Once-Marking and Always-Marking 1-Limited Automata

Giovanni Pighizzini[1] Luca Prigioniero[2]

[1]Dipartimento di Informatica
Università degli Studi di Milano, Italy

[2]Department of Computer Science
Loughborough University, UK

AFL 2023 – Eger, Hungary
September 6, 2023

UNIVERSITÀ DEGLI STUDI DI MILANO

Loughborough University

Introduction to Limited Automata

Limited Automata [Hibbard '67, *scan limited automata*]

One-tape Turing machines with restricted rewritings

Definition

Fixed an integer $d \geq 1$, a *d-limited automaton* is

- a one-tape Turing machine
- which is allowed to replace the content of each tape cell *only in the first d visits*

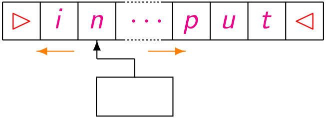
Limited Automata [Hibbard '67, *scan limited automata*]

One-tape Turing machines with restricted rewritings

Definition

Fixed an integer $d \geq 1$, a *d-limited automaton* is

- a one-tape Turing machine
- which is allowed to replace the content of each tape cell *only in the first d visits*



Technical details:

- Input surrounded by two end-markers
- End-markers are never changed
- The head cannot exceed the end-markers

Limited Automata [Hibbard '67, *scan limited automata*]

One-tape Turing machines with restricted rewritings

Definition

Fixed an integer $d \geq 1$, a *d-limited automaton* is

- a one-tape Turing machine
- which is allowed to replace the content of each tape cell *only in the first d visits*

Computational power

- For each $d \geq 2$, d-limited automata characterize context-free languages [Hibbard '67]

Limited Automata [Hibbard '67, *scan limited automata*]

One-tape Turing machines with restricted rewritings

Definition

Fixed an integer $d \geq 1$, a *d-limited automaton* is

- a one-tape Turing machine
- which is allowed to replace the content of each tape cell *only in the first d visits*

Computational power

- For each $d \geq 2$, d-limited automata characterize context-free languages [Hibbard '67]
- 1-limited automata characterize regular languages [Wagner&Wechsung '86]

Simulation of 1-Limited Automata by Finite Automata

Derived from the simulation of 2NFAs by 1DFAs [Shepherdson '59]:

- First visit to a cell: direct simulation
- Further visits: *transition tables*

- Finite control of the simulating automaton: 2^{n^2+n} states
  - *transition table* τ_x — 2^{n^2} possible tables
  - *set* of possible current states — 2^n possible sets

Simulation of 1-Limited Automata by Finite Automata

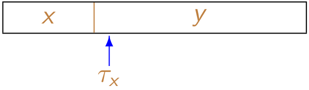
Derived from the simulation of 2NFAs by 1DFAs [Shepherdson '59]:

- First visit to a cell: direct simulation
- Further visits: *transition tables*
- Finite control of the simulating automaton: 2^{n^2+n} states
  - *transition table* τ_x — 2^{n^2} possible tables
  - *set* of possible current states — 2^n possible sets

Simulation of 1-Limited Automata by Finite Automata

Derived from the simulation of 2NFAs by 1DFAs [Shepherdson '59]:

- First visit to a cell: direct simulation
- Further visits: *transition tables*



$\tau_x \subseteq Q \times Q$

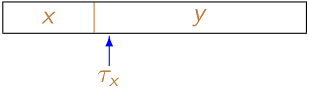
$(p, q) \in \tau_x$ iff 

- Finite control of the simulating automaton: 2^{n^2+n} states
  - *transition table* τ_x — 2^{n^2} possible tables
  - *set* of possible current states — 2^n possible sets

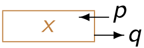
Simulation of 1-Limited Automata by Finite Automata

Derived from the simulation of 2NFAs by 1DFAs [Shepherdson '59]:

- First visit to a cell: direct simulation
- Further visits: *transition tables*



$\tau_x \subseteq Q \times Q$

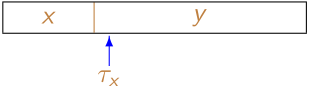
$(p, q) \in \tau_x$ iff 

- Finite control of the simulating automaton: 2^{n^2+n} states
  - *transition table* τ_x — 2^{n^2} possible tables
  - *set* of possible current states — 2^n possible sets

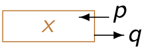
Simulation of 1-Limited Automata by Finite Automata

Derived from the simulation of 2NFAs by 1DFAs [Shepherdson '59]:

- First visit to a cell: direct simulation
- Further visits: *transition tables*



$\tau_x \subseteq Q \times Q$

$(p, q) \in \tau_x$ iff 

- Finite control of the simulating automaton: 2^{n^2+n} states
  - *transition table* τ_x — 2^{n^2} possible tables
  - *set* of possible current states — 2^n possible sets

Simulation of 2NFAs:

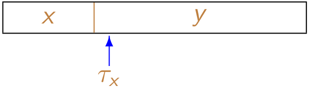
τ_x depends only on x

$\Rightarrow$ The resulting automaton is *deterministic!*

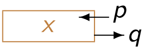
Simulation of 1-Limited Automata by Finite Automata

Derived from the simulation of 2NFAs by 1DFAs [Shepherdson '59]:

- First visit to a cell: direct simulation
- Further visits: *transition tables*



$\tau_x \subseteq Q \times Q$

$(p, q) \in \tau_x$ iff 

- Finite control of the simulating automaton: 2^{n^2+n} states
  - *transition table* τ_x — 2^{n^2} possible tables
  - *set* of possible current states — 2^n possible sets

Simulation of 2NFAs:

τ_x depends only on x

$\Rightarrow$ The resulting automaton is *deterministic!*

Simulation of 1-LAs:

τ_x depends on the choices made while reading x

$\Rightarrow$ The resulting automaton is *nondeterministic!*

Size Costs of Simulations

1-LAs versus Finite Automata [P.&Pisoni '14]

- 1-LA*s* → 1NFA*s*
  exponential
- 1-LA*s* → 1DFA*s*
  double exponential
- det-1-LA*s* → 1DFA*s*
  exponential

Size Costs of Simulations

1-LAs versus Finite Automata [P.&Pisoni '14]

- 1-LA*s* → 1NFA*s*
  exponential
- 1-LA*s* → 1DFA*s*
  double exponential

- det-1-LA*s* → 1DFA*s*
  exponential

Size Costs of Simulations

1-LAs versus Finite Automata [P.&Pisoni '14]

- 1-LAs $\rightarrow$ 1NFAs
  exponential
- 1-LAs $\rightarrow$ 1DFAs
  double exponential
- det-1-LAs $\rightarrow$ 1DFAs
  exponential

Size Costs of Simulations

1-LAs versus Finite Automata [P.&Pisoni '14]

- 1-LA*s* $\rightarrow$ 1NFA*s*
  exponential
- 1-LA*s* $\rightarrow$ 1DFA*s*
  double exponential
- det-1-LA*s* $\rightarrow$ 1DFA*s*
  exponential

All these bounds are tight!

