Descriptional Complexity of Ambiguity in Symmetric Difference NFAs

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

Jaco Geldenhuys
(Stellenbosch University, Stellenbosch, South Africa
jaco@cs.sun.ac.za)

Abstract: We investigate ambiguity for symmetric difference nondeterministic finite automata. We show the existence of unambiguous, finitely ambiguous, polynomially ambiguous and exponentially ambiguous symmetric difference nondeterministic finite automata. We show that, for each of these classes, there is a family of n-state nondeterministic finite automata such that the smallest equivalent deterministic finite automata have $O(2^n)$ states.

Key Words: nondeterminism, ambiguity, succinctness
Category: F.1.1, F.1.2

1 Introduction

The ambiguity of nondeterministic finite automata (NFAs) measures the maximum number of different accepting paths for all the words in the language accepted by the NFA. An NFA is said to be k-ambigious if every word in the language is accepted with at most k different accepting paths [Leung(1998)]. The bound on the number of different accepting paths then indicates whether a given NFA is unambigious ($k = 1$), finitely ambigious (k is a positive integer), polynomially ambigious (the bound is polynomial in the length of the input word) or exponentially ambigious. The concept of ambiguity has been investigated extensively (for example, see [Leung(2005), Ravikumar and Ibarra(1989), Weber and Seidl(1991)]).

There are a number of questions related to ambiguity. Firstly, of interest is showing the existence of families of NFAs such that the family exhibits a specific type of ambiguity. Secondly, one would wish to demonstrate for each type of ambiguity, the existence of a family of NFAs which belongs *strictly* to that class. Thirdly, it is interesting to consider the descriptional complexity of the family of NFAs of the given ambiguity [Leung(2005), Ravikumar and Ibarra(1989)]. These questions have mostly been solved for standard NFAs. However, ambiguity in symmetric difference NFAs ($\oplus$-NFAs) has not yet been investigated. Moreover, $\oplus$-NFAs were recently shown to have interesting properties as far as descriptional complexity is concerned. In particular, the $\oplus$-NFAs can succinctly represent languages with cyclic properties [Van Zijl(2004)]. Hence, in this work we consider ambiguity for $\oplus$-NFAs, and in particular the first and third questions above, namely, the existence of families of ambigious $\oplus$-NFA, as well as their descriptional complexity in each case. We shall demonstrate for each ambiguity class, a family of n-state $\oplus$-NFAs such that their minimal equivalent DFAs have $O(2^n)$ states.

2 Definitions

We assume that the reader has a basic knowledge of automata theory, as for example in [Sipser(1997)]. We briefly define the concepts used in the rest of this work.

Definition 1. DFA: A deterministic finite automaton (DFA) M is a 5-tuple $M = (Q, \Sigma, \delta, q_0, F)$, where Q is a finite non-empty set of states, Σ is a finite non-empty set of alphabet symbols, $q_0 \in Q$ is the start state and $F \subseteq Q$ is a set of final states. The transition function is $\delta : Q \times \Sigma \to Q$.

Definition 2. Acceptance: Let Σ^* denote the Kleene closure of Σ, and let $w = w_0 w_1 \dots w_k$ be a string in Σ^*. Then a given DFA $M = (Q, \Sigma, \delta, q_0, F)$ accepts the string w iff there is a sequence of states $s_0, s_1, \dots, s_{k+1}$, where $s_i \in Q$ for all $0 \le i \le k+1$, such that $s_0 = q_0$, $s_{k+1} \in F$, and $\delta(s_i, w_i) = s_{i+1}$, for all $0 \le i \le k$.

In the definition of acceptance, the sequence of states $s_0, s_1, \dots, s_{k+1}$ is called an accepting path for the string w. Also, the set of all strings (words) accepted by a DFA forms the language accepted by the DFA.

It is often of interest to consider classes of words with particular properties. In our case, we are interested in so-called *primitive words*:

Definition 3. Primitive word: For a word $w = w_0 w_1 \dots w_k$, the string $w_0 \dots w_i$ is a prefix of w, for every i such that $0 \le i \le k$. A word w is called *primitive* if it cannot be written as s^t where s is a prefix of w and $t > 1$[Domaratzki et al.(2002)].

In contrast to a DFA, an NFA allows multiple choices for each transition. Let 2^Q indicate the power set of Q. Then we define an NFA as follows:

Definition 4. NFA: A nondeterministic finite automaton (NFA) M is a 5-tuple $M = (Q, \Sigma, \delta, Q_0, F)$, where Q is a finite non-empty set of states, Σ is a finite non-empty set of alphabet symbols, $Q_0 \subseteq Q$ is the set of start states and $F \subseteq Q$ is a set of final states. The transition function is $\delta : Q \times \Sigma \to 2^Q$.

Acceptance for an NFA is defined as follows:

Definition 5. NFA acceptance: An NFA accepts a word $w \in \Sigma^*$ if there is at least one accepting path $s_0 s_1 \dots s_{k+1}$ for $w = w_0 w_1 \dots w_k$, where $s_0 \in Q_0$, $s_{k+1} \in F$ and $s_{i+1} \in \delta(s_i, w_i)$ for all $0 \leq i \leq k$.

