On Simulation Costs of Unary Limited Automata

Martin Kutrib   Matthias Wendlandt

Institut für Informatik, Universität Giessen
Arndtstr. 2, 35392 Giessen, Germany
{kutrib,matthias.wendlandt}@informatik.uni-giessen.de

Limited Automata

- A k-limited automaton is a linear bounded automaton that may rewrite each tape square only in the first k visits, where k is a fixed constant.
- Afterwards the squares can still be visited any number of times, but without rewriting their contents.

Former Results

- Hennie machines are linear bounded automata that are only allowed to visit any tape square a constant number of times. The accepted language is regular (Hennie, 1965).
- It is shown that even linear-time computations cannot accept non-regular languages (Hennie, 1965).
- This result has been improved to $O(n \log n)$ time in (Hartmanis, 1968).

Former Results – Limited Automata

- Limited automata have been firstly studied in (Hibbard, 1967).
- It is shown that the nondeterministic variant characterizes the context-free languages provided $k \geq 2$.
- For the deterministic variant it has been shown that if $k = 2$, then the accepted family of languages is equal to the deterministic context-free languages (Pighizzini and Pisoni, 2013).
- There is a tight and strict hierarchy of language classes depending on k for the deterministic variant (Hibbard, 1967).
- One-limited automata, deterministic and nondeterministic, can accept only regular languages.

Former Results – Descriptional Complexity

Among other results, Pighizzini and Pisoni (2013) showed:

- The trade-off between 1-LA and DFA is $2^{n \cdot 2^{n^2}}$.
- The trade-off between 1-DLA and DFA is $n \cdot (n+1)^n$.
- These results imply an exponential trade-off between 1-LA and 1-DLA.

Limited Automata – Definition

A *deterministic* k*-limited automaton* (k-DLA, for short) is a system $M = \langle S, \Sigma, \Gamma, \delta, \triangleright, \triangleleft, s_0, F \rangle$.

S is the finite, nonempty set of internal states

Limited Automata – Definition

A *deterministic k-limited automaton* (k-DLA, for short) is a system $M = \langle S, \Sigma, \Gamma, \delta, \triangleright, \triangleleft, s_0, F \rangle$.

Σ is the finite set of input symbols

Limited Automata – Definition

A *deterministic k-limited automaton* (k-DLA, for short) is a system $M = \langle S, \Sigma, \Gamma, \delta, \triangleright, \triangleleft, s_0, F \rangle$.

Γ is the finite set of tape symbols partitioned into $\Gamma_k \cup \Gamma_{k-1} \cup \cdots \cup \Gamma_0$ where $\Gamma_0 = \Sigma$

Limited Automata – Definition

A *deterministic k-limited automaton* (k-DLA, for short) is a system $M = \langle S, \Sigma, \Gamma, \delta, \rhd, \lhd, s_0, F \rangle$.

$\rhd \notin \Gamma$ is the left and $\lhd \notin \Gamma$ is the right endmarker

Limited Automata – Definition

A *deterministic k-limited automaton* (k-DLA, for short) is a system $M = \langle S, \Sigma, \Gamma, \delta, \rhd, \lhd, s_0, F \rangle$.

$s_0 \in S$ is the initial state

Limited Automata – Definition

A *deterministic k-limited automaton* (k-DLA, for short) is a system $M = \langle S, \Sigma, \Gamma, \delta, \rhd, \lhd, s_0, F \rangle$.

$F \subseteq S$ is the set of accepting states

Limited Automata – Definition

A *deterministic k-limited automaton* (k-DLA, for short) is a system $M = \langle S, \Sigma, \Gamma, \delta, \rhd, \lhd, s_0, F \rangle$.

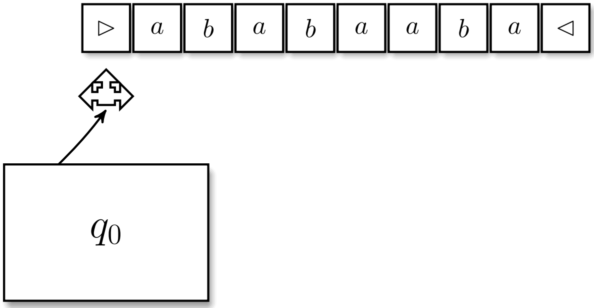
$\delta : S \times (\Gamma \cup \{\rhd, \lhd\}) \to S \times (\Gamma \cup \{\rhd, \lhd\}) \times \{-1, 1\}$