Size Costs of Simulations

1-LAs versus Finite Automata [P.&Pisoni '14]

- 1-LA*s* $\rightarrow$ 1NFA*s*
  exponential
- 1-LA*s* $\rightarrow$ 1DFA*s*
  double exponential
- det-1-LA*s* $\rightarrow$ 1DFA*s*
  exponential

All these bounds are tight!

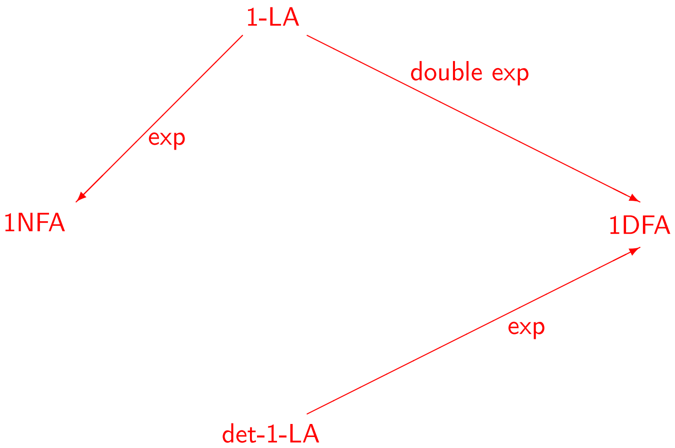
Double role of nondeterminism in 1-LA*s*

On a tape cell:

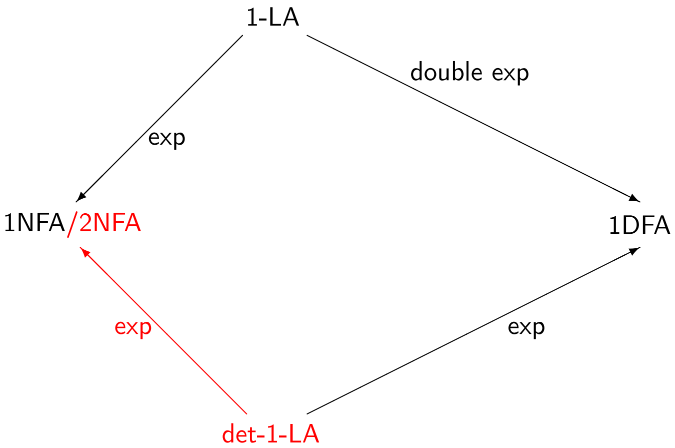
First visit: To replace the content
by a nondeterministically chosen symbol γ

Next visits: To select a transition
the set of available transitions depends on γ!

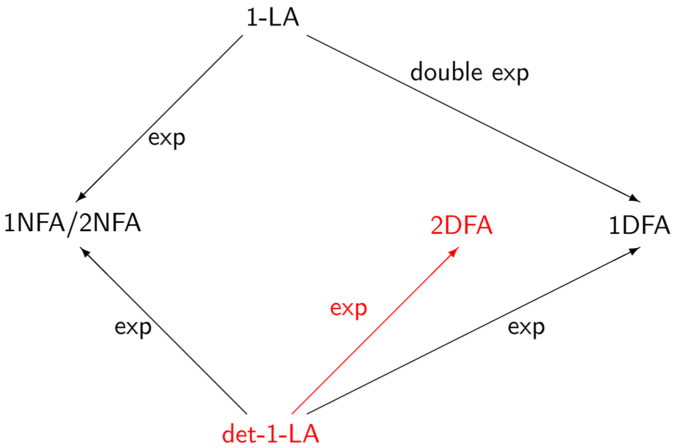
Size Costs of Simulations



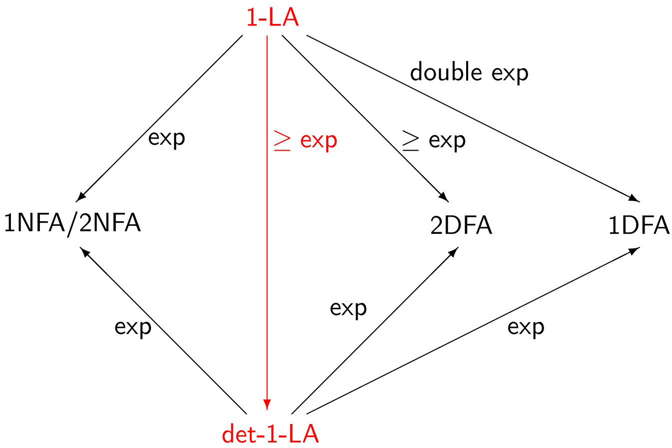
Size Costs of Simulations



Size Costs of Simulations



Size Costs of Simulations



Size Costs of Simulations

1-LAs $\rightarrow$ 1DFAs: double exponential gap

Size Costs of Simulations

1-LAs $\rightarrow$ 1DFAs: double exponential gap

Problem: How much can we restrict the moves of 1-LAs in order to keep this gap double exponential?

Size Costs of Simulations

1-LAs $\rightarrow$ 1DFAs: double exponential gap

Problem: How much can we restrict the moves of 1-LAs in order to keep this gap double exponential?

$\swarrow$

Once-Marking
1-Limited Automata

Size Costs of Simulations

1-LAs $\rightarrow$ 1DFAs: double exponential gap

Problem: How much can we restrict the moves of 1-LAs in order to keep this gap double exponential?

