

# Ambiguity in Symmetric Difference NFAs

Lynette van Zijl

Department of Computer Science  
University of Stellenbosch  
South Africa

22 October 2007

- Theoretical automata theory:  $\oplus$ -NFAs
  - Succinctness
  - Minimization and compression
- Practical work: assistive technologies
  - English text to South African Sign Language // sign recognition
  - Assistive software for autism – games for speech therapy, cognitive robotics

- Theoretical automata theory:  $\oplus$ -NFAs
  - Succinctness
  - Minimization and compression
- Practical work: assistive technologies
  - English text to South African Sign Language // sign recognition
  - Assistive software for autism – games for speech therapy, cognitive robotics

- Theoretical automata theory:  $\oplus$ -NFAs
  - Succinctness
  - Minimization and compression
- Practical work: assistive technologies
  - English text to South African Sign Language // sign recognition
  - Assistive software for autism – games for speech therapy, cognitive robotics

- Theoretical automata theory:  $\oplus$ -NFAs
  - Succinctness
  - Minimization and compression
- Practical work: assistive technologies
  - English text to South African Sign Language // sign recognition
  - Assistive software for autism – games for speech therapy, cognitive robotics

- Theoretical automata theory:  $\oplus$ -NFAs
  - Succinctness
  - Minimization and compression
- Practical work: assistive technologies
  - English text to South African Sign Language // sign recognition
  - Assistive software for autism – games for speech therapy, cognitive robotics

- Theoretical automata theory:  $\oplus$ -NFAs
  - Succinctness
  - Minimization and compression
- Practical work: assistive technologies
  - English text to South African Sign Language // sign recognition
  - Assistive software for autism – games for speech therapy, cognitive robotics

- NFAs and  $\oplus$ -NFAs
  - Ambiguity: definition, importance
  - Ambiguity in  $\oplus$ -NFAs

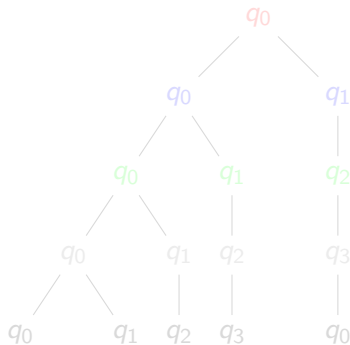
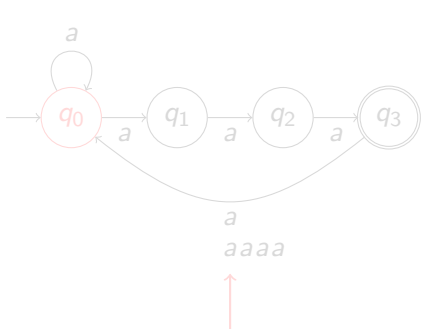


- NFAs and  $\oplus$ -NFAs
- Ambiguity: definition, importance
- Ambiguity in  $\oplus$ -NFAs

- NFAs and  $\oplus$ -NFAs
- Ambiguity: definition, importance
- Ambiguity in  $\oplus$ -NFAs

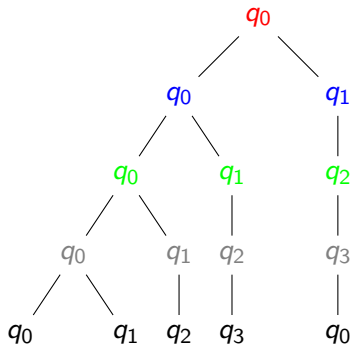
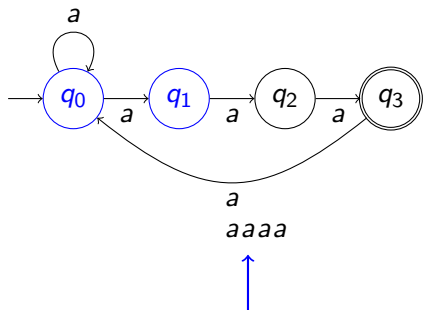
## NFA reminder