Note that there can be many paths on the given word w, as there is a choice of several subsequent states at each state. The combination of all the possible paths for a given string w forms an acyclic graph. This graph is known as the execution tree. The root of the execution tree is a start state of the NFA. The nodes on level $i+1$ are given by calculating $\delta(q_j, w_i)$ for every node q_j on level i. Note that the different branches of the execution tree are independent of each other (this will not be the case for the execution tree of a $\oplus$-NFA).

Definition 6. Execution tree (NFA): Let $M = (Q, \Sigma, \delta, q_0, F)$ be an NFA. Define $G = (V, E)$ as the acyclic graph such that $V = Q$, and $q_0 \in V$ is the root of G. Let $w = w_0 w_1 \dots w_k \in \Sigma^*$, and consider any associated sequence of states $s_0, s_1, \dots s_{k+1}$ such that $s_{i+1} \in \delta(s_i, w_i)$, for all $0 \leq i \leq k$. Then $(s_i, s_{i+1}) \in E$ and we say that s_i is a node on level i.

The class of NFAs cannot accept more languages than the class of DFAs:

Theorem 7. Subset construction: *Any NFA $M = (Q, \Sigma, \delta, q_0, F)$ has an equivalent DFA $M' = (Q', \Sigma, \delta', q_0, F')$ which accepts the same language, and which can be found by using the subset construction: Let $Q' = 2^Q$. Then, for any $A \subseteq Q$,*

$$\delta'(A, \sigma) = \bigcup_{q \in A} \delta(q, \sigma) \quad .$$

Also, $A \in F'$ iff there is at least one $q \in A$ such that $q \in F$. That is, a state A in the DFA is a final state if there is at least one element in A which is a final state in the original NFA.

Proof. See [Sipser(1997)].

NFAs, however, do have an advantage over DFAs, in that there are regular languages that can be recognized by n-state NFAs, but for which the smallest DFA recognizing that language requires 2^n states. This difference in descriptional complexity is called *succinctness*:

Definition 8. Succinctness: An NFA $M = (Q, \Sigma, \delta, q_0, F)$, with $|Q| = n$, is called succinct if its equivalent minimal DFA requires at least $O(2^n)$ states.

We now consider symmetric difference NFAs ($\oplus$-NFAs) – for more detail, see for example [Dornhoff and Hohn(1977), Van Zijl(2004)]. We use symmetric difference here in the normal set-theoretic sense, so that $A \oplus B = (A \cup B) \backslash (A \cap B)$

for any two sets A and B. Then, a $\oplus$-NFA is simply an NFA, except that the subset construction is applied using symmetric difference:

$$\delta'(A, \sigma) = \bigoplus_{q \in A} \delta(q, \sigma) \quad .$$

This seemingly simple change in the subset construction has far-reaching implications for the behavioural and language-theoretic properties of NFAs versus $\oplus$-NFAs. Given an NFA $M = (Q, \Sigma, \delta, q_0, F)$ and $\oplus$-NFA $M' = (Q, \Sigma, \delta, q_0, F)$ with identical definitions, their execution trees are typically different, and they accept different languages. Hence, the behaviour of $\oplus$-NFAs with respect to ambiguity, is not the same as for NFAs.

Acceptance for a $\oplus$-NFA is again defined as for traditional NFAs [1]. However, execution trees for $\oplus$-NFAs require a new definition:

Definition 9. Execution tree ($\oplus$-NFA): Let $M = (Q, \Sigma, \delta, q_0, F)$ be an $\oplus$-NFA. Define $G = (V, E)$ as the acyclic graph such that $V = Q$, and $q_0 \in V$ is the root of G. Let $w = w_0 w_1 \ldots w_k \in \Sigma^*$, and consider all associated sequences of states $s_0^j, s_1^j, \ldots, s_{k+1}^j$ such that $s_{i+1} \in \delta(s_i, w_i)$, for $1 \leq j \leq m$. If m is even, then $(s_i, s_{i+1}) \notin E$.

We note that the definition of the execution tree above implies that the different accepting paths interact in the execution tree, in the sense that there can only ever be an odd number of occurrences of a node on a given level – if an even number occurs, then all these nodes are cancelled out and their corresponding paths terminate on the previous level.

$\oplus$-NFAs with one alphabet symbol (unary $\oplus$-NFAs) have been investigated extensively, as these machines are equivalent to linear feedback shift registers (LFSRs) [Dornhoff and Hohn(1977)] and have many practical applications such as random number generation and perfect hashing [Chaudhuri et al.(1997)]. As we make extensive use of unary $\oplus$-NFAs, we now give a short summary of their properties (for more detail, see [Dornhoff and Hohn(1977), Van Zijl(2004)]).

For an n-state unary $\oplus$-NFA, an $n \times n$ binary matrix $A = [a_{ij}]_{n \times n}$ over the Galois field GF(2) can be used[2] to encode its transition function so that for every state $q_i \in Q$,

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

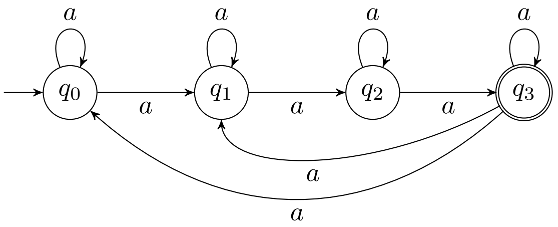
[1] Another variation [Van Zijl(2004), Vuillemin and Gama(2009)] on acceptance is possible when acceptance is defined to reflect the parity characteristic of the symmetric difference operation. That is, a string is accepted if there is an odd number of accepting paths.