Limited Automata – Definition

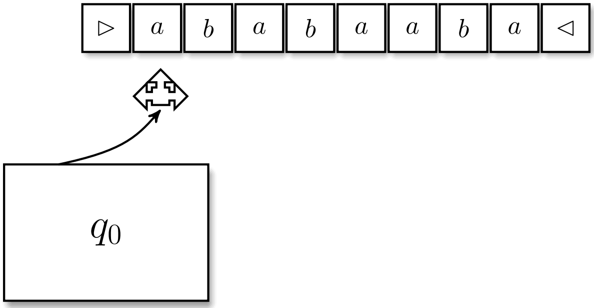
For each $(s', y, d) = \delta(s, x)$ with $x \in \Gamma_i$,

- if $i = k$ then $x = y$,
- if $i < k$ and $d = 1$ then $y \in \Gamma_j$ with $j = \min\{\lceil \frac{i}{2} \rceil \cdot 2 + 1, k\}$, and
- if $i < k$ and $d = -1$ then $y \in \Gamma_j$ with $j = \min\{\lceil \frac{i+1}{2} \rceil \cdot 2, k\}$.

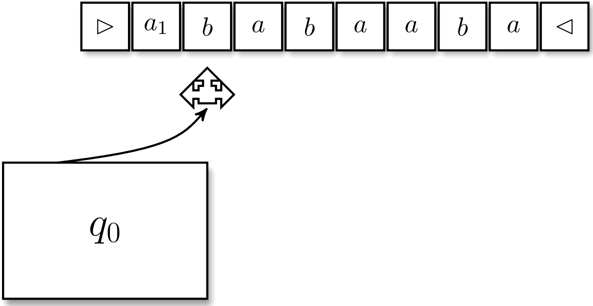
3-Limited Automaton – Example



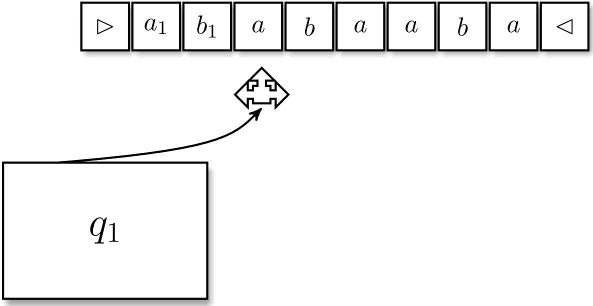
3-Limited Automaton – Example



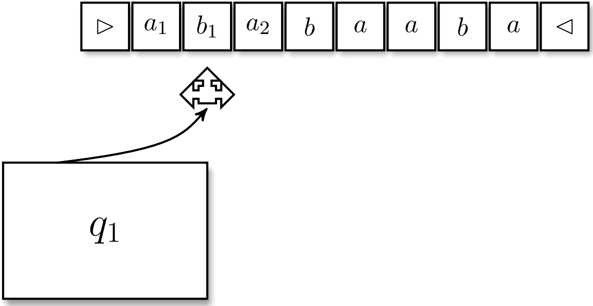
3-Limited Automaton – Example



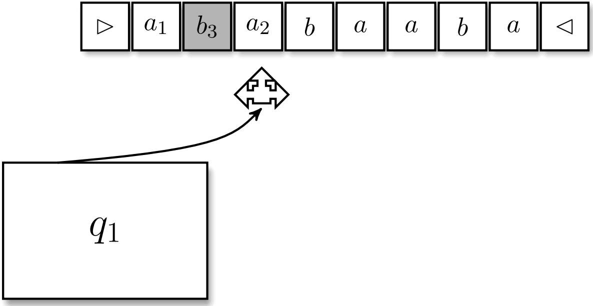
3-Limited Automaton – Example



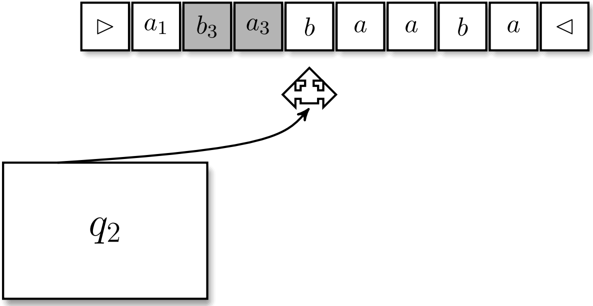
3-Limited Automaton – Example



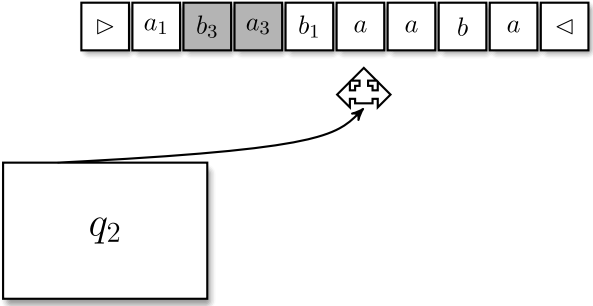
3-Limited Automaton – Example



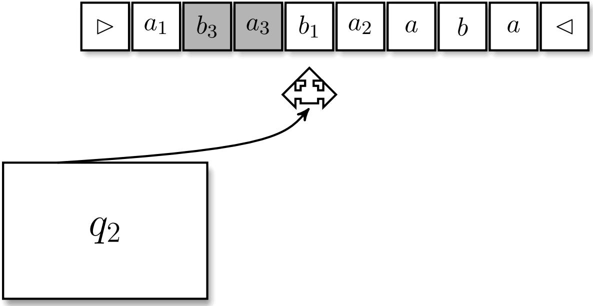
3-Limited Automaton – Example



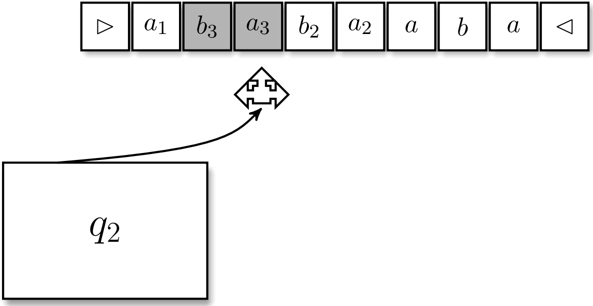
3-Limited Automaton – Example



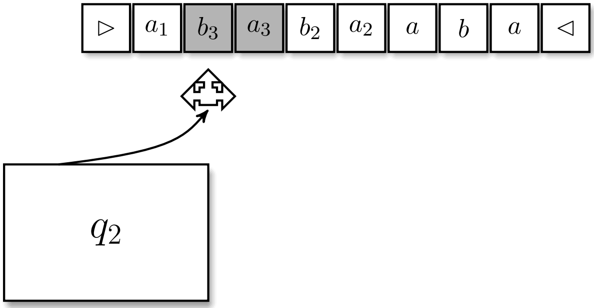
3-Limited Automaton – Example



3-Limited Automaton – Example



3-Limited Automaton – Example



Limited Automata – Definition

An input is accepted if the automaton halts at some time in an accepting state, otherwise it is rejected.

