Size of Unary One-Way Multi-Head Finite Automata

Martin Kutrib Andreas Malcher Matthias Wendlandt

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

The Costs of Optimal Simulations of Unary Automata

In several cases the Landau function describes the costs of the optimal conversion between unary automata models. But there are still some other functions.

$$\left.\begin{array}{r} \text{(Chrobak 1986)}, \mathrm{NFA} \to \mathrm{DFA} \\ 2\mathrm{DFA} \to \mathrm{DFA} \\ 2\mathrm{DFA} \to \mathrm{NFA} \\ \text{(Mereghetti, Pighizzini 2001)}, 2\mathrm{NFA} \to \mathrm{DFA} \\ 2\mathrm{NFA} \to \mathrm{NFA} \end{array}\right\} e^{\Theta(\sqrt{n \cdot \ln(n)})}$$

The Costs of Optimal Simulations of Unary Automata

In several cases the Landau function describes the costs of the optimal conversion between unary automata models. But there are still some other functions.

$$\text{(Chrobak 1986)}, \mathrm{NFA} \to \mathrm{2DFA} \} \, \Theta(n^2)$$

The Costs of Optimal Simulations of Unary Automata

In several cases the Landau function describes the costs of the optimal conversion between unary automata models. But there are still some other functions.

$$\text{(Pighizzini 2009)}, \left.\begin{array}{r} \mathrm{DPDA} \to \mathrm{DFA} \\ \mathrm{DPDA} \to \mathrm{NFA} \\ \mathrm{DPDA} \to 2\mathrm{NFA} \end{array}\right\} \Theta(2^n)$$

How about Deterministic One-Way k-Head Finite Automata?

How about Deterministic One-Way k-Head Finite Automata?

- The unary languages accepted by deterministic one-way k-head finite automata are semilinear and so regular.

How about Deterministic One-Way k-Head Finite Automata?

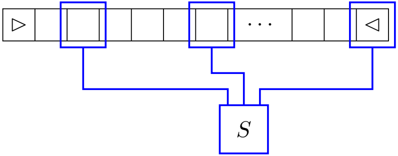
- The unary languages accepted by deterministic one-way k-head finite automata are semilinear and so regular.
- There is an infinite proper double hierarchy with respect to the number of states as well as to the number of heads (MK,AM,MW 2012).

How about Deterministic One-Way k-Head Finite Automata?

- The unary languages accepted by deterministic one-way k-head finite automata are semilinear and so regular.
- There is an infinite proper double hierarchy with respect to the number of states as well as to the number of heads (MK,AM,MW 2012).

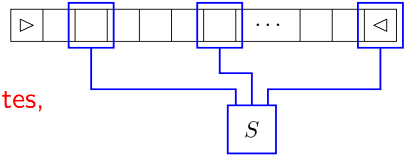
$$\begin{tikzcd}
\mathrm{DFA} & & \mathrm{NFA} \arrow[ll, bend right, "e^{\Theta(\sqrt{n\cdot\log(n)})}"'] \arrow[dl, bend left] \\
& \mathrm{1DFA}(k) \arrow[ul, bend left] \arrow[ur, bend left] &
\end{tikzcd}$$

One-Way k-Head Finite Automata



One-Way k-Head Finite Automata

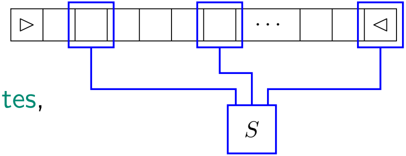
$$M = \langle S, A, k, \delta, \rhd, \lhd, s_0, F \rangle$$



- ➜ S is the finite set of internal states,
- ➜ $s_0 \in S$ is the initial state,

One-Way k-Head Finite Automata

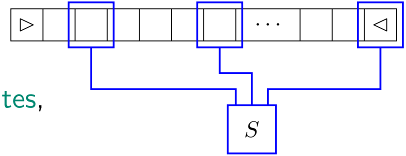
$$M = \langle S, A, k, \delta, \triangleright, \triangleleft, s_0, F \rangle$$



- S is the finite set of internal states,
- $s_0 \in S$ is the initial state,
- A is the finite set of input symbols,
- $\triangleright \notin A$ is the left and $\triangleleft \notin A$ is the right endmarker of the workspace,

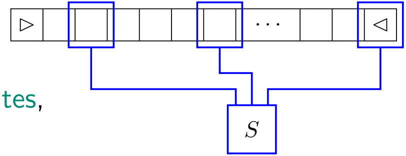
One-Way k-Head Finite Automata

$M = \langle S, A, {\color{red}k}, \delta, \triangleright, \triangleleft, s_0, F \rangle$



