NFA, Jirásková and Pighizzini [4] showed a tight upper bound $g(n)$, where $g(n)$ grows like $3^{\frac{n}{3}}$. We give an upper bound of $2^n - 1$ and a lower bound of 2^{n-1} for SV-XNFA.

2 Preliminaries

An NFA N is a five-tuple $N = (Q, \Sigma, \delta, Q_0, F)$, where Q is a finite set of states, Σ is a finite alphabet, $\delta : Q \times \Sigma \to 2^Q$ is a transition function (where 2^Q indicates the power set of Q), $Q_0 \subseteq Q$ is a set of initial states, and $F \subseteq Q$ is the set of final, or acceptance, states. The transition function δ can be extended to strings in the Kleene closure Σ^* of the alphabet. Let $w = \sigma_0\sigma_1 \ldots \sigma_k$, then

$$\delta'(q, w) = \delta'(q, \sigma_0\sigma_1 \cdots \sigma_k) = \delta(\delta(\cdots \delta(q, \sigma_0), \sigma_1), \ldots, \sigma_k) \ .$$

For convenience, we write $\delta(q, w)$ to mean $\delta'(q, w)$.

An NFA N is said to accept a string $w \in \Sigma^*$ if $q_0 \in Q_0$ and $\delta(q_0, w) \in F$, and the set of all strings (also called words) accepted by N is the language $\mathcal{L}(N)$ accepted by N. Any NFA has an equivalent DFA which accepts the same language. The DFA $N_D = (Q_D, \delta_D, Q_{0D}, F_D)$ that is equivalent to a given NFA is found by performing the subset construction [5]. In essence, the subset construction keeps track of all the states that the NFA may be in at the same time, and forms the states of the equivalent DFA by a grouping of the states of the DFA. In short,

$$\delta_D(A, \sigma) = \bigcup_{q \in A} \delta(q, \sigma)$$

for any $A \subseteq Q$ and $\sigma \in \Sigma$. Any A is a final state in the DFA if $A \cap F \neq \emptyset$.

2.1 Symmetric difference automata (XNFA)

A symmetric difference NFA (XNFA) is defined similarly to an NFA, except that the DFA equivalent to the XNFA is found by taking the symmetric difference (in the set theoretic sense) in the subset construction. That is, for any two sets A and B, the symmetric difference is given by $\oplus(A, B) = (A \cup B) \setminus (A \cap B)$. The subset construction is then applied as

$$\delta_D(A, \sigma) = \bigoplus_{q \in A} \delta(q, \sigma)$$

for any $A \subseteq Q$ and $\sigma \in \Sigma$.

For clarity, the DFA equivalent to an XNFA N is termed an XDFA and denoted with $N_D = (Q_D, \delta_D, Q_{0D}, F_D)$. Note that $\delta_D : 2^Q \times \Sigma \to 2^Q$. It is customary to require that an XDFA final state consist of an odd number of final XNFA states, as an analogy to the symmetric difference set operation [6] – this is known as parity acceptance. XNFA accept the class of regular languages [6].

Given parity acceptance, XNFA have been shown to be equivalent to weighted automata over the finite field of two elements, or GF(2) [6,7]. For an XNFA $N = (Q, \Sigma, \delta, Q_0, F)$, the transitions for each alphabet symbol σ can be represented as a matrix over GF(2). Each row represents a mapping from a state $q \in Q$ to a set of states $P \in 2^Q$. P is written as a vector with a one in position i if $q_i \in P$, and a zero in position i if $q_i \notin P$. Hence, the transition table is represented as a matrix M_σ of zeroes and ones (see Example 1). This is known as the characteristic or transition matrix for σ of the XNFA. In the rest of this paper, we consider only SV-XNFA with non-singular matrices, whose cycle structures do not include transient heads, i.e. states that are only reached once before a cycle is reached.

Initial and final states are similarly represented by vectors, and appropriate vector and matrix multiplications over GF(2) represent the behaviour of the XNFA[3]. For instance, in the unary case we would have a single matrix M_a that describes the transitions on a for some XNFA with n states. We encode the initial states Q_0 as vector of length n over GF(2), namely $v(Q_0) = [q_{0_0}\ q_{0_1}\ \cdots\ q_{0_{n-1}}]$, where $q_{0_i} = 1$ if $q_i \in Q_0$ and 0 otherwise. Similarly, we encode the final states as a length n vector $v(F) = [q_{F_0}\ q_{F_1}\ \cdots\ q_{F_{n-1}}]$. Then $v(Q_0)M_a$ is a vector that encodes the states reached after reading the symbol a exactly once, and

[3] In GF(2), $1 + 1 = 0$.

$v(Q_0)M_a^k$ encodes the states reached after reading the symbol a k times. The weight of a word w_k of length k is given by

$$\Delta(w_k) = v(Q_0)M_a^k v(F)^T \ .$$

We can say that M_a represents the word a, and $M_{a^k} = M_a^k$ represents the word a^k. In the binary case, we would have two matrices, M_a for transitions on a and M_b for transitions on b. Reading an a corresponds to multiplying by M_a, while reading a b corresponds to multiplying by M_b. Let M_w be the result of the appropriate multiplications of M_a and M_b representing some $w \in \{a, b\}^*$, then the weight of w is given by $\Delta(w) = v(Q_0)M_w v(F)^T$.