$\swarrow$ $\searrow$

Once-Marking 1-Limited Automata

Always-Marking 1-Limited Automata

Keeping a Double Exponential Gap

Once-Marking 1-Limited Automata

The Language K_n $(n > 0)$

$$K_n = \{x_1\, x_2 \cdots x_k\, x \quad | \quad k > 0,\ x_1, \ldots, x_k, x \in \{\mathtt{a}, \mathtt{b}\}^n, \text{ and } \exists j \in \{1, \ldots, k\} \text{ s.t. } x_j = x\}$$

The Language K_n $(n > 0)$

$$K_n = \{x_1\, x_2 \cdots x_k\, x \quad | \quad k > 0,\ x_1, \ldots, x_k, x \in \{\mathtt{a}, \mathtt{b}\}^n, \text{ and } \exists j \in \{1, \ldots, k\} \text{ s.t. } x_j = x\}$$

Example $(n = 3)$:

a a b a b a b b a a b a b a a b b b b b a

The Language K_n $(n > 0)$

$$K_n = \{x_1\, x_2 \cdots x_k\, x \quad | \quad k > 0,\ x_1, \ldots, x_k, x \in \{\mathtt{a}, \mathtt{b}\}^n, \text{ and } \exists j \in \{1, \ldots, k\} \text{ s.t. } x_j = x\}$$

Example $(n = 3)$:

a a b | a b a | b b a | a b a | b a a | b b b | b b a

A Nondeterministic 1-Limited Automaton for K_n

$\triangleright$ a a b a b a b b a a b a b a a b b b b b a $\triangleleft$ $(n = 3)$

1. Scan all the tape from left to right:
   - check if the input length is a multiple of n
   - mark the rightmost cell of one nondeterministically chosen block
2. Compare symbol by symbol the last block and the one ending with the marked cell
3. Accept if the two blocks are equal

A Nondeterministic 1-Limited Automaton for K_n

$\triangleright$ a a b a b a b b $\dot{\text{a}}$ a b a b a a b b b b b a $\triangleleft$ $(n = 3)$

1. Scan all the tape from left to right:
   - check if the input length is a multiple of n
   - mark the rightmost cell of one nondeterministically chosen block
2. Compare symbol by symbol the last block and the one ending with the marked cell
3. Accept if the two blocks are equal

A Nondeterministic 1-Limited Automaton for K_n

$\triangleright$ a a b a b a b b $\dot{\text{a}}$ a b a b a a b b b b b a $\triangleleft$ $\quad (n = 3)$

1. Scan all the tape from left to right:
   - check if the input length is a multiple of n
   - mark the rightmost cell of one nondeterministically chosen block
2. Compare symbol by symbol the last block and the one ending with the marked cell
3. Accept if the two blocks are equal

A Nondeterministic 1-Limited Automaton for K_n

$\triangleright$ a a b a b a b b ȧ a b a b a a b b b b b a $\triangleleft$ $(n = 3)$

1. Scan all the tape from left to right:
   - check if the input length is a multiple of n
   - mark the rightmost cell of one nondeterministically chosen block
2. Compare symbol by symbol the last block and the one ending with the marked cell
3. Accept if the two blocks are equal

A Nondeterministic 1-Limited Automaton for K_n

$\triangleright$ a a b a b a b b $\dot{\text{a}}$ a b a b a a b b b b b a $\triangleleft$ $\qquad (n = 3)$

1. Scan all the tape from left to right:
   - check if the input length is a multiple of n
   - mark the rightmost cell of one nondeterministically chosen block
2. Compare symbol by symbol the last block and the one ending with the marked cell
3. Accept if the two blocks are equal

Complexity:

- $O(n)$ states
- Fixed working alphabet

$\Rightarrow$ 1-LA of size $O(n)$

A Nondeterministic 1-Limited Automaton for K_n

▷ a a b a b a b b ȧ a b a b a a b b b b b a ◁ $(n = 3)$

1. Scan all the tape from left to right:
   - check if the input length is a multiple of n
   - mark the rightmost cell of one nondeterministically chosen block
2. Compare symbol by symbol the last block and the one ending with the marked cell
3. Accept if the two blocks are equal

Complexity:

- $O(n)$ states
- Fixed working alphabet

$\Rightarrow$ 1-LA of size $O(n)$

Rewritings: to accept K_n it is enough to mark one tape cell during the first visit!

Recognizing K_n with Finite Automata

$$K_n = \{x_1\, x_2 \cdots x_k\, x \quad | \quad k > 0,\ x_1, \ldots, x_k, x \in \{\mathtt{a}, \mathtt{b}\}^n, \\ \text{and } \exists j \in \{1, \ldots, k\} \text{ s.t. } x_j = x\}$$

Recognizing K_n with Finite Automata

$$K_n = \{x_1\, x_2 \cdots x_k\, x \quad | \quad k > 0,\ x_1, \ldots, x_k, x \in \{\mathtt{a}, \mathtt{b}\}^n, \\ \text{and } \exists j \in \{1, \ldots, k\} \text{ s.t. } x_j = x\}$$

Finite automata

To recognize K_n each 1DFA requires a number of states at least *double exponential* in n

Proof: standard distinguishability arguments

Recognizing K_n with Finite Automata

$$K_n = \{x_1\, x_2 \cdots x_k\, x \quad | \quad k > 0,\ x_1, \ldots, x_k, x \in \{\mathtt{a}, \mathtt{b}\}^n, \\ \text{and } \exists j \in \{1, \ldots, k\} \text{ s.t. } x_j = x\}$$

Finite automata

To recognize K_n each 1DFA requires a number of states at least *double exponential* in n

Proof: standard distinguishability arguments

1-LA*s* $\rightarrow$ 1DFA*s*

The gap remains double exponential even for 1-LAs that are allowed to rewrite only one cell!

Recognizing K_n with Finite Automata

$$K_n = \{x_1\, x_2 \cdots x_k\, x \;\mid\; k > 0,\ x_1, \ldots, x_k, x \in \{\mathtt{a}, \mathtt{b}\}^n, \text{ and } \exists j \in \{1, \ldots, k\} \text{ s.t. } x_j = x\}$$

Finite automata

To recognize K_n each 1DFA requires a number of states at least *double exponential* in n

Proof: standard distinguishability arguments

1-LAs $\rightarrow$ 1DFAs

The gap remains double exponential even for 1-LAs that are allowed to rewrite only one cell!