[2] GF(2) has the usual binary addition and multiplication, but with the property that $1 + 1 = 0$. In other words, it behaves like an XOR gate.

The powers of A then represent the successive states of the equivalent DFA. Any such matrix A also has a characteristic polynomial $c(X) = \det(A - IX)$. The properties of $c(X)$ determine the cyclic behaviour of the $\oplus$-NFA. In particular, if for an n-state $\oplus$-NFA its characteristic polynomial $c(X)$ is primitive and irreducible, then it is known that the corresponding DFA has $2^n - 1$ states.

Consider the following example which contrasts a unary NFA with a unary $\oplus$-NFA.

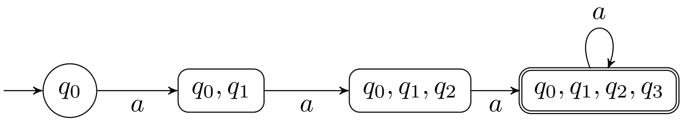
Example 1. Let $M = (\{q_0, q_1, q_2, q_3\}, \{a\}, \delta, \{q_0\}, \{q_3\})$ be defined as below:



Here, δ is given by

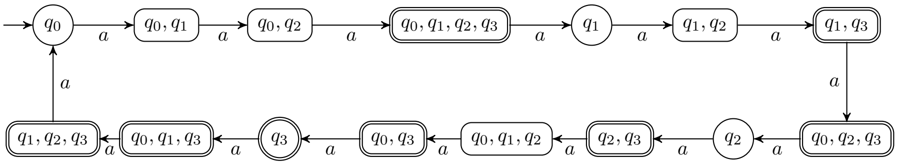
δ	a
q_0	$\{q_0, q_1\}$
q_1	$\{q_1, q_2\}$
q_2	$\{q_2, q_3\}$
q_3	$\{q_0, q_1, q_3\}$.

When we consider M to be an NFA, its equivalent DFA is given by



The language accepted by M is then $L_\cup = \{a^k \mid m \geq 3\}$.

On the other hand, consider the situation where M is a $\oplus$-NFA. Then application of the subset construction results in the DFA below:



Thus, the language accepted by the $\oplus$-NFA M is given by

$$L(M) = \{a^{15k+i} \mid i \in \{3, 6, 7, 9, 11, 12, 13, 14\} \text{ and } k \geq 0\} \quad .$$

The execution tree for the NFA M on the string *aaaa* is given in Figure 1, while the execution tree for M as a $\oplus$-NFA is given in Figure 2.

Note now that the *symmetric difference* of the nodes on a given level represent the corresponding node in the equivalent DFA. This also means that all occurrences of a given node label, on a specific level in the execution tree, are cancelled out if there is an even number of occurrences of that node label on that level. On the other hand, if there is an odd number of occurrences, then none cancel out on that level. For example, on the last level in Figure 2, there are two occurrences of q_0, and hence all q_0's are cancelled out. But since there are three occurrences of q_1, none of the q_1's are cancelled out.

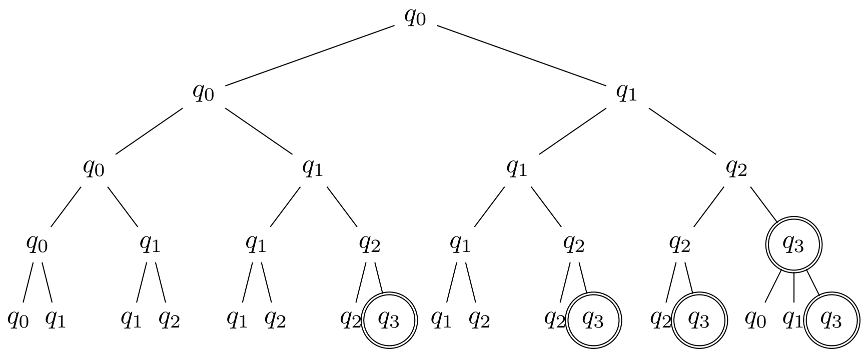

q_0
q_0 q_1
q_0 q_1 q_1 q_2
q_0 q_1 q_1 q_2 q_1 q_2 q_2 q_3
q_0 q_1 q_1 q_2 q_1 q_2 q_2 q_3 q_1 q_2 q_2 q_3 q_2 q_3 q_0 q_1 q_3

Figure 1: Execution tree for NFA M on string aaaa.