We now show that, in the unary case, a so-called change of basis is possible, where for some $n \times n$ transition matrix M_a of an XNFA and any non-singular $n \times n$ matrix A, $M'_a = A^{-1}M_a A$ is the transition matrix of an equivalent XNFA with $v(Q'_0) = v(Q_0)A$ and $v(F')^T = A^{-1}v(F)^T$. For any word w_k of length k, we have the following:

$$\begin{aligned}
\Delta'(w_k) &= v(Q'_0)M_a'^k v(F')^T \\
&= v(Q_0)A(A^{-1}M_a A)^k A^{-1}v(F)^T \\
&= v(Q_0)M_a^k v(F)^T \\
&= \Delta(w_k) \ .
\end{aligned}$$

This also applies to the binary case. For some XNFA N, let $M_w = \prod_{i=1}^{k} M_{\sigma_i}$ represent a word $w = \sigma_1\sigma_2...\sigma_k$, where $M_{\sigma_i} = M_a$ if $\sigma_i = a$, and similarly for b. Now, let N' be an XNFA whose transition matrices are $M'_a = A^{-1}M_a A$ and $M'_b = A^{-1}M_b A$ for some non-singular A. Then w is represented by

$$\begin{aligned}
M'_w &= \prod_{i=1}^{k} M'_{\sigma_i} \\
&= M'_{\sigma_1}M'_{\sigma_2}\cdots M'_{\sigma_k} \\
&= (A^{-1}M_{\sigma_1}A)(A^{-1}M_{\sigma_2}A)\cdots(A^{-1}M_{\sigma_k}A) \\
&= A^{-1}M_{\sigma_1}M_{\sigma_2}\cdots M_{\sigma_k}A \\
&= A^{-1}M_w A \ .
\end{aligned}$$

And so the weight of any word w_k on N' is

$$\begin{aligned}
\Delta'(w) &= v(Q'_0)M'_w v(F')^T \\
&= v(Q_0)A(A^{-1}M_w A)A^{-1}v(F)^T \\
&= v(Q_0)M_w v(F)^T \\
&= \Delta(w) \ .
\end{aligned}$$

Note that the above discussion does not rely on the fact that there are only two alphabet symbols, and so applies in general to the r-ary case as well.

2.2 Self-verifying automata (SV-NFA)

Self-verifying NFA (SV-NFA) [4,8,9] are automata with two kinds of final states, namely accept states and reject states, as well as neutral non-final states. It is required that for any word, one or more of the paths for that word reach a single kind of final state, i.e. either accept states or reject states are reached, but not both. Consequently, self-verifying automata reject words explicitly if they reach a reject state, in contrast to NFA, where rejection is the result of a failure to reach an accept state.

Definition 1. *A self-verifying nondeterministic finite automaton (SV-NFA) is a 6-tuple* $N = (Q, \Sigma, \delta, Q_0, F^a, F^r)$*, where* Q, Σ, δ *and* Q_0 *are defined as for standard NFA.* $F^a \subseteq Q$ *and* $F^r \subseteq Q$ *are the sets of accept and reject states, respectively. The remaining states, that is, the states belonging to* $Q \setminus (F^a \cup F^r)$*, are called neutral states. For each input string* w *in* Σ^**, it is required that there exists at least one path ending in either an accept or a reject state; that is,* $\delta(q_0, w) \cap (F^a \cup F^r) \neq \emptyset$ *for any* $q_0 \in Q_0$*, and there are no strings* w *such that both* $\delta(q_0, w) \cap F^a$ *and* $\delta(q_1, w) \cap F^r$ *are nonempty, for any* $q_0, q_1 \in Q_0$*.*

Since any SV-NFA either accepts or rejects any string $w \in \Sigma^*$ explicitly, its equivalent DFA must do so too. The path for each w in a DFA is unique, so each state in the DFA is an accept or reject state. Hence, for any DFA state d, there is some SV-NFA state $q_i \in d$ such that $q_i \in F^a$ (and hence $d \in F_D^a$) or $q_i \in F^r$ (and hence $d \in F_D^r$). Since each state in the DFA is a subset of states of the SV-NFA, accept and reject states cannot occur together in a DFA state. That is, if d is a DFA state, then for any $p, q \in d$, if $p \in F^a$ then $q \notin F^r$ and vice versa. We refer to the equivalent DFA of some SV-XNFA as its equivalent SV-XDFA to indicate that every state must accept or reject and that parity acceptance holds given the subset construction. Any SV-XDFA is equivalent to an XDFA, so SV-XNFA accept the class of regular languages.