$\Rightarrow$ Once-Marking 1-Limited Automata

Once-Marking 1-Limited Automata

Definition

A 1-limited automaton is said to be *once marking* (OM-1-LA) if

- in each computation there is a unique tape cell whose input symbol σ is replaced with its marked version $\mathring{\sigma}$
- all the remaining cells are never changed

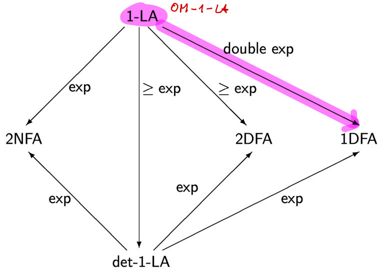
Once-Marking 1-Limited Automata

Definition

A 1-limited automaton is said to be *once marking* (OM-1-LA) if

- in each computation there is a unique tape cell whose input symbol σ is replaced with its marked version $\dot{\sigma}$
- all the remaining cells are never changed

- Computational power: Regular languages
- Costs of the conversion to 1DFAs: Double exponential



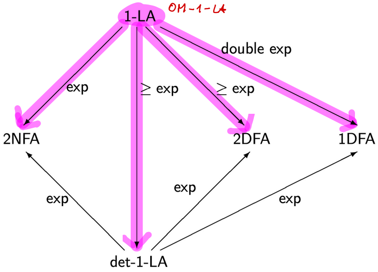
Once-Marking 1-Limited Automata

Definition

A 1-limited automaton is said to be *once marking* (OM-1-LA) if

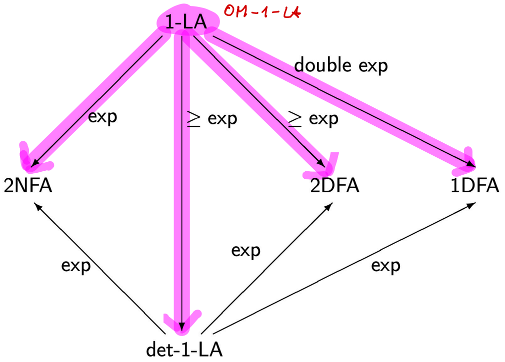
- in each computation there is a unique tape cell whose input symbol σ is replaced with its marked version $\dot{\sigma}$
- all the remaining cells are never changed

- Computational power: Regular languages
- Costs of the conversion to 1DFAs: Double exponential



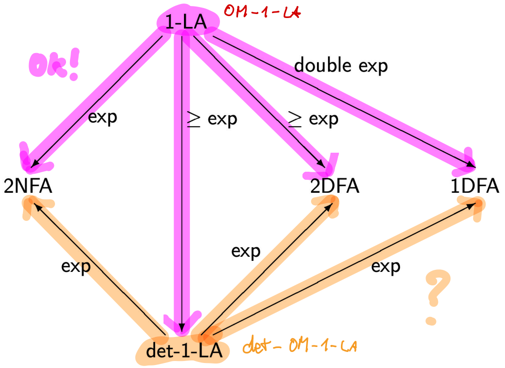
Conversions from Once-Marking 1-Limited Automata

Costs from *nondeterministic* OM-1-LAs:



Conversions from Once-Marking 1-Limited Automata

Costs from *nondeterministic* OM-1-LAs:



Costs from *deterministic* OM-1-LAs?

Conversions of *deterministic* OM-1-LAs

Theorem

For each n-state deterministic OM-1-LA $\mathcal{A}$ *there exists an equivalent* 2DFA $\mathcal{A}'$ *with* $O(n^3)$ *states.*

Conversions of *deterministic* OM-1-LAs

Theorem

For each n-state deterministic OM-1-LA $\mathcal{A}$ *there exists an equivalent* 2DFA $\mathcal{A}'$ *with* $O(n^3)$ *states.*

Proof idea

- At the beggining $\mathcal{A}'$ makes the same moves as $\mathcal{A}$
- When the marking move is reached:
  - $\mathcal{A}'$ simulates it without marking
  - $\mathcal{A}'$ saves the state q and the symbol σ in its control
- Remaining moves:
  - if symbol on the tape $\neq \sigma$: simulated as in $\mathcal{A}$
  - otherwise $\mathcal{A}'$ calls a *verification procedure* to decide if the symbol is the marked one
  - According to the result $\mathcal{A}'$ choose the move
- *Verification procedure*:
  - "backward search" in the computation tree [Sipser '80]

Conversions of *deterministic* OM-1-LAs

Theorem

For each n-state deterministic OM-1-LA $\mathcal{A}$ *there exists an equivalent* 2DFA $\mathcal{A}'$ *with* $O(n^3)$ *states.*

Proof idea

- At the beggining $\mathcal{A}'$ makes the same moves as $\mathcal{A}$
- When the marking move is reached:
  - $\mathcal{A}'$ simulates it without marking
  - $\mathcal{A}'$ saves the state q and the symbol σ in its control
- Remaining moves:
  - if symbol on the tape $\neq \sigma$: simulated as in $\mathcal{A}$
  - otherwise $\mathcal{A}'$ calls a *verification procedure* to decide if the symbol is the marked one
  - According to the result $\mathcal{A}'$ choose the move
- *Verification procedure*:
  - "backward search" in the computation tree [Sipser '80]

Conversions of *deterministic* OM-1-LAs

Theorem

For each n-state deterministic OM-1-LA $\mathcal{A}$ *there exists an equivalent* 2DFA $\mathcal{A}'$ *with* $O(n^3)$ *states.*

Proof idea

- At the beggining $\mathcal{A}'$ makes the same moves as $\mathcal{A}$
- When the marking move is reached:
  - $\mathcal{A}'$ simulates it without marking
    $\mathcal{A}'$ saves the state q and the symbol σ in its control
- Remaining moves:
  - if symbol on the tape $\neq \sigma$: simulated as in $\mathcal{A}$
  - otherwise $\mathcal{A}'$ calls a *verification procedure* to decide if the symbol is the marked one
  - According to the result $\mathcal{A}'$ choose the move
- *Verification procedure*:
  - "backward search" in the computation tree [Sipser '80]

Conversions of *deterministic* OM-1-LAs

Theorem

For each n-state deterministic OM-1-LA $\mathcal{A}$ *there exists an equivalent* 2DFA $\mathcal{A}'$ *with* $O(n^3)$ *states.*

Proof idea

- At the beggining $\mathcal{A}'$ makes the same moves as $\mathcal{A}$
- When the marking move is reached:
  - $\mathcal{A}'$ simulates it without marking
    $\mathcal{A}'$ saves the state q and the symbol σ in its control
- Remaining moves:
  - if symbol on the tape $\neq \sigma$: simulated as in $\mathcal{A}$
  - otherwise $\mathcal{A}'$ calls a *verification procedure* to decide if the symbol is the marked one
  - According to the result $\mathcal{A}'$ choose the move
- *Verification procedure*:
  - "backward search" in the computation tree [Sipser '80]