- S is the finite set of internal states,
- $s_0 \in S$ is the initial state,
- A is the finite set of input symbols,
- $\triangleright \notin A$ is the left and $\triangleleft \notin A$ is the right endmarker of the workspace,
- $k \geq 1$ is the number of heads,

One-Way k-Head Finite Automata

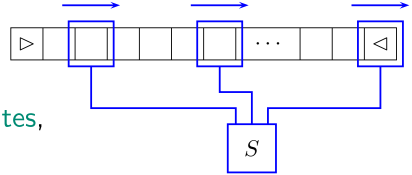


$$M = \langle S, A, k, \delta, \triangleright, \triangleleft, s_0, F \rangle$$

- S is the finite set of internal states,
- $s_0 \in S$ is the initial state,
- A is the finite set of input symbols,
- $\triangleright \notin A$ is the left and $\triangleleft \notin A$ is the right endmarker of the workspace,
- $k \geq 1$ is the number of heads,
- F is the finite set of accepting states,

One-Way k-Head Finite Automata

$M = \langle S, A, k, \delta, \triangleright, \triangleleft, s_0, F\rangle$



- S is the finite set of internal states,
- $s_0 \in S$ is the initial state,
- A is the finite set of input symbols,
- $\triangleright \notin A$ is the left and $\triangleleft \notin A$ is the right endmarker of the workspace,
- $k \geq 1$ is the number of heads,
- F is the finite set of accepting states,
- $\delta : S \times (A \cup \{\triangleright, \triangleleft\})^k \to S \times \{0, 1\}^k$ is the partial transition function.

Language Recognition

- It starts with all of its heads on the left endmarker.
- It halts when the transition function is not defined for the current situation.
- An input is accepted, if the automaton halts in an accepting state.

Example: For each $k, n \geq 2$, the unary singleton language

$$L_{k,n} = \{\, a^{(k-1)n^k} \,\}$$

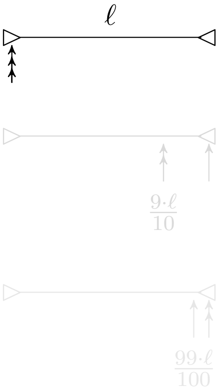
is accepted by some $\mathrm{1DFA}(k)$ with n states.

Example: For each $k, n \geq 2$, the unary singleton language

$$L_{k,n} = \{\, a^{(k-1)n^k} \,\}$$

is accepted by some $1\mathrm{DFA}(k)$ with n states.

n=10,k=3

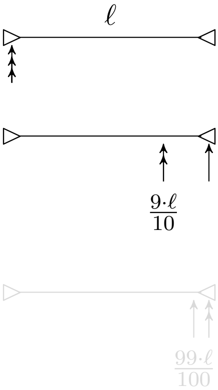


Example: For each $k, n \geq 2$, the unary singleton language

$$L_{k,n} = \{ a^{(k-1)n^k} \}$$

is accepted by some $1\mathrm{DFA}(k)$ with n states.

n=10,k=3

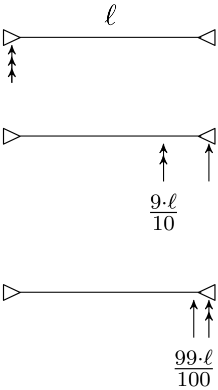


Example: For each $k, n \geq 2$, the unary singleton language

$$L_{k,n} = \{\, a^{(k-1)n^k} \,\}$$

is accepted by some $1\mathrm{DFA}(k)$ with n states.

n=10,k=3



$\mathbf{1DFA}(k)$ to $\mathbf{1DFA}(1)$

The Landau Function

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

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

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

is used.

The Landau Function

As an approximation of the landau function it is often used:

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

A closer look (Ellul 2004) shows that

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

Head Reduction—Lower Bound

Theorem

For any integers $k, n \geq 2$ so that n is prime, there is a unary n-state $\mathrm{1DFA}(k)$ M, such that $n \cdot F(n)^{k-1}$ states are necessary for any DFA to accept the language $L(M)$.

Construction of a n-state $\mathrm{1DFA}(k)$

- Let $c_1, c_2, \ldots, c_l \geq 2$ be integers such that $c_1 + c_2 + \cdots + c_l \leq n$ and $\mathrm{lcm}(c_1, c_2, \ldots, c_l) = F(n)$.
- The first head moves in a cycle of length n until it reaches the right endmarker. Depending on in which state it arrives at the right endmarker, a c_i is chosen, for which the divisibility is tested afterwards.
- Now the idea of the first example is used. So each time the next head h_j moves in a cycle of length c_i, while the others move only $c_i - 1$ times, until head h_j reaches the right endmarker.
- All together we have that $\ell = x_1 \cdot n + c_i$ and $\ell = x_k \cdot c_i^{k-1}$.
- Since both numbers n, c_i are relatively prime, an DFA accepting this language needs at least nc_i^{k-1} states.