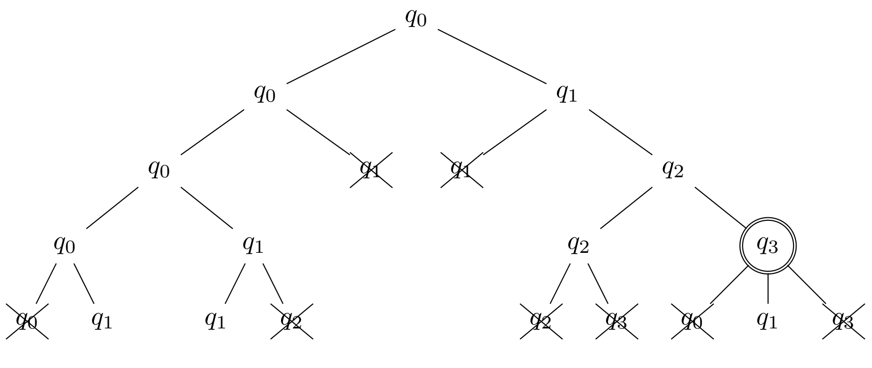


Figure 2: Execution tree for $\oplus$-NFA M on string aaaa.

□

We now give an example to illustrate the $\oplus$-NFA specific properties mentioned above.

Example 2. Let M be the $\oplus$-NFA as defined in Example 1 above. Then the transition function δ is encoded into the matrix A below:

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

Some simple algebraic calculation shows that $c(X) = \det(A - IX)$ is given by $c(X) = X^4 + X + 1$. It is known that this polynomial is primitive and irreducible. Hence, the DFA equivalent to M should have a cycle of length $2^4 - 1 = 15$ states. As shown in Example 1 above, this is indeed the case. The reader may also note that this DFA is minimal.

□

We finallly give formal definitions for the various classes of ambiguity.

Definition 10. Unambigious (UNA,$\oplus$-UNA): An NFA or $\oplus$-NFA is said to be unambigious if every word in the language is accepted with at most one accepting path.

Definition 11. Finitely ambigious (FNA,$\oplus$-FNA): An NFA or $\oplus$-NFA is said to be finitely ambigious if every word in the language is accepted with at most k accepting paths, where k is a positive integer.

Definition 12. Polynomially ambigious (PNA,$\oplus$-PNA): An NFA or $\oplus$-NFA is said to be polynomially ambigious 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 13. Exponentially ambigious (ENA,$\oplus$-ENA): An NFA or $\oplus$-NFA is said to be exponentially ambigious if every word in the language is accepted with at most k accepting paths, where k is bound exponentially in the length of the input word.

We now demonstrate the existence of ambigious $\oplus$-NFAs.

3 Ambiguity for $\oplus$-NFAs

In the following sections, we show the existence of families of unambiguous $\oplus$-NFAs, finitely ambiguous $\oplus$-NFAs, polynomially ambiguous $\oplus$-NFAs and finally

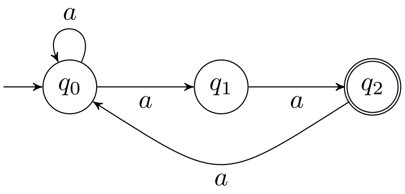


Figure 3: $\oplus$-NFA M_3^u.

exponentially ambiguous $\oplus$-NFAs. Note that each example in essence provides a method for constructing an n-state $\oplus$-NFAs with the required ambiguity. None of the demonstrated families are succinct (that is, the equivalent minimal DFA does not have $O(2^n)$ states), except in the case of the $\oplus$-UNA. We address the issue of succinctness in Section 4.

3.1 Unambiguous unary $\oplus$-NFAs

Consider a unary $\oplus$-NFA M_n^u, such that $M_n^u = (Q, \Sigma, \delta, q_0, F)$, where $Q = \{q_0, q_1, \ldots, q_{n-1}\}$, $\Sigma = \{a\}$, the start state set is $\{q_0\}$ and the final state set is $\{q_{n-1}\}$. Define the transition function δ as follows (see Figure 3 above):

$$\delta(q_i, a) = \begin{cases} \{q_0, q_1\}, & \text{if } i = 0, \\ \{q_{i+1}\}, & \text{if } 0 < i < n-1, \\ \{q_0\}, & \text{if } i = n-1 \quad . \end{cases}$$

As an example, the execution tree for M_3^u on the string $aaaaaaa$ is shown in Figure 4.

We now show that M_n^u is unambiguous.

Theorem 14. *M_n^u is a $\oplus$-UNA.*

Proof: If there is at most one final state on level k of an execution tree, then there is only one path for a string of length k to be accepted. If this holds for every k, then all the words in the language have only one acceptance path and by definition the $\oplus$-NFA is unambiguous. The proof follows by induction, to show that there is at most one final state on each level.

Base case: Let $k = 0$. Since $q_0 \notin F$, the base case holds.

Induction case: Assume that there is at most one final state (that is, q_{n-1}) on level k. We now show that there is at most one final state on level $k + 1$.

We first note that the nodes on every level of the execution tree are all distinct. This always holds, as the execution tree forms branches $q_0 \to q_1 \to \ldots \to q_{n-1}$, each of which is offset by one level which starts when q_0 splits into

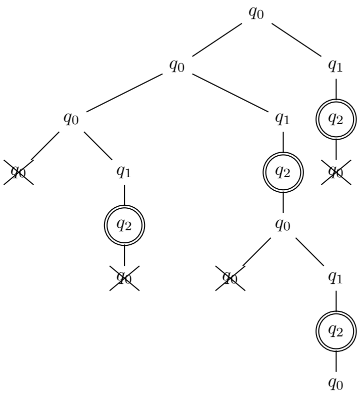


Figure 4: Execution tree of M_3^u on string *aaaaaaa*.

q_0 and q_1. The exception occurs when q_0 and q_{n-1} appear on the same level – in that case, the next level has two occurrences of node q_0. But these two occurrences of q_0 cancel out under symmetric difference, and the other nodes are still distinct. It therefore follows that there can be only one occurrence of any node on any level.

