Ambiguity of Unary Symmetric Difference NFAs

Brink van der Merwe, Lynette van Zijl, and Jaco Geldenhuys *

Department of Computer Science
Stellenbosch University, Private Bag X1, 7602 Matieland, SOUTH AFRICA
abvdm@cs.sun.ac.za,lvzijl@sun.ac.za,jaco@cs.sun.ac.za

Abstract. Okhotin [9] showed an exponential trade-off in the conversion from nondeterministic unary finite automata to unambiguous nondeterministic unary finite automata. In this paper, we consider the trade-off in the case of unary symmetric difference finite automata to finitely ambiguous unary symmetric difference finite automata. Surprisingly, the trade-off is linear in the number of states of the finite automaton. In particular, for every n-state unary nondeterministic symmetric difference finite automaton, there is an equivalent finitely ambiguous n-state unary symmetric difference nondeterministic finite automaton. We also note other relevant ambiguity issues in the unary case, such as the ambiguity of k-deterministic finite automata.

Keywords: nondeterminism, ambiguity

1 Introduction

Symmetric difference nondeterministic finite automata ($\oplus$-NFAs) are well-suited to the investigation of periodic or cyclic behaviour in regular languages. The succinctness of $\oplus$-NFAs has been investigated in some detail [15], but little work has been done on the language-theoretic properties of $\oplus$-NFAs. In this work, we therefore consider the issue of the *ambiguity* of unary $\oplus$-NFAs.

The ambiguity of a nondeterministic finite automaton (NFA) M refers to the maximum number of different accepting paths of M for all the words in the language accepted by M. For example, if M has only one accepting path for any word, then M is unambiguous. Or, M may have no more than c accepting paths for any word (with c a constant), in which case M is finitely ambiguous. Similarly, M may be polynomially or exponentially ambiguous, if the number of accepting paths is at most polynomial or exponential in the number of letters in an accepted word. The ambiguity of NFAs has been extensively investigated (see for example [5, 6, 10]), and recently Okhotin [9] considered the difference between unary NFAs and NFAs with larger alphabets as far as ambiguity is concerned.

In previous work [17] we investigated the ambiguity of $\oplus$-NFAs (as opposed to the ambiguity of traditional NFAs). We showed the existence of families of $\oplus$-NFAs for each ambiguity class, and also considered the descriptional complexity

* This research was supported by NRF grant #69872.

of ambiguous $\oplus$-NFAs. In particular, we showed that for each ambiguity class, there exists an n-state binary $\oplus$-NFA for which the minimal equivalent DFA has $O(2^n)$ states. In this work, we specifically consider the ambiguity of *unary* $\oplus$-NFAs. Here, we are interested in the state trade-off between equivalent finitely ambiguous unary $\oplus$-NFAs and unary $\oplus$-NFAs falling in any other ambiguity class. Okhotin [9] showed, for traditional NFAs, an exponential trade-off between unary NFAs and unary unambiguous NFAs. Surprisingly, quite different results hold for $\oplus$-NFAs, and we shall show that for any unary n-state $\oplus$-NFA, there is an equivalent unary finitely ambiguous $\oplus$-NFA with n states.

The remainder of this article is organised as follows: Sect. 2 gives background and definitions, and specifically establishes the algebraic background required in the rest of the paper. In Sect. 3 we prove the state trade-off between unary $\oplus$-NFAs and unary finitely ambiguous $\oplus$-NFAs. The next section notes some related results, such as the ambiguity of k-deterministic finite automata. We conclude in Sect. 5.

2 Background

$\oplus$-NFAs were defined in [15], and Vuillemin and Gama give an overview of the mathematical basis for $\oplus$-NFAs [14]. We therefore only briefly summarize the necessary definitions and background. We assume that the reader has a basic knowledge of automata theory and formal languages, as for example in [12], and a background in linear algebra, as for example in [11]. Note that symmetric difference is used in the usual set theoretic sense: for any two sets A and B, the symmetric difference of A and B is defined as $A \oplus B = (A \cup B) \setminus (A \cap B)$. Also note that for n sets $A_1, \ldots, A_n$, the expression $A_1 \oplus \ldots \oplus A_n$ is equal to the set of elements appearing in an odd number of the sets $A_1, \ldots, A_n$.