Example

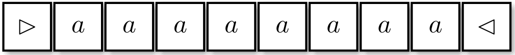
The language $L_{n,k} = \{a^{n^{k+1}}\}$ is accepted by a sweeping k-limited automaton with $n + 2$ states and $2k + 1$ tape symbols.

$\triangleright$	a	a	a	a	a	a	a	a	$\triangleleft$

$n = 2, k = 2$

Example

The language $L_{n,k} = \{a^{n^{k+1}}\}$ is accepted by a sweeping k-limited automaton with $n + 2$ states and $2k + 1$ tape symbols.



$n = 2, k = 2$

Example

The language $L_{n,k} = \{a^{n^{k+1}}\}$ is accepted by a sweeping k-limited automaton with $n + 2$ states and $2k + 1$ tape symbols.

$\triangleright$	a_1	a	a	a	a	a	a	a	$\triangleleft$

$n = 2, k = 2$

Example

The language $L_{n,k} = \{a^{n^{k+1}}\}$ is accepted by a sweeping k-limited automaton with $n + 2$ states and $2k + 1$ tape symbols.

$\triangleright$	a_1	a_1'	a	a	a	a	a	a	$\triangleleft$

$n = 2, k = 2$

Example

The language $L_{n,k} = \{a^{n^{k+1}}\}$ is accepted by a sweeping k-limited automaton with $n+2$ states and $2k+1$ tape symbols.

$\triangleright$	a_1	a_1'	a_1	a	a	a	a	a	$\triangleleft$

$n = 2, k = 2$

Example

The language $L_{n,k} = \{a^{n^{k+1}}\}$ is accepted by a sweeping k-limited automaton with $n+2$ states and $2k+1$ tape symbols.