Hence, if there is at most one occurence of q_{n-1} on level k, then there can be at most one occurrence of q_{n-1} on level $k+1$, and only if node q_{n-2} appear in level k.

The result holds by induction. □

We now wish to show that M_n^u is a succinct $\oplus$-UNA. To that end, we first establish the following lemma.

Lemma 15. *Let M be any n-state unary $\oplus$-NFA with one final state, and characteristic polynomial $c(X)$ which is primitive and irreducible. Then its corresponding DFA is minimal.*

Proof. Let $M = (Q, \Sigma, \delta, q_0, F)$ be a unary $\oplus$-NFA, with $|Q| = n$ and $|F| = 1$, such that its characteristic polynomial $c(X)$ is primitive and irreducible. Let $M' = (Q', \Sigma, \delta', q_0', F')$ be its corresponding DFA. Then M' has a cycle of length $2^n - 1$ states (from [Dornhoff and Hohn(1977), Van Zijl(2004)]), which includes all possible subsets of states from Q, except the empty subset. This means that there are exactly 2^{n-1} final states and $2^{n-1} - 1$ nonfinal states in the cycle of the DFA.

Encode the final states in the cycle of M' with ones, and the non-final states with zeroes to obtain a binary sequence w. Then M' is minimal only if w is a primitive word [Domaratzki et al.(2002)]. Suppose that w is not primitive. Then $w = s^m$ for some prefix s of w, and some integer $m > 1$. Hence, if $p = |s|$, then

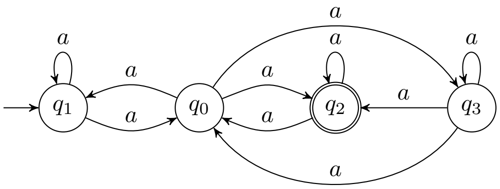


Figure 5: $\oplus$-NFA M_4^f.

$p \times m = |w|$. But $|w|$ is odd, and therefore both p and m must be odd. Let k be the number of ones in s. Then $k \times m = 2^{n-1}$. But this means that the odd number m is a factor of 2^{n-1}, which is impossible. This is a contradiction, and hence w must be primitive and therefore M' must be minimal.

□

Theorem 16. *M_n^u is a succinct $\oplus$-UNA.*

Proof. M_n^u is an n-state unary $\oplus$-NFA. To show succinctness, it is necessary that its equivalent minimal DFA has $O(2^n)$ states. Consider the characteristic polynomial $c(X)$ for M_n^u. Then, for every n such that $c(X)$ is primitive and irreducible, the subset construction for M_n^u yields an equivalent DFA with exactly $2^n - 1$ states. But M_n^u has only one final state. Hence from Lemma 15, it follows that the DFA equivalent to M_n^u is minimal, and therefore M_n^u is succinct.

□

3.2 Finitely ambiguous unary $\oplus$-NFA

Let $M_n^f = (Q, \Sigma, \delta, q_1, F)$, with $F = \{q_{n-2}\}$, and let $\{\overline{q_i}\}$ be taken to mean the complement of q_i over Q, so that $\{\overline{q_i}\} = Q \setminus q_i$. The transition function δ is defined as below:

$$\begin{aligned}
\delta(q_0, a) \quad &= \{\overline{q_0}\} \\
\delta(q_i, a) \quad &= \{q_0, q_i\} \text{ for } 0 < i < n-1 \\
\delta(q_{n-1}, a) &= \begin{cases} \{q_0, q_{n-2}, q_{n-1}\} & \text{for even } n \\ \{q_{n-2}, q_{n-1}\} & \text{for odd } n \end{cases}
\end{aligned}$$

An example of M_4^f is given in Figure 5.

The execution tree for M_4^f is given in Figure 6. It is easy to see that M_4^f has a constant ambiguity of 3. This observation holds for all values of $n > 3$: if the root q_1 is at level 0, then from level 3 onwards, there will always only be one occurrence of q_1 and three occurrences of q_{n-2}.

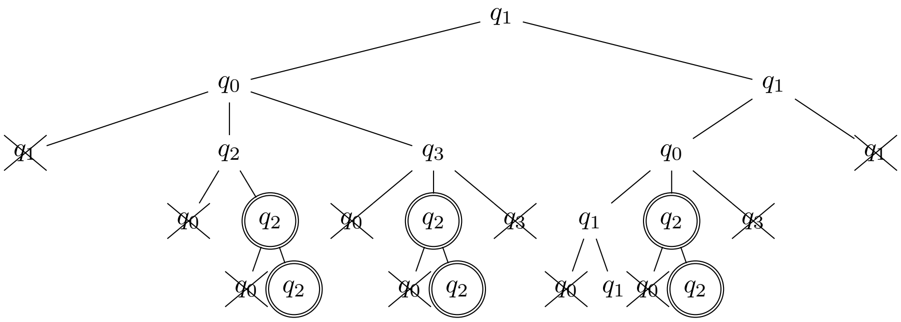


Figure 6: Execution tree of M_4^f.