2.1 Definition of $\oplus$-NFAs

Definition 1. *A $\oplus$-NFA M is a 5-tuple $M = (Q, \Sigma, \delta, Q_0, F)$, where Q is the finite non-empty set of states, Σ is the finite non-empty input alphabet, $Q_0 \subseteq Q$ is the set of start states, $F \subseteq Q$ is the set of final states and δ is the transition function such that $\delta : Q \times \Sigma \to 2^Q$.* □

The transition function δ can be extended to $\delta : 2^Q \times \Sigma \to 2^Q$ by defining

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

for any $a \in \Sigma$ and $A \in 2^Q$. The transition function of the $\oplus$-NFA can be extended to $\delta^* : 2^Q \times \Sigma^* \to 2^Q$ by defining $\delta^*(A, \epsilon) = A$ and $\delta^*(A, aw) = \delta^*(\delta(A, a), w)$ for any $a \in \Sigma$, $w \in \Sigma^*$ and $A \in 2^Q$.

Note that, if the size of the alphabet is one (that is, $|\Sigma| = 1$), then the $\oplus$-NFA is called a *unary* $\oplus$-NFA.

Definition 2. *Let M be a $\oplus$-NFA $M = (Q, \Sigma, \delta, Q_0, F)$, and let w be a word in Σ^*. Then M accepts w if and only if $|F \cap \delta(Q_0, w)|$ mod $2 \neq 0$.* □

For any word $w = w_0 w_1 \dots w_k \in \Sigma^*$ read by a $\oplus$-NFA M, there is at least one associated sequence of states $s_0, s_1, \dots, s_{k+1}$ such that $\delta(s_i, w_i) = s_{i+1}$. Such a sequence of states is a *path* for the word w. All possible paths on the word w can be combined into an *execution tree* of M. A path in the execution tree is an *accepting path* if it ends in a final state. It is important to note that in the execution tree of a $\oplus$-NFA, if there is an even number of occurrences of a state s_i on level i, then those states cancel out under the symmetric difference operation, and those paths terminate. If an odd number of occurrences of a state s_i occurs on level i, then none of the s_i cancel out and all their paths remain in the execution tree.

In other words, a $\oplus$-NFA accepts a word w by parity — if there is an odd number of accepting paths for w in the execution tree, then w is accepted; else it is rejected. This parity acceptance is motivated by the algebraic foundations of $\oplus$-NFAs, where unary $\oplus$-NFAs correspond to pseudo-noise sequences [4].

Example 1. Let $M = (\{q_1, q_2, q_3\}, \{a\}, \delta, \{q_1\}, \{q_3\})$ be a $\oplus$-NFA where δ is given by

δ	a
q_1	$\{q_2\}$
q_2	$\{q_3\}$
q_3	$\{q_1, q_3\}$.

Figure 1 shows a graphical representation of M; note that there is no visual difference from a traditional NFA. To find the DFA M' equivalent to M, we apply the subset construction using the symmetric difference operation instead of union. The transition function δ' of M' is

δ'	a
$[q_1]$	$[q_2]$
$[q_2]$	$[q_3]$
$*\,[q_3]$	$[q_1, q_3]$
$*\,[q_1, q_3]$	$[q_1, q_2, q_3]$
$*\,[q_1, q_2, q_3]$	$[q_1, q_2]$
$[q_1, q_2]$	$[q_2, q_3]$
$*\,[q_2, q_3]$	$[q_1]$.

□

Note that each accepting state is marked by a '$*$'.

It is easy to see that any unary $\oplus$-NFA is an autonomous linear machine (see [13, 15] for a formal exposition). As such, one can encode the transition table of a unary $\oplus$-NFA M as a binary matrix $\mathbf{A}$:

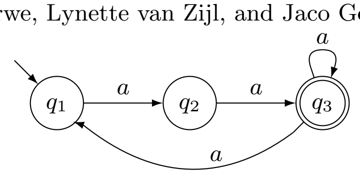

Fig. 1. The ⊕-NFA for Example 1.

$$a_{ji} = \begin{cases} 1 \text{ if } q_j \in \delta(q_i, a) \\ 0 \text{ otherwise,} \end{cases}$$

and successive matrix multiplications in the Galois field $GF(2)$ reflect the subset construction on M.