Conversions of *deterministic* OM-1-LAs

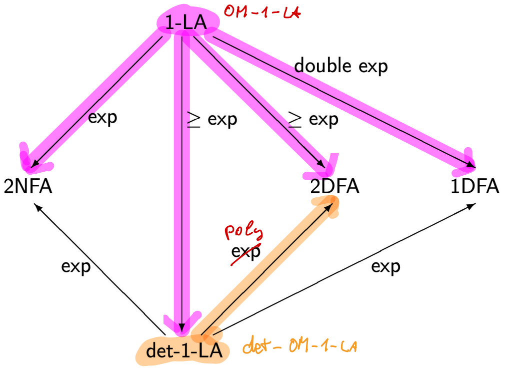
Theorem

For each n-state deterministic OM-1-LA $\mathcal{A}$ *there exists an equivalent* 2DFA $\mathcal{A}'$ *with* $O(n^3)$ *states.*

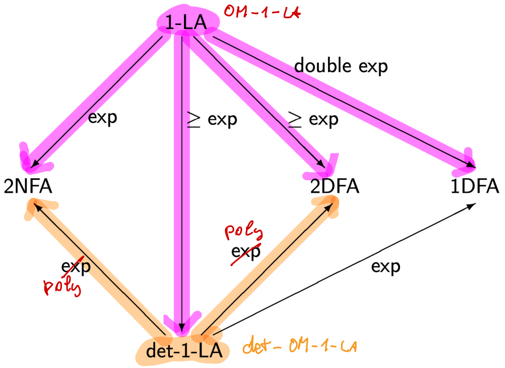
Proof idea

- At the beggining $\mathcal{A}'$ makes the same moves as $\mathcal{A}$
- When the marking move is reached:
  - $\mathcal{A}'$ simulates it without marking
    $\mathcal{A}'$ saves the state q and the symbol σ in its control
- Remaining moves:
  - if symbol on the tape $\neq \sigma$: simulated as in $\mathcal{A}$
  - otherwise $\mathcal{A}'$ calls a *verification procedure* to decide if the symbol is the marked one
  - According to the result $\mathcal{A}'$ choose the move
- *Verification procedure*:
  - "backward search" in the computation tree [Sipser '80]

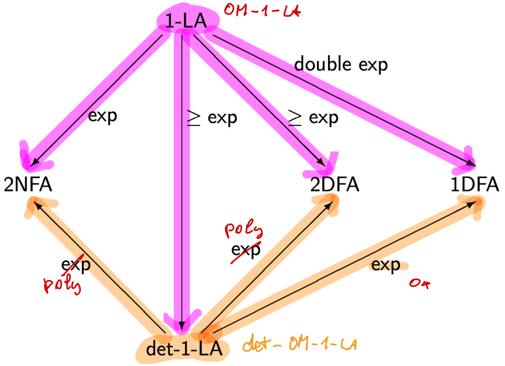
Size Costs of the Conversion From OM-1-LAs



Size Costs of the Conversion From OM-1-LAs



Size Costs of the Conversion From OM-1-LAs



Reducing the Gap to a Single Exponential

Always-Marking 1-Limited Automata

Reducing the Gap: First Attempts

Double role of nondeterminism in 1-LAs

On a tape cell:

First visit: To replace the content
by a nondeterministically chosen symbol γ

Next visits: To select a transition
the set of available transitions depends on γ!

Reducing the Gap: First Attempts

Double role of nondeterminism in 1-LAs

On a tape cell:

First visit: To replace the content
by a nondeterministically chosen symbol γ

Next visits: To select a transition
the set of available transitions depends on γ!

Two "naif" restrictions of 1-LAs:

1. *Deterministic choice* for rewritings
   Nondeterministic choice for next state and head movement
2. *Nondeterministic choice* for rewritings
   Deterministic choice for next state and head movement

Reducing the Gap: First Attempts

Double role of nondeterminism in 1-LAs

On a tape cell:

First visit: To replace the content
by a nondeterministically chosen symbol γ

Next visits: To select a transition
the set of available transitions depends on γ!

Two "naif" restrictions of 1-LAs:

1. *Deterministic choice* for rewritings
   Nondeterministic choice for next state and head movement
2. *Nondeterministic choice* for rewritings
   Deterministic choice for next state and head movement

Reducing the Gap: First Attempts

Double role of nondeterminism in 1-LA*s*

On a tape cell:

First visit: To replace the content
by a nondeterministically chosen symbol γ

Next visits: To select a transition
the set of available transitions depends on γ!

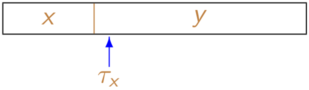
Two "naif" restrictions of 1-LAs:

1. *Deterministic choice* for rewritings
   Nondeterministic choice for next state and head movement
2. *Nondeterministic choice* for rewritings
   Deterministic choice for next state and head movement

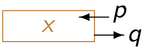
For both restrictions, in the worst case the size gap to 1DFAs remains double exponential!

Simulation of 1-Limited Automata by Finite Automata

- First visit to a cell: direct simulation
- Further visits: *transition tables*



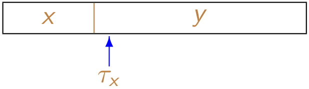
$\tau_x \subseteq Q \times Q$

$(p, q) \in \tau_x$ iff 

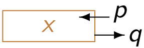
- Finite control of the simulating automaton: 2^{n^2+n} states
  - *transition table* τ_x — 2^{n^2} possible tables
  - *set* of possible current states — 2^n possible sets

Simulation of 1-Limited Automata by Finite Automata

- First visit to a cell: direct simulation
- Further visits: *transition tables*



$\tau_x \subseteq Q \times Q$

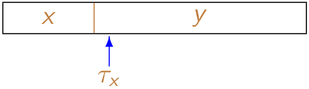
$(p, q) \in \tau_x$ iff 

- Finite control of the simulating automaton: 2^{n^2+n} states
  - *transition table* τ_x — 2^{n^2} possible tables
  - *set* of possible current states — 2^n possible sets

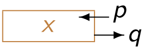
The double exponential gap is due to the fact that different computations can produce different τ_x for the same prefix x

Simulation of 1-Limited Automata by Finite Automata

- First visit to a cell: direct simulation
- Further visits: *transition tables*



$\tau_x \subseteq Q \times Q$

$(p, q) \in \tau_x$ iff 

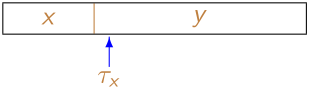
- Finite control of the simulating automaton: 2^{n^2+n} states
  - *transition table* τ_x — 2^{n^2} possible tables
  - *set* of possible current states — 2^n possible sets

The double exponential gap is due to the fact that different computations can produce different τ_x for the same prefix x

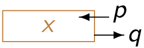
Consider restrictions that avoid that!