2.3 Self-verifying symmetric difference automata (SV-XNFA)

In [1], self-verifying symmetric difference automata (SV-XNFA) were defined as a combination of the notions of symmetric difference automata and self-verifying automata, but only the unary case was examined. We now restate the definition of SV-XNFA in order to present results on larger alphabets in Section 4. Note, however, that the definition is slightly amended: in [1], the implicit assumption was made that no SV-XNFA state could be both an accept state and a reject state. This assumption is explored in detail for the unary case in [3], but for our current purposes it suffices to say that such a requirement removes the equivalence between XNFA and weighted automata over GF(2), which is essential for certain operations on XNFA, such as minimisation [7]. This implies that parity acceptance applies to SV-XNFA, where the condition for self-verification (SV-condition) is that for any word, an odd number of paths end in either accept states or reject states, but not both. In terms of the equivalent XDFA, this is equivalent to requiring that any XDFA state contain either an odd number of accept states or an odd number of rejects states, but not both. If an XNFA state is both an accept state and a reject state, it contributes to both counts.

Definition 2. *A self-verifying symmetric difference finite automaton (SV-XNFA) is a 6-tuple* $N = (Q, \Sigma, \delta, Q_0, F^a, F^r)$*, where* Q, Σ, δ *and* Q_0 *are defined as for XNFA, and* F^a *and* F^r *are defined as for SV-NFA, except that* $F^a \cap F^r$ *need not be empty. That is, each state in the SV-XDFA equivalent to* N *must contain an odd number of states from either* F^a *or* F^r*, but not both, and some SV-XNFA states may belong to both* F^a *and* F^r*.*

The SV-condition for XNFA implies that if a state in the SV-XDFA of an SV-XNFA N contains an odd number of states from F^a, it may also contain an even number of states from F^r, and hence belong to F^a_D, and vice versa. An SV-XDFA state may contain any number of neutral states from N.

The choice of F^a and F^r for a given SV-XNFA N is called an *SV-assignment* of N. An SV-assignment where either F^a or F^r is empty, is called a *trivial SV-assignment*. Otherwise, if both F^a and F^r are nonempty, the SV-assignment is non-trivial.

3 XNFA and linear feedback shift registers

In [10] it is shown that unary XNFA are equivalent to linear feedback shift registers (LFSRs). Specifically, a matrix M with characteristic polynomial $c(X)$ is associated with a certain cycle structure of sets of XNFA states (or of XDFA states), and the choice of Q_0 determines which cycle represents the behaviour of a specific unary XNFA. The cycle structure is induced by $c(X)$, so any matrix that has $c(X)$ as its characteristic polynomial has the same cycle structure, although the states ocurring in the cycles differ according to each specific matrix.

For the r-ary case, the transition matrix for each symbol is associated with its own cycle structure, and the choice of Q_0 determines which cycle is realised in the r-ary XNFA for each symbol. There are $2^n - 1$ possible choices for Q_0 (we exclude the empty set). Evidently, the cycles associated with each symbol might overlap, and so the structure of the r-ary XNFA would not be cyclic itself, although the transitions for each symbol would exhibit cyclic behaviour. Specifically, for an r-ary XNFA N and some symbol $\sigma \in \Sigma$, we refer to the cycle structure of N on σ as the cycle structure resulting from considering only transitions on σ. Our main results will be derived from examining the cycle structure induced by each symbol of the alphabet of the automaton, as well as the ways in which the cycles overlap.

For any $c(X) = X^n + c_{n-1}X^{n-1} + \cdots + c_1X + c_0$ there is a normal form matrix M of the form given below, such that $c(X) = \det(XI - M)$, where I is the identity matrix. We say that M is in canonical form.

$$M = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & & \cdots & 1 \\ c_0 & c_1 & \cdots & c_{n-2} & c_{n-1} \end{bmatrix}$$

In the next lemma, it will be convenient to represent XDFA states $d_s \subseteq Q$ as $s = \langle s_{n-1}, s_{n-2}, ..., s_1, s_0 \rangle$, where $s_i = 1$ if $q_i \in d_s$ and 0 otherwise. The lemma is adapted from [11] on the basis of the equivalence between unary XNFA and LFSRs.

Lemma 1. *Let M_σ be a transition matrix representing transitions on σ for some XNFA N, with characteristic polynomial $c_\sigma(X)$, and let M_σ be in canonical form. Let f be a bijection of the states of the equivalent XDFA N_D onto polynomials of degree $n-1$, such that f maps the state $s = \langle s_{n-1}, s_{n-2}, ..., s_1, s_0 \rangle$ into the polynomial $f(s) = s_{n-1}X^{n-1} + s_{n-2}X^{n-2} + \cdots + s_1 X + s_0$. Then f maps the state $M_\sigma \cdot s$ into the polynomial $Xf(s)$ mod $c_\sigma(X)$.*

Lemma 1 provides a mapping between polynomials over $GF(2)$ and the states of XDFA. The XDFA state arrived at after a transition from state s on σ corresponds to the polynomial which results from multipying $f(s)$ by X in the polynomial algebra of $GF(2)[X]$ modulo $c(X)$.