$\mathbf{A}$ is called the *characteristic matrix* of M, and $c(x) = \det(\mathbf{A} - x\mathbf{I})$ is known as its characteristic polynomial.

Similarly, we can encode any set of states $X \subseteq Q$ as an n-entry row vector $\boldsymbol{v}$ by defining

$$v_i = \begin{cases} 1 \text{ if } q_i \in X \\ 0 \text{ otherwise .} \end{cases}$$

Note that we place an arbitrary but fixed order on the elements of Q. We refer to $\boldsymbol{v}$ as the *vector encoding* of X, and to X as the *set encoding* of $\boldsymbol{v}$.

If $\boldsymbol{y}$ encodes the initial states of a ⊕-NFA M, and $\mathbf{A}$ is its characteristic matrix, then $\mathbf{A}\boldsymbol{y}^{\mathrm{T}}$ encodes the states reachable from the initial state after reading one letter, $\mathbf{A}^2\boldsymbol{y}^{\mathrm{T}}$ encodes the states reachable after two letters, and in general $\mathbf{A}^k\boldsymbol{y}^{\mathrm{T}}$ encodes the states reachable after k letters. If $\boldsymbol{z}$ encodes the final states of M, then standard linear algebra shows the following:

$$M \text{ accepts } a^k \text{ if and only if } \boldsymbol{z}\mathbf{A}^k\boldsymbol{y}^{\mathrm{T}} = 1.$$

Example 2. Consider the ⊕-NFA in Example 1. Its characteristic matrix is

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

and its characteristic polynomial is $c(x) = x^3 + x^2 + 1$. Interested readers may note that $c(x)$ is a primitive polynomial in $GF(2)$. If we encode the start state as a column vector $\boldsymbol{y}^T$, with only the first component of $\boldsymbol{y}$ equal to one, and compute $\mathbf{A}^k\boldsymbol{y}^T$, we end up with the k-th entry in the on-the-fly subset construction on M. For example, with the start state q_1 encoded as $\boldsymbol{y} = [1\ 0\ 0]$, we see that $\mathbf{A}^4$ is given by

$$\mathbf{A}^4 = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix},$$

and hence $\mathbf{A}^4\boldsymbol{y}^T$ is given by $[\,1\ 1\ 1\,]$. This corresponds to the state $[\,q_1, q_2, q_3\,]$, which is reached after four applications of the subset construction on M. Similarly, $\mathbf{A}^6\boldsymbol{y}^T$ is given by $[\,0\ 1\ 1\,]$, which corresponds to $[\,q_2, q_3\,]$. □

2.2 Analysis of ⊕-NFA Behaviour

In [15] we formally showed that the state behaviour of a unary ⊕-NFA is the same as that of a linear feedback shift register (LFSR). The similarity is intuitively straightforward, as an LFSR is a linear machine over $GF(2)$, and we can encode a unary ⊕-NFA as a linear machine over $GF(2)$ as shown above. This correspondence means that we can exploit the wealth of literature on LFSRs to analyse the behaviour of unary ⊕-NFAs, and in particular their cyclic behaviour (see, for example, [3] or [13]).

Because a unary ⊕-NFA is characterized by a matrix in $GF(2)$, one can perform a change of basis without changing the accepted language. It is precisely this observation that we use in the proof of the next theorem.