$\triangleright$	a_1	a'_1	a_1	a'_1	a	a	a	a	$\triangleleft$

$n = 2, k = 2$

Example

The language $L_{n,k} = \{a^{n^{k+1}}\}$ is accepted by a sweeping k-limited automaton with $n + 2$ states and $2k + 1$ tape symbols.

$\triangleright$	a_1	a'_1	a_1	a'_1	a_1	a	a	a	$\triangleleft$

$n = 2, k = 2$

Example

The language $L_{n,k} = \{a^{n^{k+1}}\}$ is accepted by a sweeping k-limited automaton with $n+2$ states and $2k+1$ tape symbols.

$\triangleright$	a_1	a_1'	a_1	a_1'	a_1	a_1'	a	a	$\triangleleft$

$n = 2, k = 2$

Example

The language $L_{n,k} = \{a^{n^{k+1}}\}$ is accepted by a sweeping k-limited automaton with $n + 2$ states and $2k + 1$ tape symbols.

$\triangleright$	a_1	a'_1	a_1	a'_1	a_1	a'_1	a_1	a	$\triangleleft$

$n = 2, k = 2$

Example

The language $L_{n,k} = \{a^{n^{k+1}}\}$ is accepted by a sweeping k-limited automaton with $n+2$ states and $2k+1$ tape symbols.

$\triangleright$	a_1	a'_1	a_1	a'_1	a_1	a'_1	a_1	a'_1	$\triangleleft$

$n = 2, k = 2$

Example

The language $L_{n,k} = \{a^{n^{k+1}}\}$ is accepted by a sweeping k-limited automaton with $n + 2$ states and $2k + 1$ tape symbols.

$\triangleright$	a'_2	a'_2	a_2	a'_2	a'_2	a'_2	a_2	a'_2	$\triangleleft$

$n = 2, k = 2$

Simulation Costs of 1-DLA

The Landau Function

As is often the case in connection with unary languages, the Landau function

$$F(n) = \max\{ \operatorname{lcm}(c_1, c_2 \ldots, c_l) \mid c_1, c_2, \ldots, c_l \geq 1$$

$$\text{and } c_1 + c_2 + \cdots + c_l = n \},$$

is used.

The Landau Function

The following approximation of the Landau function is often used:

$$F(n) \in e^{\Theta(\sqrt{n \cdot \ln n})}$$

A closer look (Ellul 2004) shows that

$$F(n) \in \Omega\left(e^{\sqrt{n \cdot \ln(n)}}\right) \quad \text{and} \quad F(n) \in e^{\sqrt{n \cdot \ln(n)}(1+o(1))}.$$

Simulation Costs of 1-DLA

Theorem

Let $n \geq 2$ be a prime number. Then there is a unary $4n$-state and $n+1$ tape symbol 1-DLA M, such that $n \cdot F(n)$ states are necessary for any 2NFA to accept the language $L(M)$.

Simulation Costs of 1-DLA

$\mathrm{lcm}(c_1, c_2, \ldots, c_l) = F(n)$

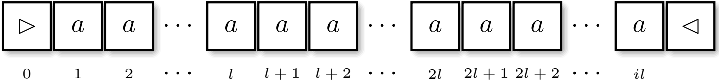
$\triangleright \; a \; a \cdots a \; a \; a \cdots a \; a \; a \cdots a \; \triangleleft$

test:

$|w| \equiv l - 1 \bmod c_1$
$|w| \equiv l - 2 \bmod c_2$
$\cdots$
$|w| \equiv 0 \bmod c_l$
$|w| \equiv 0 \bmod l$

Simulation Costs of 1-DLA

$\text{lcm}(c_1, c_2, \ldots, c_l) = F(n)$



$$\text{test:}\quad \begin{array}{l} |w| \equiv l-1 \text{ mod } c_1 \\ |w| \equiv l-2 \text{ mod } c_2 \\ \cdots \\ |w| \equiv 0 \text{ mod } c_l \\ |w| \equiv 0 \text{ mod } l \end{array}$$

Simulation Costs of 1-DLA

$\mathrm{lcm}(c_1, c_2, \ldots, c_l) = F(n)$

$\triangleright$	a	a	$\cdots$	a	a	a	$\cdots$	a	a	a	$\cdots$	a	$\triangleleft$
0	1	2	$\cdots$	l	$l+1$	$l+2$	$\cdots$	$2l$	$2l+1$	$2l+2$	$\cdots$	il	
	c_1	c_2	$\cdots$	c_l	c_1	c_2	$\cdots$	c_l	c_1	c_2	$\cdots$	c_l	

test:

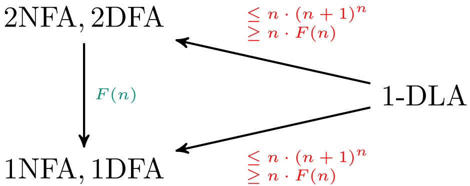
$$|w| \equiv l - 1 \text{ mod } c_1$$
$$|w| \equiv l - 2 \text{ mod } c_2$$
$$\cdots$$
$$|w| \equiv 0 \text{ mod } c_l$$
$$|w| \equiv 0 \text{ mod } l$$

Simulation Costs of 1-DLA

Corollary

Let $n \geq 2$ be a prime number. Then there is a unary $4n$-state and $n+1$-tape-symbol 1-DLA M, such that $n \cdot F(n)$ states are necessary for any 2DFA, 1DFA, or 1NFA to accept the language $L(M)$.

Summary



Simulation Costs of k-DLA

Simulation Costs of sweeping k-DLA

Theorem

Let $k, n \geq 1$ be integers and M be a unary n-state sweeping k-DLA. Then $O(n^{\frac{k^2+3k+2}{2}})$ states are sufficient for a 2DFA to accept the language $L(M)$. The 2DFA can effectively be constructed from M.

Simulation Costs of sweeping k-DLA

Theorem

Let $k, n \geq 1$ be integers and M be a unary n-state sweeping k-DLA. Then $O(n^{\frac{k^2+3k+2}{2}})$ states are sufficient for a $2\mathrm{DFA}$ to accept the language $L(M)$. The $2\mathrm{DFA}$ can effectively be constructed from M.

Simulation Costs of sweeping k-DLA

Theorem

Let $k, n \geq 1$ be integers and M be a unary n-state sweeping k-DLA. Then $O(n^{\frac{k^2+3k+2}{2}})$ states are sufficient for a 2DFA to accept the language $L(M)$. The 2DFA can effectively be constructed from M.

Simulation Costs of sweeping k-DLA

Theorem

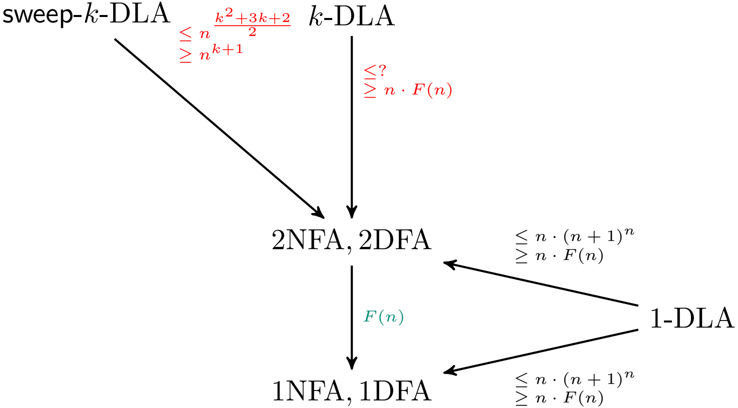
Let $k, n \geq 1$ be integers and M be a unary n-state sweeping k-DLA. Then $O(n^{\frac{k^2+3k+2}{2}})$ states are sufficient for a 2DFA to accept the language $L(M)$. The 2DFA can effectively be constructed from M.

Simulation Costs of sweeping k-DLA

Theorem

Let $k \geq 1$ and $n \geq 2$ be integers. Then there is a unary sweeping $(n+2)$-state, $(2k+1)$-tape-symbol k-DLA M, so that n^{k+1} states are necessary for any 2NFA, 2DFA, 1NFA, or 1DFA to accept the language $L(M)$.