Example 1. Let N be a binary XNFA (shown in Figure 1), where M_a is the normal form matrix of $c_a(X) = X^4 + X^2 + X + 1$ and M_b is the normal form matrix of $c_b(X) = X^4 + X^3 + X + 1$. M_a and M_b are given below. The resulting XDFA is shown in Figure 2, while some examples comparing state transitions and polynomial multiplication are shown in Table 1. Note that, for now, the focus is on the cyclic behaviour of the equivalent XDFA, and so we do not refer to any final states.

$$M_a = \begin{bmatrix} 0&1&0&0 \\ 0&0&1&0 \\ 0&0&0&1 \\ 1&1&1&0 \end{bmatrix} \qquad M_b = \begin{bmatrix} 0&1&0&0 \\ 0&0&1&0 \\ 0&0&0&1 \\ 1&1&0&1 \end{bmatrix}$$

4 Non-unary SV-XNFA

The upper bound on state complexity is simply $2^n - 1$, since this is the number of non-empty subsets for any set of n XNFA states. We now work towards establishing a lower bound on state complexity. First, we restate the following lemma from [1] for the unary case.

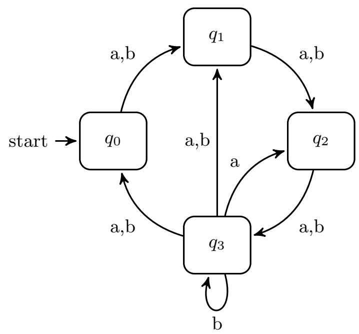

Fig. 1. Example 1: N

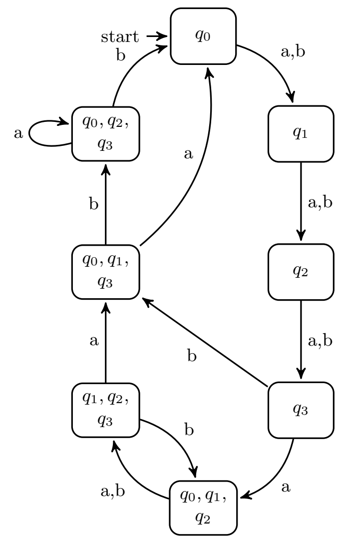

Fig. 2. Example 1: N_D

Table 1. Transitions on δ correspond to multiplication by X

$\delta_D(s, \sigma)$	$Xf(s) \bmod c_\sigma(X)$	
$\delta_D(\{q_0\}, a) = \{q_1\}$	$X(1)$	$= X$
$\delta_D(\{q_3\}, a) = \{q_0, q_1, q_2\}$	$X(X^3)$	$= X^4 \bmod c_a(X)$
		$= X^2 + X + 1$
$\delta_D(\{q_0, q_2, q_3\}, a) = \{q_0, q_2, q_3\}$	$X(X^3 + X^2 + 1)$	$= X^4 + X^3 + X \bmod c_a(X)$
		$= X^3 + X^2 + 1$
$\delta_D(\{q_1\}, b) = \{q_2\}$	$X(X)$	$= X^2$
$\delta_D(\{q_0, q_1, q_3\}, b) = \{q_0, q_2, q_3\}$	$X(X^3 + X + 1)$	$= X^4 + X^2 + X \bmod c_b(X)$
		$= X^3 + X^2 + 1$
$\delta_D(\{q_1, q_2, q_3\}, b) = \{q_0, q_1, q_2\}$	$X(X^3 + X^2 + X)$	$= X^4 + X^3 + X^2 \bmod c_b(X)$
		$= X^2 + X + 1$

Lemma 2. *Let $c(X) = (X + 1)\phi(X)$ be a polynomial of degree n with non-singular normal form matrix M, and let N be a unary XNFA with transition matrix M and $Q_0 = \{q_0\}$. Then the equivalent XDFA N_D has the following properties:*

1. $|Q_D| > n$
2. $|d|$ *is odd for* $d \in Q_D$
3. $[q_0], [q_1], ..., [q_{n-1}] \in Q_D$

$|d|$ is the number of XNFA states in the XDFA state $d \subseteq Q$, or the number of one's in the representation of d as $\langle s_{n-1}, s_{n-2}, ..., s_1, s_0\rangle$ where $s_i = 1$ if $q_i \in d$ and 0 otherwise.

Theorem 1. *Let M_{σ_1}, M_{σ_2}, ..., M_{σ_r} be the normal form matrices of r polynomials $c_{\sigma_1}(X) = (X + 1)\phi_{\sigma_1}(X)$, $c_{\sigma_2}(X) = (X + 1)\phi_{\sigma_2}(X)$, ..., $c_{\sigma_r}(X) = (X + 1)\phi_{\sigma_r}(X)$, respectively, and let M_{σ_1}, M_{σ_2}, ..., M_{σ_r} be the transition matrices of some r-ary XNFA N with $\Sigma = \{\sigma_1, \sigma_2, ..., \sigma_r\}$ and $Q_0 = \{q_0\}$. Then the number of states in the equivalent XDFA N_D does not exceed 2^{n-1}. Furthermore, any choice of F^a and F^r such that $F^a \cup F^r = Q$ and $F^a \cap F^r = \emptyset$ is an SV-assignment.*

Proof. By Lemma 2, $|d|$ is odd for $d \in Q_D$ in the unary case. That is, for any symbol with a transition matrix whose polynomial has $X + 1$ as a factor, a transition from an odd-sized XDFA state is to another odd-sized XDFA state. Since $Q_0 = \{q_0\}$ and $|\{q_0\}|$ is odd, and $c_{\sigma_1}(X)$, $c_{\sigma_2}(X)$,...$c_{\sigma_r}(X)$, have $X + 1$ as a factor, only odd-sized states are reachable on any transition. The number of XDFA states d such that $|d|$ is odd is $2^n/2 = 2^{n-1}$, and so N_D can have at most 2^{n-1} states. Since every XDFA state contains an odd number of XNFA states, any choice of F^a and F^r such that $F^a \cup F^r = Q$ and $F^a \cap F^r = \emptyset$ is an SV-assignment. □

The following lemma provides further information on the cycle structure induced by polynomials with $X + 1$ as a factor.