Theorem 1. *Let $M = (Q, \{a\}, \delta, Q_0, F)$ be a unary n-state ⊕-NFA that accepts a non-empty language L. Then for any non-empty subset of states $X \subseteq Q$ there exists unary ⊕-NFAs M' and M'', both accepting L, such that:*

1. *$M' = (Q, \{a\}, \delta', X, F')$, and*
2. *$M'' = (Q, \{a\}, \delta'', Q_0'', X)$.*

Proof. Both claims are based on the same principle; we only show the first. Let $\boldsymbol{s}$, $\boldsymbol{f}$, and $\boldsymbol{x}$ be the vector encodings of sets Q_0, F, and X, respectively, and let $\mathbf{A}$ be the characteristic matrix of M. Choose a non-singular matrix[1] $\mathbf{P}$ such that $\boldsymbol{x}^T = \mathbf{P}\boldsymbol{s}^{\mathrm{T}}$. Note that Q_0 and F are non-empty sets, since L is non-empty, and thus $\boldsymbol{s}$ and $\boldsymbol{f}$ are non-zero. Let $\boldsymbol{y} = \boldsymbol{f}\mathbf{P}^{-1}$, and let $\mathbf{B} = \mathbf{P}\mathbf{A}\mathbf{P}^{-1}$. Let F' be the set encoding of $\boldsymbol{y}$ and define δ' by

$$q_i \in \delta'(q_j, a) \text{ if and only if } b_{ij} = 1.$$

Then

$$\begin{aligned}
&M' \text{ accepts } a^k \\
\Leftrightarrow\ &\boldsymbol{y}\mathbf{B}^k\boldsymbol{x}^{\mathrm{T}} = 1 \\
\Leftrightarrow\ &(\boldsymbol{f}\mathbf{P}^{-1})(\mathbf{P}\mathbf{A}\mathbf{P}^{-1})^k(\mathbf{P}\boldsymbol{s}^{\mathrm{T}}) = 1 \\
\Leftrightarrow\ &\boldsymbol{f}\mathbf{A}^k\boldsymbol{s}^{\mathrm{T}} = 1 \\
\Leftrightarrow\ &M \text{ accepts } a^k.
\end{aligned}$$

□

[1] Such a matrix $\mathbf{P}$ must exist. We can obtain $\mathbf{P}$ as the product $\mathbf{P_1}\mathbf{P_2}^{-1}$, where $\mathbf{P_1}$ and $\mathbf{P_2}$ are non-singular matrices obtained as follows: Denote by $\boldsymbol{e_1}$ the vector with a 1 in the first component and 0's in all the other components. Let $\mathbf{P_1}$ and $\mathbf{P_2}$ be non-singular matrices such that $\mathbf{P_1}\boldsymbol{e_1}^{\mathrm{T}} = \boldsymbol{x}^{\mathrm{T}}$ and $\mathbf{P_2}\boldsymbol{e_1}^{\mathrm{T}} = \boldsymbol{s}^{\mathrm{T}}$. We construct for example $\mathbf{P_1}$ (and similarly $\mathbf{P_2}$) by setting the first column of $\mathbf{P_1}$ to be equal to $\boldsymbol{x}^{\mathrm{T}}$ and then select the remaining columns in any way such that the column vectors of $\mathbf{P_1}$ are linearly independent.

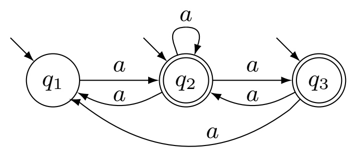

Fig. 2. The ⊕-NFA from Example 3.

In essence, the technique in Theorem 1 changes the basis of the characteristic matrix. This makes it possible to transform any unary ⊕-NFA into an equivalent automaton while controlling either the choice of start states, final states, or (non-singular) characteristic matrix.

Example 3. Consider again the ⊕-NFA M in Example 1. To transform it to an automaton where all of the states are start states, we solve $\boldsymbol{x} = \mathbf{P}\boldsymbol{s}^{\mathrm{T}}$. We know from the definition of M that $\boldsymbol{s} = [1\ 0\ 0]$ and we want $\boldsymbol{x} = [1\ 1\ 1]$. We can take for example

$$\mathbf{P} = \begin{bmatrix} 1\,0\,0 \\ 1\,1\,0 \\ 1\,1\,1 \end{bmatrix} \quad \text{and} \quad \mathbf{P}^{-1} = \begin{bmatrix} 1\,0\,0 \\ 1\,1\,0 \\ 0\,1\,1 \end{bmatrix}$$

and therefore

$$\mathbf{B} = \mathbf{PAP}^{-1} = \begin{bmatrix} 0\,1\,1 \\ 1\,1\,1 \\ 0\,1\,0 \end{bmatrix}$$

and the vector encoding for the final states is $\boldsymbol{y} = \boldsymbol{f}\mathbf{P}^{-1} = [0\ 1\ 1]$. The resulting automaton is shown in Figure 2. One can also verify the fact that the language is not changed by the change in basis, by calculating the DFA corresponding to the new ⊕-NFA:

δ'	a
$[q_1, q_2, q_3]$	$[q_2, q_3]$
$[q_2, q_3]$	$[q_3]$
$*\,[q_3]$	$[q_1, q_2]$
$*\,[q_1, q_2]$	$[q_1, q_3]$
$*\,[q_1, q_3]$	$[q_1]$
$[q_1]$	$[q_2]$
$*\,[q_2]$	$[q_1, q_2, q_3]$.

It is easy to check that this DFA is isomorphic to the DFA in Example 1. □

2.3 Ambiguity

We briefly state the formal definitions for ambiguity.

Definition 3. Unambiguous: *An NFA or $\oplus$-NFA is said to be unambiguous if every word in the language is accepted with at most one accepting path.*

Definition 4. Finitely ambiguous: *An NFA or $\oplus$-NFA is said to be finitely ambiguous if every word in the language is accepted with at most k accepting paths, where k is a positive integer.*

Definition 5. Polynomially ambiguous: *An NFA or $\oplus$-NFA is said to be polynomially ambiguous if every word in the language is accepted with at most k accepting paths, where k is bound polynomially in the length of the input word.*

Definition 6. Exponentially ambiguous: *An NFA or $\oplus$-NFA is said to be exponentially ambiguous if every word in the language is accepted with at most k accepting paths, where k is bounded exponentially in the length of the input word.*

Ambiguity for a given NFA or $\oplus$-NFA can be visually demonstrated by drawing the execution tree of the corresponding automaton. Given an NFA or $\oplus$-NFA M, we assume that the root is on level 0 of the execution tree, and is a start state of M. Note that, if M has multiple start states, then one considers the forest of execution trees, where the root of each tree is one of the start states of M.

3 Ambiguity of Unary $\oplus$-NFAs

It is known that the conversion of a traditional n-state NFA to a DFA has an upper bound of $O(2^n)$ states, and this bound is tight [7, 8]. This does not hold in the case of unary NFAs, where the bound is $g(n) + n^2$ states, where $g(n)$ is Landau's function [1]. In the case of unary n-state $\oplus$-NFAs, we recall a tight upper bound of $2^n - 1$ for the $\oplus$-NFA to DFA conversion [16].

Okhotin [9] showed that the $g(n) + n^2$ bound also holds for the unary NFA to unary unambiguous NFA trade-off. Surprisingly, the unary $\oplus$-NFA to unary finitely ambiguous $\oplus$-NFA trade-off is simply linear – any unary n-state $\oplus$-NFA has an equivalent n-state finitely ambiguous $\oplus$-NFA, as we show in detail below.

For any state in a $\oplus$-NFA, its *indegree* denotes the number of transitions entering the state in its graphical representation.

Theorem 2. *Any unary n-state $\oplus$-NFA such that each state has indegree at most two, is finitely ambiguous.*

Proof. Let N be any unary $\oplus$-NFA such that each state has indegree at most two. We have an execution tree associated with each initial state in N, but it is enough to show that there are finitely many accepting paths for a given word in each of the execution trees. Thus we may in fact assume that we have only one initial state in N and hence a single execution tree. First we show by induction that each state appears at most once on a given level in the execution tree. This is certainly true at the root of the tree where we have only the initial state.

Assume the result for level i in the execution tree and consider level $i+1$, and let q be any state in N. Since the indegree of q is at most two, precisely one of the following three statements hold:

- there is no transition from a state at level i to state q, and therefore state q is not present at level $i+1$;
- there is a transition from a single state at level i to state q;
- there are transitions from precisely two states at level i to state q, and thus by definition of an $\oplus$-NFA, state q is not present at level $i+1$.