An NFA is a finite state machine that, on reading its input string, can nondeterministically decide to which of its possible next states it wants to move. *Accepts language.*



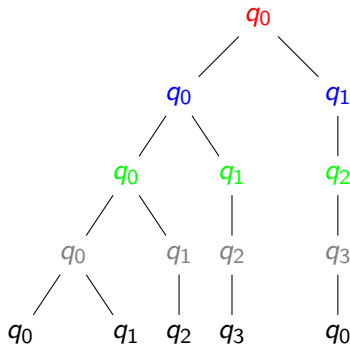
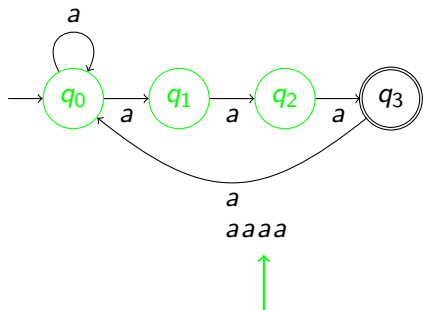
## NFA reminder

An NFA is a finite state machine that, on reading its input string, can nondeterministically decide to which of its possible next states it wants to move. *Accepts language.*



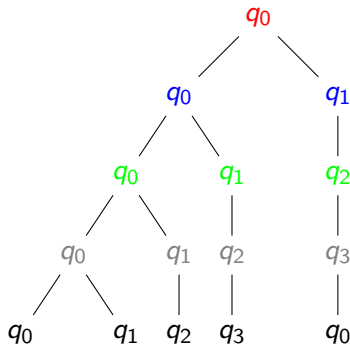
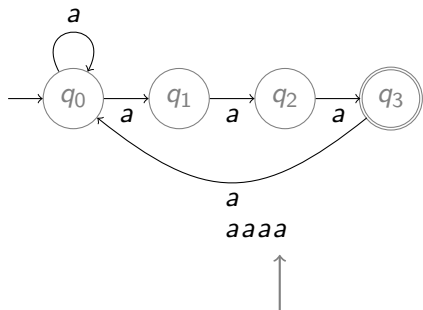
## NFA reminder

An NFA is a finite state machine that, on reading its input string, can nondeterministically decide to which of its possible next states it wants to move. *Accepts language.*



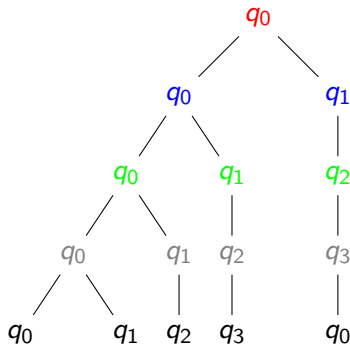
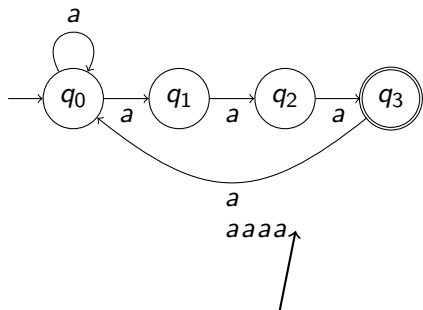
## NFA reminder

An NFA is a finite state machine that, on reading its input string, can nondeterministically decide to which of its possible next states it wants to move. *Accepts language.*



## NFA reminder

An NFA is a finite state machine that, on reading its input string, can nondeterministically decide to which of its possible next states it wants to move. *Accepts language.*



## What is a $\oplus$ -NFA?

Parity machine – in execution tree, takes XOR of possible choices.

## Why $\oplus$ -NFAs?

- Sequencing versus ringlike repetition
- Hardware implementation as LFSR
- Regular languages distribution



## What is a $\oplus$ -NFA?

Parity machine – in execution tree, takes XOR of possible choices.