Simulation of 1-Limited Automata by Finite Automata

- First visit to a cell: direct simulation
- Further visits: *transition tables*



$\tau_x \subseteq Q \times Q$

$(p, q) \in \tau_x$ iff 

- Finite control of the simulating automaton: 2^{n^2+n} states
  - *transition table* τ_x — 2^{n^2} possible tables
  - *set* of possible current states — 2^n possible sets

The double exponential gap is due to the fact that different computations can produce different τ_x for the same prefix x

Consider restrictions that avoid that!

$\Rightarrow$ Always-Marking 1-Limited Automata

Always-Marking 1-Limited Automata

Definition

A 1-LA is said to be *always marking* if

each time the head visits a tape cell for the first time,

the input symbol σ in it is replaced with its marked version $\dot{\sigma}$

Always-Marking 1-Limited Automata

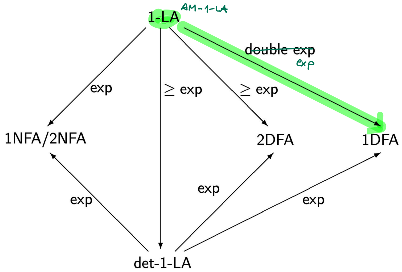
Definition

A 1-LA is said to be *always marking* if

each time the head visits a tape cell for the first time,

the input symbol σ in it is replaced with its marked version $\dot{\sigma}$

- Computational power: Regular languages
- Costs of the conversion to 1DFAs: *Single* exponential



Always-Marking 1-Limited Automata

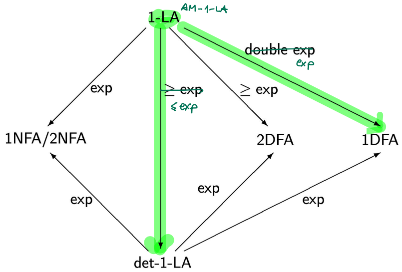
Definition

A 1-LA is said to be *always marking* if

each time the head visits a tape cell for the first time,

the input symbol σ in it is replaced with its marked version $\dot{\sigma}$

- Computational power: Regular languages
- Costs of the conversion to 1DFAs: *Single* exponential



The Language J_n $(n > 0)$

$$J_n = \{x\, x_1\, x_2 \cdots x_k \quad | \quad k > 0,\ x_1, \ldots, x_k, x \in \{\mathtt{a}, \mathtt{b}\}^n, \text{ and } \exists j \in \{1, \ldots, k\} \text{ s.t. } x_j = x\}$$

The Language J_n $(n > 0)$

$$J_n = \{x\, x_1\, x_2 \cdots x_k \quad | \quad k > 0,\ x_1, \ldots, x_k, x \in \{\mathtt{a}, \mathtt{b}\}^n, \text{ and } \exists j \in \{1, \ldots, k\} \text{ s.t. } x_j = x\}$$

Then:

- $J_n = (K_n)^R$
- Each 2NFA accepting J_n has a number of states at least exponential in n
  - 2NFAs $\to$ 1DFAs costs exponential
  - K_n needs at least 2^{2^n} states to be accepted by 1DFAs
  - J_n and K_n have 2NFAs of the same size

The Language J_n $(n > 0)$

$$J_n = \{x\, x_1\, x_2 \cdots x_k \quad | \quad k > 0,\ x_1, \ldots, x_k, x \in \{\mathtt{a}, \mathtt{b}\}^n, \\ \text{and } \exists j \in \{1, \ldots, k\} \text{ s.t. } x_j = x\}$$

Then:

- $J_n = (K_n)^R$
- Each 2NFA accepting J_n has a number of states at least exponential in n
  - 2NFAs $\rightarrow$ 1DFAs costs exponential
  - K_n needs at least 2^{2^n} states to be accepted by 1DFAs
  - J_n and K_n have 2NFAs of the same size

Recognizing J_n with AM-1-LAs

▷ a b a a b b a b b a b a b a a b b b b b a ◁ $(n = 3)$

- Visit and mark the first n tape cells

  Then inspect the following blocks as follows:

- When the head reaches a cell for the first time:
  - Mark it and locate the corresponding cell in the first block
  - If *the symbols in the two cells do not match*

    then skip the remaining symbols of the current block

    otherwise if *the current block is not finished*

    then continue inspection

    otherwise *accepts* if the length of the remaining part of the input is a multiple of n

Recognizing J_n with AM-1-LAs

$\triangleright$ $\dot{\text{a}}\,\dot{\text{b}}\,\dot{\text{a}}$ a b b a b b a b a b a a b b b b b a $\triangleleft$ $(n = 3)$

- Visit and mark the first n tape cells

  Then inspect the following blocks as follows:
- When the head reaches a cell for the first time:
  - Mark it and locate the corresponding cell in the first block
  - If *the symbols in the two cells do not match*

    then skip the remaining symbols of the current block

    otherwise if *the current block is not finished*

    then continue inspection

    otherwise *accepts* if the length of the remaining part of the input is a multiple of n

Recognizing J_n with AM-1-LAs

$\triangleright\ \dot{\texttt{a}}\ \dot{\texttt{b}}\ \dot{\texttt{a}}\ \texttt{a b b a b b a b a b a a b b b b b a}\ \triangleleft$ $\qquad (n = 3)$

- Visit and mark the first n tape cells

  Then inspect the following blocks as follows:
- When the head reaches a cell for the first time:
  - Mark it and locate the corresponding cell in the first block
  - If *the symbols in the two cells do not match*

    then skip the remaining symbols of the current block

    otherwise if *the current block is not finished*

    then continue inspection

    otherwise *accepts* if the length of the remaining part of the input is a multiple of n

Recognizing J_n with AM-1-LAs

$\triangleright\ \dot{a}\ \dot{b}\ \dot{a}\ \dot{a}\ \dot{b}\ \dot{b}\ \dot{a}\ b\ b\ a\ b\ a\ b\ a\ a\ b\ b\ b\ b\ b\ a\ \triangleleft$ $\qquad (n = 3)$

- Visit and mark the first n tape cells

  Then inspect the following blocks as follows:
- When the head reaches a cell for the first time:
  - Mark it and locate the corresponding cell in the first block
  - If *the symbols in the two cells do not match*

    then skip the remaining symbols of the current block

    otherwise if *the current block is not finished*

    then continue inspection

    otherwise *accepts* if the length of the remaining part of the input is a multiple of n

Recognizing J_n with AM-1-LAs

$\triangleright\ \dot{a}\ \dot{b}\ \dot{a}\ \dot{a}\ \dot{b}\ \dot{b}\ \dot{a}\ \dot{b}\ b\ a\ b\ a\ b\ a\ a\ b\ b\ b\ b\ b\ a\ \triangleleft$ $(n = 3)$