We thus conclude that each state appears at most once on a given level in the execution tree. Since we therefore have only finitely many acceptance states at each level in the execution tree, we conclude that N is finitely ambiguous. □

Theorem 3. *Let M be any unary $\oplus$-NFA. Then there exists a finitely ambiguous unary $\oplus$-NFA N, accepting the same language as M, and also with the same number of states as M.*

Proof. Let $\mathbf{A}$ be the characteristic matrix of M.

First recall from linear algebra that the companion matrix of the monic polynomial $p(x) = c_0 + c_1 x + \ldots + c_{n-1}x^{n-1} + x^n$ over $GF(2)$ is the square matrix

$$\mathbf{C}(p) = \begin{bmatrix} 0 & 0 \ldots 0 & c_0 \\ 1 & 0 \ldots 0 & c_1 \\ 0 & 1 \ldots 0 & c_2 \\ \vdots & \vdots \quad\;\; \vdots & \vdots \\ 0 & 0 \ldots 1 & c_{n-1} \end{bmatrix} .$$

A classic result from linear algebra (see [11], Theorem 7.14), states that $\mathbf{A}$ has a rational canonical form with entries from $GF(2)$. More precisely, there exists an invertible matrix $\mathbf{Q}$ such that $\mathbf{B} = \mathbf{Q}^{-1}\mathbf{AQ} = \mathrm{diag}[\mathbf{L}(f_1), \mathbf{L}(f_2), \ldots, \mathbf{L}(f_s)]$, where $\mathbf{L}(f_i)$ is the companion matrix for a monic polynomial f_i. By $\mathrm{diag}[\mathbf{A_1}, \mathbf{A_2}, \ldots, \mathbf{A_s}]$, with the $\mathbf{A}_i$'s square matrices, we denote the matrix with the $\mathbf{A}_i$'s on the diagonal and all other entries equal to 0. In our context, the important property of the matrix $\mathrm{diag}[\mathbf{L}(f_1), \mathbf{L}(f_2), \ldots, \mathbf{L}(f_s)]$ is that the indegree of each state of a $\oplus$-NFA N, with characteristic matrix $\mathrm{diag}[\mathbf{L}(f_1), \mathbf{L}(f_2), \ldots, \mathbf{L}(f_s)]$, must be at most two. One can see this by noting that we have at most two non-zero entries in each row of $\mathrm{diag}[\mathbf{L}(f_1), \mathbf{L}(f_2), \ldots, \mathbf{L}(f_s)]$. We thus apply a change of basis (by using Q) to $\mathbf{A}$ to obtain the characteristic matrix $\mathbf{B}$ of a $\oplus$-NFA N, and also to the initial and accepting vectors of M to obtain the initial and accepting vectors for N. Since a change of basis preserves the language accepted by a $\oplus$-NFA, M and N accept the same language. Note that M and N will also have the same number of states, and from the previous result we have that N is finitely ambiguous.

□

4 Other Ambiguity Results

There are a number of other results that follow from Theorem 2 in the previous section, in particular for the complement of a language, and for k-deterministic $\oplus$-NFAs.

We note that for any n-state unary $\oplus$-NFA M which accepts a language L, it is possible to construct an $n+1$-state unary $\oplus$-NFA M' such that M' accepts the complement $\overline{L}$. This is in constrast to traditional unary NFAs [9], where the state complexity of complementation for unambigous unary finite automata lies between the bounds $\frac{1}{42}n\sqrt{n}$ (for $n \geq 867$) and $e^{\Theta}(\sqrt[3]{n \ln^2 n})$.

Theorem 4. *Let M be a unary $\oplus$-NFA accepting the language L. Then there is an $n+1$-state unary $\oplus$-NFA M' such that M' accepts $\overline{L}$.*

Proof. By construction. Let $M = (Q, \{a\}, \delta, Q_0, F)$ such that $Q = \{q_1, \ldots, q_n\}$. We introduce a new state $q_0 \notin Q$. Define $M' = (Q \cup \{q_0\}, \{a\}, \delta', Q_0 \cup \{q_0\}, F \cup \{q_0\})$, where $\delta'(q_i, a) = \delta(q_i, a)$ when $i > 0$ and $\delta'(q_0, a) = \{q_0\}$. The DFA equivalent to M' is isomorphic to that of M, and during the subset construction the subsets are identical, except that q_0 is added to every single subset. The execution of M' is identical to that of M, except that an independent linear branch consisting of q_0 states is added "on the side".