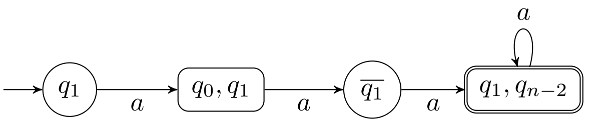


Figure 7: DFA for $\oplus$-FNA M_n^f.

Theorem 17. *M_n^f is a $\oplus$-FNA.*

Proof: Directly from the definition of the transition function: $\delta(q_1, aa)$ leads to two branches with one branch ending in $\overline{q_0}$, and the other branch ending in $\{q_0, q_1\}$, on level two of the execution tree. But as $\delta(q_i, a) = \{q_0, q_i\}$, for $i \neq 0, i \neq n-1$, the symmetric difference ensures that these two branches simply result in one occurrence of q_1 and three occurrences of the final state q_{n-1}. The same argument holds for all succeeding levels of the execution tree. □

We note that M_n^f is not succinct for any values of n. In fact, the DFA has four successive states, with a self-loop on the final state, for all values of n (see Figure 7). In Section 4 we develop a method to take a unary $\oplus$-NFA of any type of ambiguity, and create a binary $\oplus$-NFA with the same ambiguity, but which is succinct. Hence, we show the existence of a succinct $\oplus$-FNA in Section 4.

We now consider polynomial ambiguity.

3.3 Polynomially ambiguous unary $\oplus$-NFA

Let M_n^p be a $\oplus$-NFA with start state set $\{q_0\}$, final state set $\{q_{n-1}\}$, and transition function as below:

$$\delta(q_i, a) = \begin{cases} \{q_1, q_2, \ldots, q_{n-1}\}, & \text{if } i = 0, \\ \{q_i, q_{n-1}\}, & \text{if } 1 \le i \le n-2 \\ \{q_i\}, & \text{if } i = n-1. \end{cases}$$

Figure 8 illustrates the $\oplus$-NFA M_n^p, and Figure 9 shows the execution tree of M_4^p.

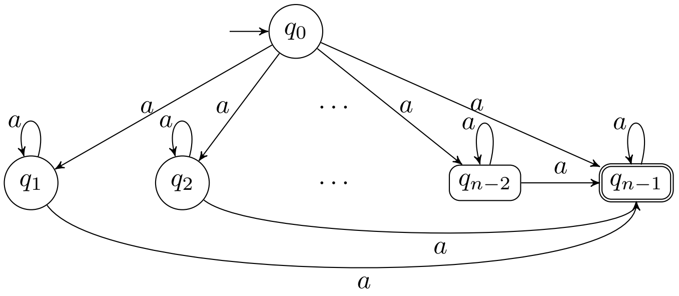


Figure 8: $\oplus$-NFA M_n^p.

Theorem 18. *M_n^p is a $\oplus$-PNA for all even n.*

Proof: It is sufficient to note that the pattern of states in the execution tree of M_n^p ensures that the number of occurrences of state q_{n-1} increases by $n-2$ more for every next level. This is due to the fact that states q_1 to q_{n-2} each add one more occurrence of q_{n-1} for the next level. Therefore, level $k+1$ has $n-2$ more occurrences of q_{n-1} than level k, and a word of length m has $m(m-2)-(m-3)$ different accepting paths. □

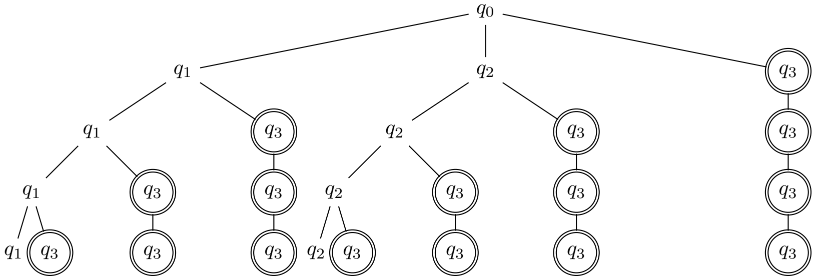


Figure 9: Execution tree of M_4^p.

As for $\oplus$-FNAs, we postpone the presentation of a succinct $\oplus$-PNA to Section 4.

3.4 Exponentially ambiguous unary $\oplus$-NFA

Let M_n^e be a $\oplus$-NFA with start state set $\{q_0\}$ and final state set $\{q_0\}$, and transition function δ given by

$$\delta(q_i, a) = \begin{cases} \{q_0, q_1, q_2, \ldots, q_{n-1}\}, & \text{if } i = 0, \\ \{\overline{q_i}\}, & \text{otherwise.} \end{cases}$$

Figure 10 illustrates M_3^e.

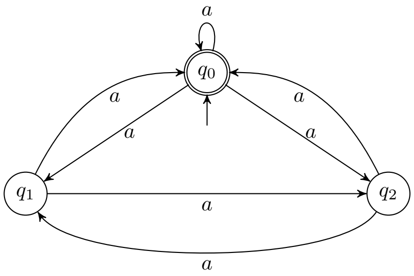


Figure 10: $\oplus$-NFA M_3^e.