## Why $\oplus$ -NFAs?

- Sequencing versus ringlike repetition
- Hardware implementation as LFSR
- Regular languages distribution

## What is a $\oplus$ -NFA?

Parity machine – in execution tree, takes XOR of possible choices.

## Why $\oplus$ -NFAs?

- Sequencing versus ringlike repetition
- Hardware implementation as LFSR
- Regular languages distribution

## What is a $\oplus$ -NFA?

Parity machine – in execution tree, takes XOR of possible choices.

## Why $\oplus$ -NFAs?

- Sequencing versus ringlike repetition
- Hardware implementation as LFSR
- Regular languages distribution

## What is ambiguity?

Given an NFA  $M$ , we define the ambiguity of a string  $w$  to be the number of different accepting paths for  $w$  in  $M$ . Function  $amb_M(n)$  is max of ambiguities of strings of length  $n$  or less.

## Types of ambiguity

- unambiguous: ambiguity of any string is zero or one
- finitely ambiguous:  $amb_M(n)$  bounded by constant function
- polynomially ambiguous:  $amb_M(n)$  bounded by polynomial function
- exponentially ambiguous:  $amb_M(n)$  bounded by exponential function

## What is ambiguity?

Given an NFA  $M$ , we define the ambiguity of a string  $w$  to be the number of different accepting paths for  $w$  in  $M$ . Function  $amb_M(n)$  is max of ambiguities of strings of length  $n$  or less.

## Types of ambiguity

- unambiguous: ambiguity of any string is zero or one
- finitely ambiguous:  $amb_M(n)$  bounded by constant function
- polynomially ambiguous:  $amb_M(n)$  bounded by polynomial function
- exponentially ambiguous:  $amb_M(n)$  bounded by exponential function

## What is ambiguity?

Given an NFA  $M$ , we define the ambiguity of a string  $w$  to be the number of different accepting paths for  $w$  in  $M$ . Function  $amb_M(n)$  is max of ambiguities of strings of length  $n$  or less.

## Types of ambiguity

- unambiguous: ambiguity of any string is zero or one
- finitely ambiguous:  $amb_M(n)$  bounded by constant function
- polynomially ambiguous:  $amb_M(n)$  bounded by polynomial function
- exponentially ambiguous:  $amb_M(n)$  bounded by exponential function

## What is ambiguity?

Given an NFA  $M$ , we define the ambiguity of a string  $w$  to be the number of different accepting paths for  $w$  in  $M$ . Function  $amb_M(n)$  is max of ambiguities of strings of length  $n$  or less.

## Types of ambiguity

- unambiguous: ambiguity of any string is zero or one
- finitely ambiguous:  $amb_M(n)$  bounded by constant function
- polynomially ambiguous:  $amb_M(n)$  bounded by polynomial function
- exponentially ambiguous:  $amb_M(n)$  bounded by exponential function

## What is ambiguity?

Given an NFA  $M$ , we define the ambiguity of a string  $w$  to be the number of different accepting paths for  $w$  in  $M$ . Function  $amb_M(n)$  is max of ambiguities of strings of length  $n$  or less.

## Types of ambiguity

- unambiguous: ambiguity of any string is zero or one
- finitely ambiguous:  $amb_M(n)$  bounded by constant function
- polynomially ambiguous:  $amb_M(n)$  bounded by polynomial function
- exponentially ambiguous:  $amb_M(n)$  bounded by exponential function



## Why is ambiguity important?

- Nondeterminism is a **resource** of a finite automaton.
- Does the size of an NFA vary as the amount of ambiguity increases?
- (related to number of nondeterministic moves vs size)

## Why is ambiguity important?

- Nondeterminism is a **resource** of a finite automaton.
- Does the size of an NFA vary as the amount of ambiguity increases?
- (related to number of nondeterministic moves vs size)