- Visit and mark the first n tape cells

  Then inspect the following blocks as follows:
- When the head reaches a cell for the first time:
  - Mark it and locate the corresponding cell in the first block
  - If *the symbols in the two cells do not match*
    then skip the remaining symbols of the current block
    otherwise if *the current block is not finished*
    then continue inspection
    otherwise *accepts* if the length of the remaining part of the input is a multiple of n

Recognizing J_n with AM-1-LAs

$\triangleright$ $\dot{a}\,\dot{b}\,\dot{a}\,\dot{a}\,\dot{b}\,\dot{b}\,\dot{a}\,\dot{b}$ b a b a b a a b b b b b a $\triangleleft$ $(n = 3)$

- Visit and mark the first n tape cells

  Then inspect the following blocks as follows:

- When the head reaches a cell for the first time:
  - Mark it and locate the corresponding cell in the first block
  - If *the symbols in the two cells do not match*

    then skip the remaining symbols of the current block

    otherwise if *the current block is not finished*

    then continue inspection

    otherwise *accepts* if the length of the remaining part of the input is a multiple of n

Recognizing J_n with AM-1-LAs

$\triangleright$ a b a a b b a b b a b a b a a b b b b b a $\triangleleft$ $(n = 3)$

- Visit and mark the first n tape cells

  Then inspect the following blocks as follows:
- When the head reaches a cell for the first time:
  - Mark it and locate the corresponding cell in the first block
  - If *the symbols in the two cells do not match*

    then skip the remaining symbols of the current block

    otherwise if *the current block is not finished*

    then continue inspection

    otherwise *accepts* if the length of the remaining part of the input is a multiple of n

Recognizing J_n with AM-1-LAs

$\triangleright$ a b a a b b a b b a b a b a a b b b b b a $\triangleleft$ $(n = 3)$

- Visit and mark the first n tape cells

  Then inspect the following blocks as follows:

- When the head reaches a cell for the first time:
  - Mark it and locate the corresponding cell in the first block
  - If *the symbols in the two cells do not match*

    then skip the remaining symbols of the current block

    otherwise if *the current block is not finished*

    then continue inspection

    otherwise *accepts* if the length of the remaining part of the input is a multiple of n

Recognizing J_n with AM-1-LAs

$\triangleright$ a b a a b b a b b a b a b a a b b b b b a $\triangleleft$ $(n = 3)$

- Visit and mark the first n tape cells

  Then inspect the following blocks as follows:
- When the head reaches a cell for the first time:
  - Mark it and locate the corresponding cell in the first block
  - If *the symbols in the two cells do not match*
    then skip the remaining symbols of the current block
    otherwise if *the current block is not finished*
    then continue inspection
    otherwise *accepts* if the length of the remaining part of the input is a multiple of n

Implementation with $O(n)$ states

Recognizing J_n with AM-1-LAs

$\triangleright$ $\dot{a}\ \dot{b}\ \dot{a}\ \dot{a}\ \dot{b}\ \dot{b}\ \dot{a}\ \dot{b}$ b a b a b a a b b b b b a $\triangleleft$ $(n = 3)$

- Visit and mark the first n tape cells

  Then inspect the following blocks as follows:

- When the head reaches a cell for the first time:
  - Mark it and locate the corresponding cell in the first block
  - If *the symbols in the two cells do not match*

    then skip the remaining symbols of the current block

    otherwise if *the current block is not finished*

    then continue inspection

    otherwise *accepts* if the length of the remaining part of the input is a multiple of n

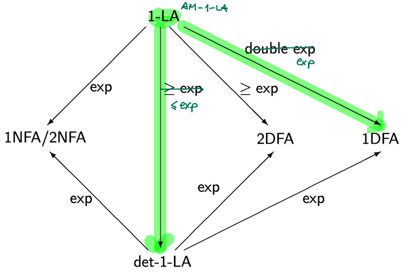
Implementation with $O(n)$ states

Only deterministic transitions!

Simulations of AM-1-LAs

Summing up:

- det-1-LAs → 2NFAs costs exponential
- J_n is accepted by a det-AM-1-LA with $O(n)$ states
- Each 2NFA accepting J_n has a number of states at least exponential in n



Simulations of AM-1-LAs

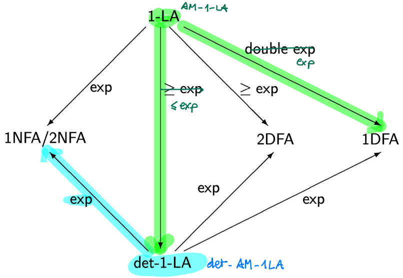
Summing up:

- det-1-LAs → 2NFAs costs exponential
- J_n is accepted by a det-AM-1-LA with $O(n)$ states
- Each 2NFA accepting J_n has a number of states at least exponential in n

Then:

det-AM-1-LAs → 2NFAs

Exponential cost!



Simulations of AM-1-LAs

Summing up:

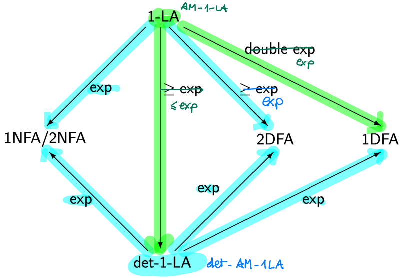
- det-1-LAs $\to$ 2NFAs costs exponential
- J_n is accepted by a det-AM-1-LA with $O(n)$ states
- Each 2NFA accepting J_n has a number of states at least exponential in n

Then:

det-AM-1-LAs $\to$ 2NFAs

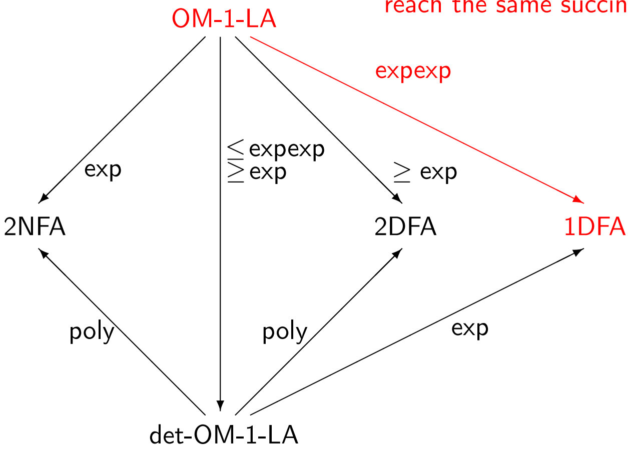
Exponential cost!