Theorem 19. *M_n^e is exponentially ambiguous.*

Proof: Again, by observation of the execution tree. The nodes on alternate levels follow a pattern with k occurrences of q_0 on level i followed by k occurrences of $\{q_0, q_1, \ldots, q_{n-1}\}$. □

4 Finding succinct examples of ambiguous $\oplus$-NFAs

All the $\oplus$-NFAs in the previous section were unary $\oplus$-NFAs. In order to give succinct examples of $\oplus$-NFAs for each type of ambiguity, we use those unary definitions to construct binary $\oplus$-NFAs which are succinct for all n, by a method first illustrated in [Van Zijl(2005)]. Recall that for a unary $\oplus$-NFA M, the succinctness depends on whether the characteristic polynomial $c(X)$ associated with M is primitive and irreducible over GF(2) or not. There is no pattern for a polynomial to be succinct for all n, and hence we cannot give a single $\oplus$-NFA which will be succinct for all n. We force succinctness in the general case by using a binary $\oplus$-NFA such that on one alphabet symbol we force succinctness, and on the other alphabet symbol we force the required ambiguity behaviour.

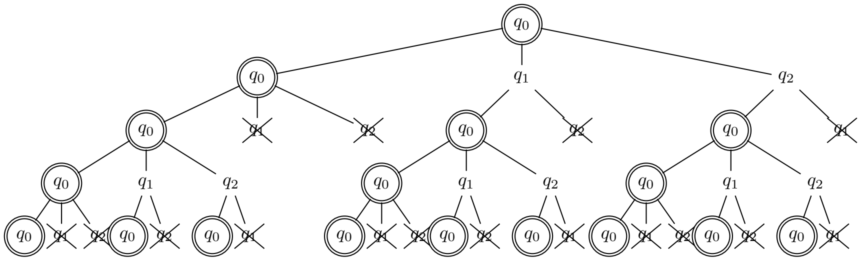


Figure 11: Execution tree of M_3^e.

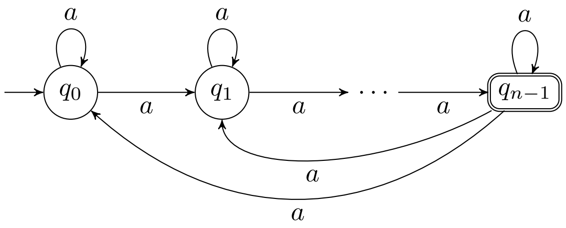
4.1 Forcing succinctness for binary ⊕-NFAs

We briefly recap the method described in [Van Zijl(2005)] – given a unary ⊕-NFA M, how do we construct an $n+1$-state binary ⊕-NFA M' which is succinct and preserves the inherent behaviour of M?
Step 1: Construct a unary n-state ⊕-NFA M_1 with $\Sigma = \{a\}$ such that its corresponding characteristic polynomial is primitive and irreducible. Then M_1 has a cycle of length $2^n - 1$. Number the states of M_1 from q_0 to q_{n-1}.
Step 2: Extend M_1 with an additional state q_n, and choose $\delta(q_n, a) = \emptyset$. Note that q_n is not a reachable state in M_1.
Step 3: Consider the original ⊕-NFA M over $\Sigma = \{b\}$. Remember that M has a given desired property – for example, M could be unambigious. Now extend M with the additional state q_n, and choose $\delta(q_n, b) = q_n$.
Step 4: Introduce q_n into the transition function of M without changing its behaviour. For example, let $q_n \in \delta(q_0, b)$, but in no other transitions from q_1 to q_{n-1}.
Step 5: Merge M and M_1 into one ⊕-NFA M' over $\Sigma = \{a, b\}$, and choose the final state set to be $\{q_n\}$.

It is easy to see M' will have a cycle of length $2^n - 1$ over the alphabet symbol a. However, a final state can be reached only by a word of the form $a^k b$, for some $k \geq 0$. Hence, to accept a word of the form $a^k b$, requires at least $k + 1$ states in the DFA. Therefore, the minimal DFA equivalent to M' has at least $2^n - 1$ states, and hence M' is succinct.

In Example 1 we showed a 4-state unary ⊕-NFA, which is succinct. If we expand the same pattern to an n-state ⊕-NFA, we get

$M = (\{q_0, q_1, \ldots, q_{n-1}\}, \{a\}, \delta, \{q_0\}, \{q_{n-1}\})$:



Here, δ is given by

δ	a
q_0	$\{q_0, q_1\}$
q_1	$\{q_1, q_2\}$
$\vdots$	$\vdots$
q_{n-1}	$\{q_0, q_1, q_{n-1}\}$.

M then has characteristic polynomial $X^n + X + 1$, which is succinct for certain values of n (such as $n = 4$ and $n = 5$). We use this $\oplus$-NFA to force the succinctness of the $\oplus$-NFAs given in the previous sections.

Firstly, the $\oplus$-UNA of Section 3.1 has already been shown to be succinct. For each of the $\oplus$-FNA, $\oplus$-PNA and $\oplus$-ENA given previously, we now use the method described above to change each M_n into an M'_n that is succinct.