Lemma 3. *Let $c_\sigma(X) = (X + 1)\phi(X)$. Then, in the normal form matrix M_σ of $c_\sigma(X)$, which is the transition matrix on some symbol σ for an XNFA, the state mapped to $\phi(X)$ as described in Lemma 1, i.e. d_ϕ, is contained in a cycle of length one, when considering only transitions on σ.*

Proof. Consider the following:

$$\begin{aligned}(X + 1)\phi(X) &= c_\sigma(X)\\ X\phi(X) + \phi(X) &= c_\sigma(X)\\ X\phi(X) &= \phi(X) + c_\sigma(X)\end{aligned}$$

Therefore, $X\phi(X) = \phi(X)$ in the representation of $GF(2^n)$ as polynomials over GF(2) modulo $c_\sigma(X)$. By Lemma 1, this corresponds to $\delta_D(d_\phi, \sigma) = d_\phi$. □

We now present a witness language for any n to show that 2^{n-1} is a lower bound on the state complexity of SV-XNFA with non-unary alphabets. First, we restate the following theorem from [1].

Theorem 2. *For any $n \geq 2$, there is an SV-XNFA N whose equivalent N_D has $2^{n-1} - 1$ states.*

Lemma 4. *Let $\phi(X) = X^{n-1} + \phi_{n-2}X^{n-2} + \cdots + \phi_1 X + \phi_0$ be any primitive polynomial of degree $n-1$. Let N be a binary XNFA, and let the transition matrix on a be the normal form matrix of $c_a(X) = (X+1)\phi(X)$ and the transition matrix on b be the normal form matrix of $c_b(X) = X^n + \phi(X)$. Then the equivalent XDFA of the XNFA with $Q_0 = \{q_0\}$ contains exactly 2^{n-1} odd-sized states.*

Proof. We write $c_a(X)$ and $c_b(X)$ in the following way:

$$c_a(X) = X^n + c_{n-1}X^{n-1} + \cdots + c_1 X + c_0$$
$$c_b(X) = X^n + \phi_{n-1}X^{n-1} + \phi_{n-2}X^{n-2} + \cdots + \phi_1 X + \phi_0$$

Since $\phi(X)$ is primitive, it has no roots in GF(2), including 1, so it must have an odd number of non-zero terms. Therefore, by Lemma 1, $|d_\phi|$ is odd. Furthermore, $c_b(X)$ has an even number of non-zero terms, and so has 1 as a root. Consequently, $c_b(X)$ has $X+1$ as a factor.

The transition matrices M_a and M_b are given below. Note that they are both non-singular. Let $Q_0 = \{q_0\}$. Then by Theorem 2, the cycle structure on a is equivalent to an XDFA cycle with $2^{n-1} - 1$ states, all of which, by Lemma 2, have odd size. Also, by Lemma 3, d_ϕ is not contained in this cycle. This means that on a, every odd-sized state in the XDFA is reached except for d_ϕ. Now,

$$M_a = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & & \cdots & 1 \\ c_0 & c_1 & \cdots & c_{n-2} & c_{n-1} \end{bmatrix} \qquad M_b = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & & \cdots & 1 \\ \phi_0 & \phi_1 & \cdots & \phi_{n-2} & \phi_{n-1} \end{bmatrix}$$

from M_b it follows directly that $\delta_D(\{q_{n-1}\}, b) = d_\phi$. Furthermore, since $X+1$ is a factor of c_b, every transition from an odd-sized state on b is to an odd-sized state. Consequently, the binary XNFA N is equivalent to an XDFA that reaches all 2^{n-1} odd-sized states and none other. $\square$

Theorem 3. *For any $n \geq 2$, there is a language $\mathcal{L}_n$ so that some n-state binary SV-XNFA accepts $\mathcal{L}_n$ and the minimal SV-XDFA that accepts $\mathcal{L}_n$ has 2^{n-1} states.*

Proof. Let $c_a(X) = (X+1)\phi(X)$ and $c_b = X^n + \phi(X)$, where $\phi(X)$ is a primitive polynomial and let $c_a(X)$ and $c_b(X)$ have degree n. We construct an SV-XNFA N with n states whose equivalent XDFA N_D has 2^{n-1} states as in Lemma 4, and let $F^a = \{q_0\}$ and $F^r = Q \setminus F^a$. Recall that for N, we have $\delta : Q \times \Sigma \to 2^Q$, and for N_D, we have $\delta_D : 2^Q \times \Sigma \to 2^Q$.