Suppose that M accepts the word $w = a^k$. This means that there is an odd number of final states on level k of M's execution tree. On the other hand, in M''s execution tree there is one extra state (q_0) on level k, which therefore contains an even number of final states. Hence M' does not accept w.

The same argument shows that if M does not accept the word $w = a^m$, then M' *does* accept w. Hence M' accepts the language $\overline{L}$. □

Theorem 5. *Assume the language L is accepted by an n-state unary finitely ambiguous $\oplus$-NFA M. Then there is an $n+1$-state unary $\oplus$-NFA M' that accepts $\overline{L}$.*

Proof. Directly from Theorem 2. □

We now consider the ambiguity of k-deterministic unary $\oplus$-FAs. A k-deterministic FA (k-DFA) or $\oplus$-FA (k-$\oplus$-DFA) is a deterministic finite automaton, except that it has multiple initial states. Hence the only nondeterminism in a k-DFA occurs in its multiple initial states.

Definition 7. *A k-DFA M is a 5-tuple $M = (Q, \Sigma, \delta, Q_0, F)$, where Q is the finite non-empty set of states, Σ is the finite non-empty input alphabet, $Q_0 \subseteq Q$ is the set of start states, $F \subseteq Q$ is the set of final states and δ is the transition function such that $\delta : Q \times \Sigma \rightarrow Q$.* □

Note that, as before, the difference between a traditional k-DFA and a k-$\oplus$-DFA lies in the application of the subset construction to get the equivalent DFA (without multiple initial states).

It is to be expected that a unary k-DFA should be either unambiguous or finitely ambiguous, and this is indeed the case both for the traditional k-DFA and the k-$\oplus$-DFA:

Theorem 6. *Any unary k-DFA or k-⊕-DFA is finitely ambiguous, with the constant for the finite ambiguity no more than* $|Q_0|$.

Proof. The multiple initial states lead to a forest of disconnected execution trees, such that each tree is a single deterministic branch. In the case of k-DFAs the ambiguity is determined by the number of possible final states on each level, which is bounded below by zero and above by $|Q_0|$. Note that the number of trees stay constant. Hence, the k-DFA is finitely ambiguous.

In the case of k-⊕-DFAs, the number of deterministic trees in the forest of disconnected execution trees cannot be more than $|Q_0|$, but may become less if an even number of identical states occur on the same level. The result holds by the same argument as above. □

It is interesting to note, however, that there exists a family $\{M_n\}_{n\geq 2}$ of k-DFAs that are finitely ambiguous when considered as k-DFAs, but are unambiguous when considered as k-⊕-DFAs.

Theorem 7. *There exists a family* $\{M_n\}_{n\geq 2}$ *of unary k-DFAs that are finitely ambiguous when considered as k-DFAs, but are unambiguous when considered as k-⊕-DFAs.*

Proof. By construction. Let $M_n = (Q, \{a\}, \delta, Q_0, F)$ be defined by $Q = \{q_1, \ldots, q_{n-1}\}$, $Q_0 = Q$, $F = \{q_{n-1}\}$ and $\delta(q_i, a) = q_{i+1}$ for $1 \leq i < n-1$, and $\delta(q_{n-1}, a) = q_{n-2}$.

Consider first the k-DFA case. The forest of execution trees contains $n = |Q_0|$ disconnected trees with no branches. Hence, on any level, there are exactly n states. The transition function ensures that each tree has the form $q_i \rightarrow q_{i+1} \rightarrow \ldots q_{n-2} \rightarrow q_{n-1} \rightarrow q_{n-2} \rightarrow q_{n-1} \ldots$. In other words, each tree consists of a single branch with consecutive states, until the last two states alternate indefinitely. Now, since all the elements in Q_0 are distinct, the trees all reach the state q_{n-1} within $n-1$ steps. Hence, from step n onwards, there are at most $|Q_0|$ final states at any level, and hence the k-DFA is finitely ambiguous.