Summary


sweep-k-DLA
$\leq n^{\frac{k^2+3k+2}{2}}$
$\geq n^{k+1}$
k-DLA
$\leq ?$
$\geq n \cdot F(n)$
2NFA, 2DFA
$\leq n \cdot (n+1)^n$
$\geq n \cdot F(n)$
$F(n)$
1-DLA
1NFA, 1DFA
$\leq n \cdot (n+1)^n$
$\geq n \cdot F(n)$

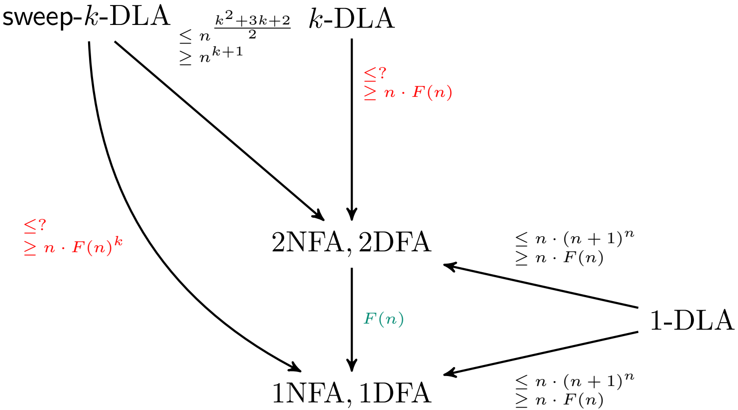
Simulation Costs of k-DLA

Theorem

Let $k, n \geq 2$ be integers so that n is prime. Then there is a unary sweeping $(n+1)$-state, $2k$-tape-symbol k-DLA M, so that $n \cdot F(n)^k$ states are necessary for any 1DFA to accept the language $L(M)$.

- The idea of the proof is to use an adapted technique of one-way k-head finite automata.

Summary



sweep-k-DLA
$\leq n^{\frac{k^2+3k+2}{2}}$
$\geq n^{k+1}$
k-DLA
$\leq ?$
$\geq n \cdot F(n)$
$\leq ?$
$\geq n \cdot F(n)^k$
2NFA, 2DFA
$\leq n \cdot (n+1)^n$
$\geq n \cdot F(n)$
$F(n)$
1-DLA
1NFA, 1DFA
$\leq n \cdot (n+1)^n$
$\geq n \cdot F(n)$

Simulation Costs of rotating k-DLA

Simulation Costs of rotating k-DLA

Theorem

Let $k, n \geq 1$ be integers and M be a unary n-state rotating k-DLA. Then $O(n^{k+1})$ states are sufficient for a (sweeping) 2DFA to accept the language $L(M)$. The 2DFA can effectively be constructed from M.

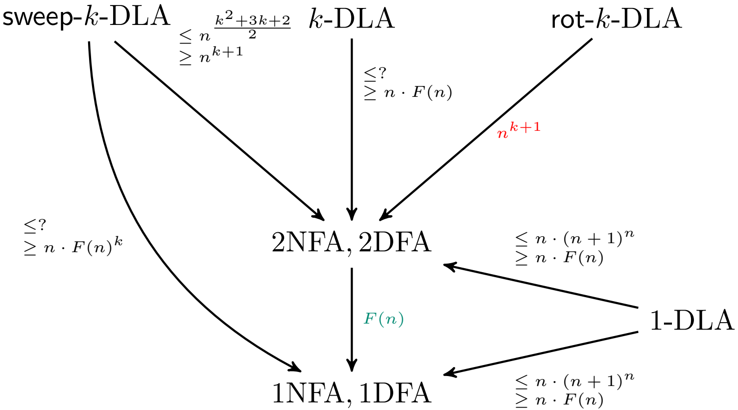
- The idea of the proof is that simulate the behaviour in the first k sweeps of M in one.

Simulation Costs of rotating k-DLA

Theorem

Let $k \geq 1$ and $n \geq 2$ be integers. Then there is a unary rotating $(n+2)$-state, $(2k+1)$-tape-symbol k-DLA M, so that n^{k+1} states are necessary for any 2NFA, 2DFA, 1NFA, or 1DFA to accept the language $L(M)$.

Summary



sweep-k-DLA
k-DLA
rot-k-DLA
$\leq n^{\frac{k^2+3k+2}{2}}$
$\geq n^{k+1}$
$\leq ?$
$\geq n \cdot F(n)$
n^{k+1}
$\leq ?$
$\geq n \cdot F(n)^k$
2NFA, 2DFA
$\leq n \cdot (n+1)^n$
$\geq n \cdot F(n)$
$F(n)$
1-DLA
1NFA, 1DFA
$\leq n \cdot (n+1)^n$
$\geq n \cdot F(n)$

Open Questions

- Is it possible to improve the upper bound for 1-DLA to DFA in the unary case?
- What is the upper bound between $k\text{-DLA and 1DFA}, \text{2DFA}$ in the unary case?
- How is the relation between sweeping and non-sweeping $k\text{-DLA}$.

Thank you for your attention!