Let $\mathcal{L}^1_n = a^{(2^{n-1}-1)i+j}$ for $i \geq 0$ and $j \in J$, where J is some set of integers, represent a subset of the language accepted by N that consists only of strings containing a. Now, from the transition matrix of N it follows that $0, n \in J$, while $1, 2, ..., n-1 \notin J$, since $q_0 \in \delta(q_0, a^n)$, but $q_0 \notin \delta(q_0, a^m)$ for $m < n$.

If there is an N'_D with fewer than $2^{n-1} - 1$ states that accepts $\mathcal{L}^1_n$, then there must be some $d_j \in Q_D$ such that $\{q_0\} \subset d_j$, $q_0 \in \delta_D(d_j, a^n)$ and there is no $m < n$ so that $q_0 \in \delta_D(d_j, a^m)$. That is, if N'_D exists, then on N_D, $\delta_D(\{q_0\}, a) = \delta_D(d_j, a)$, and $\delta_D(\{q_1\}, a) = \delta_D(d_{j+1}, a)$ etc.

Let d_k be any state in N_D such that $d_k \neq \{q_0\}$. Let $max(d_k)$ be the largest subscript of any SV-XNFA state in d_k. Then $max(d_k) > 0$. Let $m = n - max(d_k)$, so $m < n$. Then, from the transition matrix of N, it follows that $q_0 \in \delta_D(d_k, a^m)$. That is, for any d_k there is an $m < n$ so that $q_0 \in \delta_D(d_k, a^m)$. Therefore, there is no N'_D with fewer than $2^{n-1} - 1$ states that accepts $\mathcal{L}^1_n$.

Now, let $\mathcal{L}^2_n = b^n a^*$, which is also a subset of the language accepted by N. In order to accept this language, after reading b^n, a state must have been reached whereafter every transition on a must result in an accept state, i.e. an XDFA state containing q_0. But there is only one such state, and that is d_ϕ, since $\delta_D(d_\phi, a) = d_\phi$, which is excluded from the cycle needed to accept $\mathcal{L}^1_n$. Therefore, all 2^{n-1} odd-sized states are necessary to accept $\mathcal{L}^1 \cup \mathcal{L}^2$. Let $\mathcal{L}_n$ be the language accepted by N, then since $\mathcal{L}^1_n \cup \mathcal{L}^2_n \subset \mathcal{L}_n$, at least 2^{n-1} states are necessary to accept $\mathcal{L}_n$. □

We illustrate Theorem 3 for $n = 4$.

Example 2. Let $\phi(X) = X^3 + X + 1$, which is a primitive polynomial. Now, let N be an XNFA with transition matrices M_a and M_b. M_a is the normal form matrix of $c_a(X) = (X + 1)\phi(X) = X^4 + X^3 + X^2 + 1$ and M_b the normal form matrix of $c_b(X) = X^4 + \phi(X) = X^4 + X^3 + X + 1$. Let $Q_0 = \{q_0\}$ and let $F^a = \{q_0\}$ and $F^r = \{q_1, q_2, q_3\}$. M_a and M_b are shown below, while N and its equivalent XDFA N_D are shown in Figures 3 and 4. We have $\mathcal{L}^1 = a^{7i+j}$ for $i \geq 0$ and $j \in \{0, 4, 5\}$ and $\mathcal{L}^2 = bbbba^*$.

$$M_a = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix} \qquad M_b = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix}$$

□

The following is a simple corollary of Theorem 3.

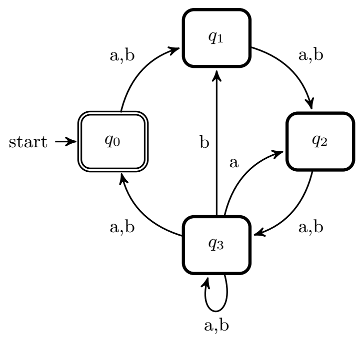

Fig. 3. Example 2: N

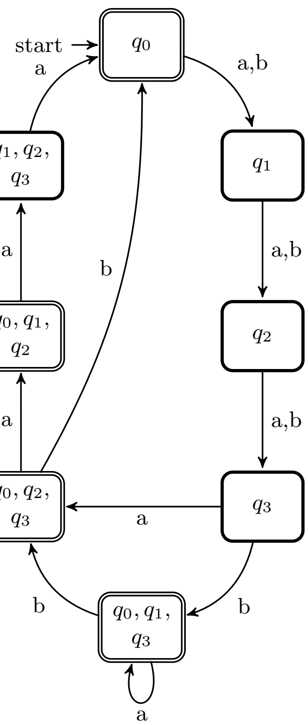

Fig. 4. Example 2: N_D

Corollary 1. *For any $m, n \geq 2$, there is a language $\mathcal{L}'_n$ so that some n-state m-ary SV-XNFA accepts $\mathcal{L}'_n$ and the minimal SV-XDFA that accepts $\mathcal{L}'_n$ has 2^{n-1} states.*

We now show that any given SV-XNFA can be used to obtain another one via a so-called change of basis.