In the case of the k-⊕-DFA, we note that on level one of the execution tree, there are n distinct states $\{q_1, q_2, \ldots, q_{n-1}\}$, and hence one final state. On level two, states q_{n-3} and q_{n-1} both result in state q_{n-2}, and symmetric difference causes the trees with state q_{n-2} to terminate. This process continues with the other branches, until only one tree remains which alternates between states q_{n-1} and q_{n-2}, or which ends in an empty set of states. Hence, the k-⊕-DFA is unambiguous. □

5 Conclusion and Future Work

We showed that, for any n-state unary ⊕-NFA, there is an equivalent n-state unary ⊕-NFA which is finitely ambiguous. This implies that, for unary regular languages, there are no languages which are strictly polynomially or strictly exponentially ambiguous with ⊕-NFAs. This result also holds for traditional unary NFAs, and a further avenue of investigation is to determine whether any kind of unary NFA (such as a ∩-NFA or XNOR-NFA) must be at most finitely ambiguous.

It also remains to investigate ambiguity issues in more detail for non-unary languages.

References

1. Chrobak, M.: Finite automata and unary languages. Theoretical Computer Science 47(3), 149–158 (1986), erratum appeared as [2]
2. Chrobak, M.: Errata on "Finite automata and unary languages". Theoretical Computer Science 302(1–3), 497–498 (Jun 2003)
3. Dornhoff, L., Hohn, F.: Applied Modern Algebra. Macmillan Publishing Company (1978)
4. Goresky, M., Klapper, A.: Pseudonoise sequences based on algebraic feedback shift registers. IEEE Transactions on Information Theory 52(4), 1649–1662 (Apr 2006)
5. Leung, H.: Separating exponentially ambiguous finite automata from polynomially ambiguous finite automata. SIAM Journal of Computing 27(4), 1073–1082 (Aug 1998)
6. Leung, H.: Descriptional complexity of NFA of different ambiguity. International Journal of Foundations of Computer Science 16(5), 975–984 (Oct 2005)
7. Meyer, A.R., Fischer, M.J.: Economy of description by automata, grammars, and formal systems. In: Proceedings of the 12 Annual IEEE Symposium on Switching and Automata Theory. pp. 188–191 (Oct 1971)
8. Moore, F.R.: On the bounds for state-set size in the proofs of equivalence between deterministic, nondeterministic and two-way finite automata by deterministic automata. IEEE Transactions on Computing C–20(10), 1211–1219 (Oct 1971)
9. Okhotin, A.: Unambiguous finite automata over a unary alphabet. In: Hlinený, P., Kucera, A. (eds.) Proceedings of the 35th International Symposium on the Mathematical Foundations of Computer Science. pp. 556–567. LNCS #6281, Springer-Verlag (Aug 2010)
10. Ravikumar, B., Ibarra, O.H.: Relating the type of ambiguity of finite automata to the succinctness of their representation. SIAM Journal of Computing 18(6), 1263–1282 (Dec 1989)
11. Roman, S.: Advanced Linear Algebra. Springer-Verlag (1992)
12. Sipser, M.: Introduction to the Theory of Computation. International Thomson Publishing (1996)
13. Stone, H.: Discrete Mathematical Structures. Science Research Associates (1973)
14. Vuillemin, J., Gama, N.: Efficient equivalence and minimization for nondeterministic XOR automata. Tech. rep., INRIA (Dec 2009)
15. van Zijl, L.: Generalized Nondeterminism and the Succinct Representation of Regular Languages. Ph.D. thesis, Stellenbosch University (Nov 1997)
16. van Zijl, L.: Magic numbers for symmetric difference NFAs. International Journal of Foundations of Computer Science 16(5), 1027–1038 (Oct 2005)
17. van Zijl, L., Geldenhuys, J.: Descriptional complexity of ambiguity in symmetric difference NFAs. Journal of Universal Computer Science (2011), accepted for publication