Consider again the $\oplus$-FNA M^f_n of Section 3.2: Let $M^f_n = (Q, \Sigma, \delta, q_1, F)$, with $F = \{q_{n-2}\}$, and δ defined as below:

$$\begin{aligned}
\delta(q_0, a) &= \{\overline{q_0}\} \\
\delta(q_i, a) &= \{q_0, q_i\} \text{ for } 0 < i < n-1 \\
\delta(q_{n-1}, a) &= \begin{cases} \{q_0, q_{n-2}, q_{n-1}\} & \text{for even } n \\ \{q_{n-2}, q_{n-1}\} & \text{for odd } n \end{cases} .
\end{aligned}$$

Based on the method above, we construct M'_{n+1} as

$$M'_{n+1} = (Q, \{a, b\}, \delta', \{q_1\}, \{q_n\})$$

with δ' given by

δ'	a	b
q_0	q_1	$\{\overline{q_0}\}$
q_i	q_{i+1}	$\{q_0, q_i\}$ for $0 < i < n-1$
q_{n-1}	$\{q_0, q_{n-1}\}$	$\begin{cases} \{q_0, q_{n-2}, q_{n-1}\} & \text{for even } n \\ \{q_{n-2}, q_{n-1}\} & \text{for odd } n \end{cases}$
q_n	$\emptyset$	q_n .

We show the DFA equivalent to M'_3 in Figure 12.

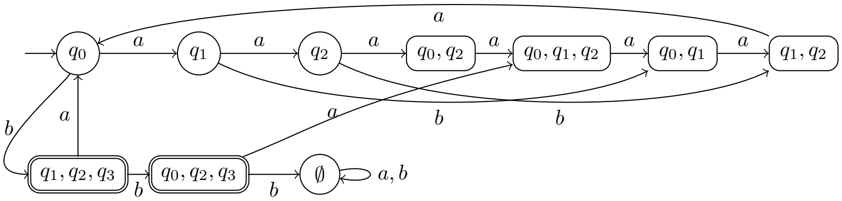


Figure 12: DFA for M'_3.

It is easy to see that the DFA in Figure 12 is minimal, as for any string of the form $a^{n-1}b$ to be accepted, n states are required.

It is trivial to apply the same construction to both the $\oplus$-PNA of Section 3.3 and the $\oplus$-ENA of Section 3.4 to obtain succinctness in those cases.

5 Conclusion

We showed the existence of families of unambiguous, finitely ambiguous, polynomially ambiguous and exponentially ambiguous $\oplus$-NFAs. We also showed succinctness for each of these classes. That is, for each of these classes, there is a family of languages such that the smallest DFA equivalent to the given n-state $\oplus$-NFA, has $O(2^n)$ states. It remains to show families of $\oplus$-NFAs that *strictly* belong to each ambiguity class.

Another issue that we intend to investigate, concerns the differences between NFAs and $\oplus$-NFAs as far as ambiguity is concerned. In particular, we intend to investigate the *structural* ambiguity of $\oplus$-NFAs.

Acknowledgement

This research was supported by NRF grant number 69872.

References

[Chaudhuri et al.(1997)] Chaudhuri, P., Chowdhury, D., Nandi, S., Chattopadhyay, S.: Additive cellular automata: theory and applications; IEEE Computer Society, 1997.

[Domaratzki et al.(2002)] Domaratzki, M., Kisman, D., Shallit, J.: “On the number of distinct languages accepted by finite automata with n states”; Journal of Automata, Languages and Computation; 7 (2002), 4, 479–486.

[Dornhoff and Hohn(1977)] Dornhoff, L., Hohn, F.: Applied Modern Algebra; MacMillan Publishing Co., Inc., 1977.

[Leung(1998)] Leung, H.: “Separating exponentially ambiguous finite automata from polynomially ambiguous finite automata”; SIAM Journal of Computing; 27 (1998), 4, 1073–1082.

[Leung(2005)] Leung, H.: “Descriptional complexity of NFA of different ambiguity”; International Journal of Foundations of Computer Science; 16 (2005), 975–984.

[Ravikumar and Ibarra(1989)] Ravikumar, B., Ibarra, O.: “Relating the type of ambiguity of finite automata to the succinctness of their representation”; SIAM Journal of Computing; 18 (1989), 1263–1282.

[Sipser(1997)] Sipser, M.: Introduction to the Theory of Computation; PWS Publishing Company, 1997.

[Van Zijl(2004)] Van Zijl, L.: “On binary symmetric difference NFAs and succinct descriptions of regular languages”; Theoretical Computer Science; 328 (2004), 1, 161–170.

[Van Zijl(2005)] Van Zijl, L.: “Magic numbers for symmetric difference NFAs”; International Journal of the Foundations of Computer Science; 6 (2005), 5, 1027–1038.

[Vuillemin and Gama(2009)] Vuillemin, J., Gama, N.: “Compact normal form for regular languages as XOR automata”; Proceedings of the 14th International Conference on the Implementation and Application of Automata, Australia; 2009.

[Weber and Seidl(1991)] Weber, A., Seidl, H.: “On the degree of ambiguity of finite automata”; Theoretical Computer Science; 88 (1991), 325–349.