$\Rightarrow$ Even the costs of the remaining simulations are exponential

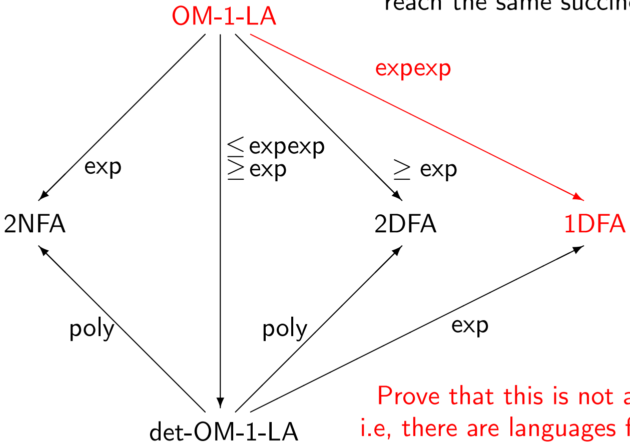


Conclusion: Once-Marking 1-Limited Automata

“One drop” of nondeterminism allows to reach the same succinctness as 1-LAs (language K_n)



Conclusion: Once-Marking 1-Limited Automata



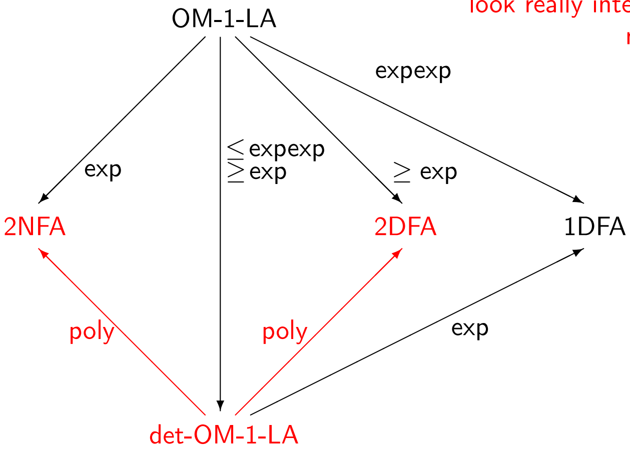
"One drop" of nondeterminism allows to reach the same succinctness as 1-LAs (language K_n)

Problem

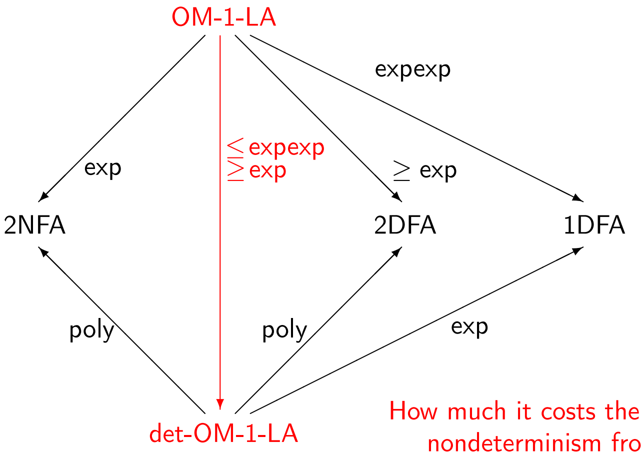
Prove that this is not always the case, i.e, there are languages for which 1-LAs are more succinct than OM-1-LAs (e.g., variants of K_n)

Conclusion: Once-Marking 1-Limited Automata

The capability of once marking does not look really interesting without nondeterminism



Conclusion: Once-Marking 1-Limited Automata

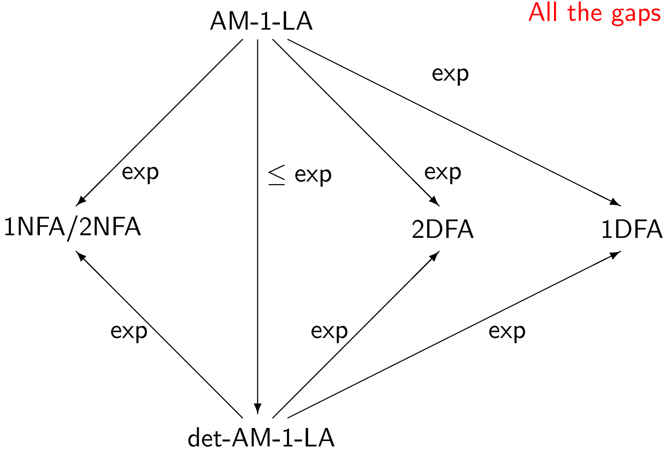


Problem

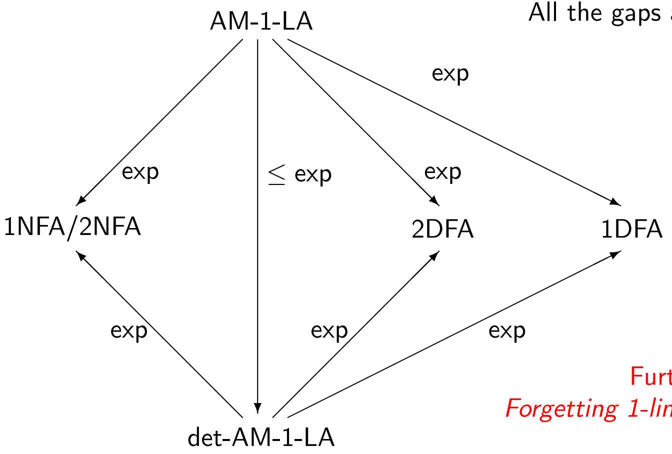
How much it costs the elimination of nondeterminism from OM-1-LAs?

Conclusion: Always-Marking 1-Limited Automata

All the gaps are exponential



Conclusion: Always-Marking 1-Limited Automata



All the gaps are exponential

Further restriction:
Forgetting 1-limited automata
(NCMA 2023)

Further possible investigation lines

- Unary case
- Connections with the Sakoda and Sisper question (costs of 2NFA$\rightarrow$ 2DFA and 1NFA$\rightarrow$ 2DFA)
- ...

Further possible investigation lines

- Unary case
- Connections with the Sakoda and Sisper question (costs of 2NFA$\rightarrow$ 2DFA and 1NFA$\rightarrow$ 2DFA)
- ...

Thank you for your attention!