## Why is ambiguity important?

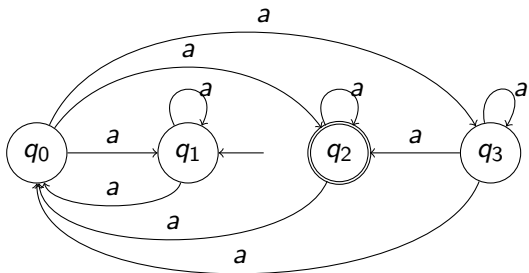
- Nondeterminism is a **resource** of a finite automaton.
- Does the size of an NFA vary as the amount of ambiguity increases?
- (related to number of nondeterministic moves vs size)





# Examples of ambiguity in $\oplus$ -NFAs

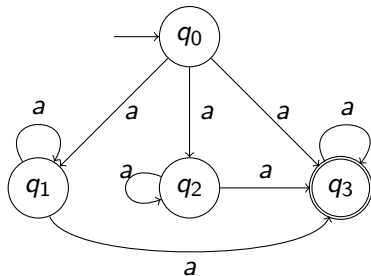
$k$ -ambiguous  $\oplus$ -NFA



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



## Polynomially ambiguous $\oplus$ -NFA

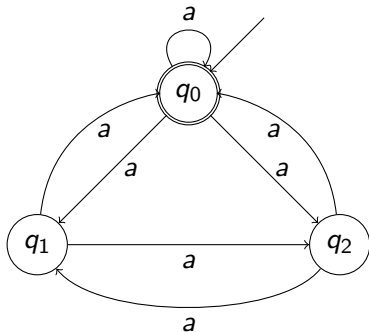


	$a$
$q_0$	$\{q_1, q_2, q_3\}$
$q_1$	$\{q_1, q_3\}$
$q_2$	$\{q_2, q_3\}$
$q_3$	$\{q_3\}$



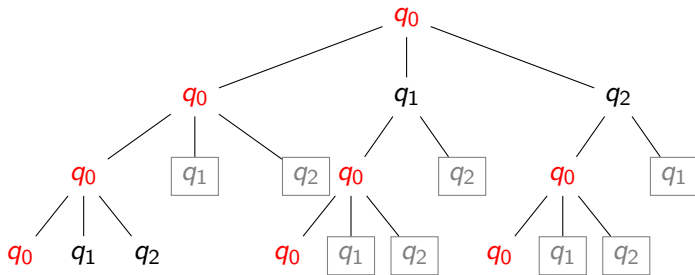


## Exponentially ambiguous $\oplus$ -NFA



	$a$
$q_0$	$\{q_0, q_1, q_2\}$
$q_1$	$\{q_0, q_2\}$
$q_2$	$\{q_0, q_1\}$

# Exponentially ambiguous $\oplus$ -NFA



## What have we achieved?

- Shown patterns to form ambiguous behaviour in structure

## What remains?

- Succinct examples for each ambiguity class
- Families of languages to show relationship between ambiguity classes
- As above, but between traditional NFAs and  $\oplus$ -NFAs

## What have we achieved?

- Shown patterns to form ambiguous behaviour in structure

## What remains?

- Succinct examples for each ambiguity class
- Families of languages to show relationship between ambiguity classes
- As above, but between traditional NFAs and  $\oplus$ -NFAs

## What have we achieved?

- Shown patterns to form ambiguous behaviour in structure

## What remains?

- Succinct examples for each ambiguity class
- Families of languages to show relationship between ambiguity classes
- As above, but between traditional NFAs and  $\oplus$ -NFAs

## What have we achieved?

- Shown patterns to form ambiguous behaviour in structure

## What remains?

- Succinct examples for each ambiguity class
- Families of languages to show relationship between ambiguity classes
- As above, but between traditional NFAs and  $\oplus$ -NFAs

## What have we achieved?

- Shown patterns to form ambiguous behaviour in structure

## What remains?

- Succinct examples for each ambiguity class
- Families of languages to show relationship between ambiguity classes
- As above, but between traditional NFAs and  $\oplus$ -NFAs



Questions?

Contact us

lvzija@gmail.com

<http://www.cs.sun.ac.za/~lynette>