Theorem 4. *Given any SV-XNFA $N = (Q, \Sigma, \delta, Q_0, F^a, F^r)$ with n states and transition matrices M_{σ_1}, M_{σ_2}, ..., M_{σ_r}, and any non-singular $n \times n$ matrix A, we encode Q_0 as a vector $v(Q_0)$ of length n over GF(2) and F^a and F^r as vectors $v(F^a)$ and $v(F^r)$ respectively. Then there is an SV-XNFA $N' = (Q, \Sigma, \delta', Q'_0, F'^a, F'^r)$ where $M'_{\sigma_i} = A^{-1} M_{\sigma_i} A$ for $0 \leq i \leq r$, $v(Q'_0) = v(Q_0)A$, $v(F'^a)^T = A^{-1} v(F^a)^T$ and $v(F'^r)^T = A^{-1} v(F^r)^T$, and N' accepts the same language as N.*

Proof. In the discussion in Section 2.1 we showed that for XNFA, the change of basis described on an XNFA N that results in N', $\Delta'(w) = \Delta(w)$. We extend this to SV-XNFA by defining two new functions. Recall that M_w represents the sequence of matrix multiplications for some w of length k, and that $M'_w = A^{-1} M_w A$. Then, let

$$\begin{aligned} accept(w) &= v(Q_0) M_w v(F^a)^T \\ reject(w) &= v(Q_0) M_w v(F^r)^T \quad . \end{aligned}$$

The SV-condition is that $accept(w) \neq reject(w)$ for any $w \in \Sigma^*$. Similar to $\Delta(w)$, we have

$$\begin{aligned} accept'(w) &= v(Q'_0) M'_w v(F'^a)^T \\ &= v(Q_0) A (A^{-1} M_w A) A^{-1} v(F^a) \\ &= v(Q_0) M_w v(F^a) \\ &= accept(w) \end{aligned}$$

and

$$\begin{aligned} reject'(w) &= v(Q'_0) M'_w v(F'^r)^T \\ &= v(Q_0) A (A^{-1} M_w A) A^{-1} v(F^r) \\ &= v(Q_0) M_w v(F^r) \\ &= reject(w) \end{aligned}$$

Clearly, the SV-condition is met by $accept'$ and $reject'$, and so N' is an SV-XNFA that accepts the same language as N. □

The number of non-singular $n \times n$ matrices over GF(2) (including the identity matrix) is $|GL(n, \mathbb{Z}_2)| = \prod_{k=0}^{n-1} (2^n - 2^k)$, and so, up to isomorphism, for any SV-XNFA at most another $|GL(n, \mathbb{Z}_2)| - 1$ equivalent SV-XNFA can be found.

Example 3. Let N be an SV-XNFA with alphabet $\Sigma = \{a, b, c\}$, and the following transition matrices: M_a is the normal form matrix of $c(X) = X^4+X^3+X^2+1$, M_b is the normal form matrix of X^4+X^3+X+1, and M_c is the normal form matrix of $c(X) = X^4+X^2+X+1$. Let $Q_0 = \{q_0\}$, $F^a = \{q_0, q_2\}$ and $F^r = \{q_1, q_3\}$. Figure 5 shows N and the equivalent XDFA N_D is given in Figure 7, where a double edge indicates an accept state and a thick edge indicates a reject state. Consider the following matrix A:

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

We use A to make a change of basis from N to N'. Let N' be an XNFA with $\Sigma = \{a, b, c\}$, where $M'_a = A^{-1}M_aA$, $M'_b = A^{-1}M_bA$ and $M'_c = A^{-1}M_cA$. Furthermore, let $v(Q'_0) = v(Q_0)A$, i.e. $Q'_0 = \{q_1, q_2, q_3\}$. Finally, let $v(F'^a)^T = A^{-1}v(F^a)^T$ and $v(F'^r)^T = A^{-1}v(F^r)^T$, i.e. $F'^a = \{q_0, q_2\}$ and $F'^r = \{q_2, q_3\}$. Figure 6, shows N', with a double edge indicating an accept state, a thick edge indicating a reject state and a thick double edge indicating a state that is both an accept state and a reject state. Figure 8 gives the equivalent XDFA N'_D. It is worth noting that, although N' has a different structure than N, N'_D has the same structure as N_D, and accepts the same language. Also, note that in N'_D, the state $\{q_0, q_1, q_2\}$ is a reject state, because it contains an even number of accept states, namely q_0 and q_2, but an odd number of reject states, namely q_2. □

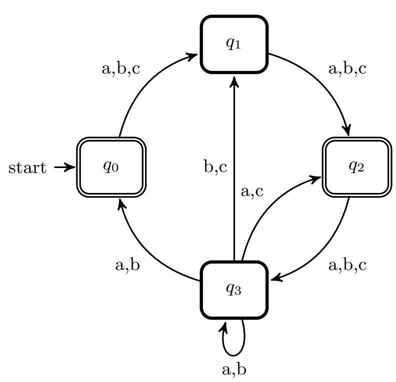

Fig. 5. Example 3: N

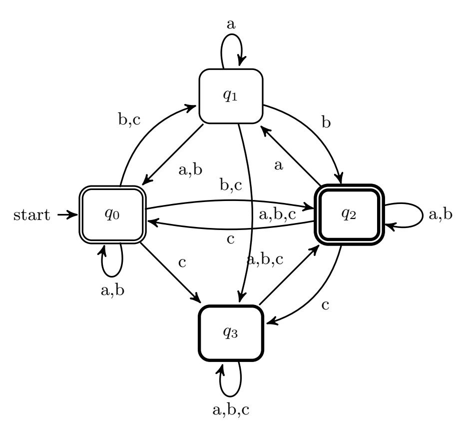

Fig. 6. Example 3: N'

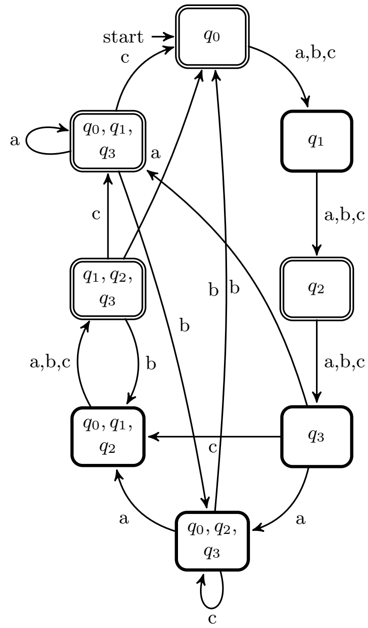

Fig. 7. Example 3: N_D

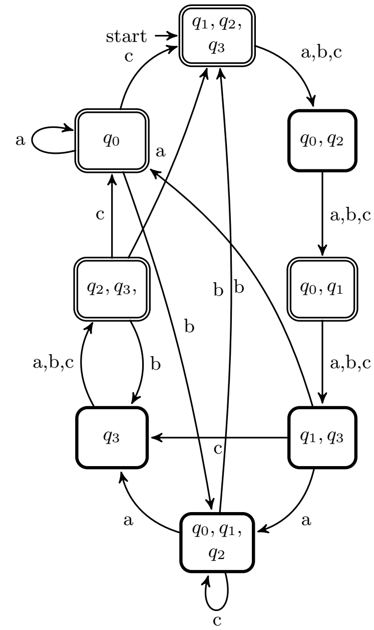

Fig. 8. Example 3: N'_D

5 Conclusion

We have given an upper bound of $2^n - 1$ on the state complexity of SV-XNFA with alphabets larger than one, and a lower bound of 2^{n-1}. We have also shown that, given any SV-XNFA with n states, it is possible, up to isomorphism, to find at most another $|GL(n, \mathbb{Z}_2)| - 1$ equivalent SV-XNFA via a change of basis.

References

1. Marais, L., van Zijl, L.: Unary Self-verifying Symmetric Difference Automata. In: Proceedings of Descriptional Complexity of Formal Systems: 18th IFIP WG 1.2 International Conference, DCFS 2016, Bucharest, Romania, July 5-8, 2016. Springer International Publishing (2016) 180–191
2. van Zijl, L., Geldenhuys, J.: Descriptional complexity of ambiguity in symmetric difference nfas. Journal of Universal Computer Science **17**(6) (2011) 874 – 890
3. Marais, L., Van Zijl, L.: State complexity of Unary SV-XNFA with Different Acceptance Conditions. Submitted for publication
4. Jirásková, G., Pighizzini, G.: Optimal simulation of self-verifying automata by deterministic automata. Information and Computation **209**(3) (2011) 528 – 535 Special Issue: 3rd International Conference on Language and Automata Theory and Applications (LATA 2009).
5. 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)

6. Vuillemin, J., Gama, N.: Compact Normal Form for Regular Languages as Xor Automata. In: Proceedings of the 14th International Conference on the Implementation and Application of Automata (CIAA 2009), Sydney, Australia, July 14-17, 2009. Springer, Berlin, Heidelberg (2009) 24–33
7. Van der Merwe, B., Tamm, H., Van Zijl, L.: Minimal DFA for Symmetric Difference NFA. In: Proceedings of the 14th International Conference on the Descriptional Complexity of Formal Systems (DCFS 2012), Braga, Portugal, July 23-25, 2012. Springer, Berlin, Heidelberg (2012) 307–318
8. Assent, I., Seibert, S.: An upper bound for transforming self-verifying automata into deterministic ones. RAIRO-Theoretical Informatics and Applications-Informatique Théorique et Applications **41**(3) (2007) 261–265
9. Hromkovič, J., Schnitger, G.: On the Power of Las Vegas II. Two-Way Finite Automata. In: Proceedings of Automata, Languages and Programming: 26th International Colloquium, ICALP'99 Prague, Czech Republic, July 11–15, 1999. Springer, Berlin, Heidelberg (1999) 433–442
10. Van Zijl, L., Harper, J.P., Olivier, F.: The MERLin Environment Applied to $\star$-NFAs. In: Implementation and Application of Automata: 5th International Conference, CIAA 2000 London, Ontario, Canada, July 24–25, 2000 Revised Papers. Springer, Berlin, Heidelberg (2001) 318–326
11. Stone, H.S.: Discrete Mathematical Structures and their Applications. Science Research Associates Chicago (1973)