Construction of an n-state $\mathbf{1DFA}(k)$

An immediate generalization of the proof of the state complexity for the union of two unary deterministic finite automata languages (Yu 2001) shows that every DFA accepting $L(M)$ has a cycle of at least

$$\mathrm{lcm}\{\, nc_i^{k-1} \mid 1 \le i \le l \,\} = n(c_1 c_2 \cdots c_l)^{k-1} = n \cdot F(n)^{k-1}$$

states.

Head Reduction—Upper Bound

Theorem

Let $k, n \geq 1$ and M be a unary n-state $1\mathrm{DFA}(k)$. Then there is a constant t depending only on k so that $O(n \cdot F(t \cdot n)^{k-1})$ states are sufficient for a DFA to accept the language $L(M)$. The DFA can effectively be constructed from M.

Summary

$$
\begin{tikzcd}
\mathrm{DFA} & & \mathrm{NFA} \arrow[ll, bend right, "e^{\Theta(\sqrt{n\cdot\log(n)})}"'] \arrow[dl, bend left] \\
& \mathrm{1DFA}(k) \arrow[ul, bend left, "{n \cdot F(n)^{k-1} \leq \cdot \leq n \cdot F(t \cdot n)^{k-1}}"] \arrow[ur, bend left] &
\end{tikzcd}
$$

$\mathbf{1DFA}(k)$ to $\mathbf{NFA}$—Upper Bound

Theorem

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

In (MK,AM,MW 2012) it is shown that the language of a unary n-state $\mathrm{1DFA}(k)$ can be described by finitely many equations of the form:

$$\ell = \frac{P}{Q} + x \cdot \frac{c_1 c_2 \cdots c_k}{Q}.$$

It holds $\frac{P}{Q} < 2^{k-1} k n^k$, $c_i \leq n$.

In (MK,AM,MW 2012) it is shown that the language of a unary n-state $1\mathrm{DFA}(k)$ can be described by finitely many equations of the form:

$$\ell = \frac{P}{Q} + x \cdot \frac{c_1 c_2 \cdots c_k}{Q}.$$

It holds $\frac{P}{Q} < 2^{k-1} k n^k$, $c_i \leq n$.

For an $1\mathrm{DFA}(k)$ M there are at most n^k different equations describing the language $L(M)$.

Form k Heads to One Head NFA

- For each equation a DFA can be constructed.
- The union of these automata describes all words in $L(M)$. The size of each automaton is at most $2^{k-1}kn^k + n^k$.
- The union of the different DFA is accepted by an NFA that initially guesses which of the DFA to simulate and, subsequently, simulates it.

$\mathrm{1DFA}(k)$ to NFA—Lower Bound

Theorem

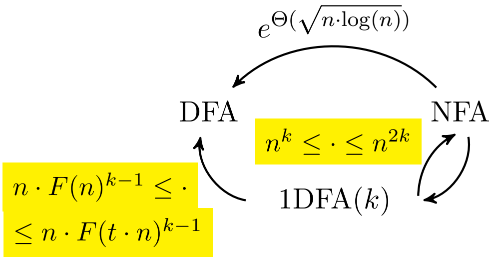
For any integers $k, n \geq 2$, there is a unary n-state $\mathrm{1DFA}(k)$ M, such that $\Omega(n^k)$ states are necessary for any NFA to accept the language $L(M)$.

Consider the unary singleton language

$$L_{k,n} = \{\, a^{(k-1)n^k} \,\}$$

is accepted by some $\mathrm{1DFA}(k)$ with n states.

Summary


$e^{\Theta(\sqrt{n\cdot\log(n)})}$
DFA
NFA
$n^k \leq \cdot \leq n^{2k}$
$n \cdot F(n)^{k-1} \leq \cdot$
$\leq n \cdot F(t \cdot n)^{k-1}$
$1\mathrm{DFA}(k)$

From One-Head NFA to k-Head DFA Upper Bound

Theorem

Let $k \geq 1$, $n \geq 2$ be constants, $t = \lfloor \frac{-3+\sqrt{8n+1}}{2} \rfloor$, and M be a unary n-state NFA. Then

$$n' \leq \begin{cases} n^2 - 2 + F(n), & \text{if } k = 1; \\ n^2 - 2 + \left(n - \frac{t^2+t}{2}\right)^{\lceil \frac{t}{k} \rceil}, & \text{if } 1 < k < t/2; \\ 2n^2, & \text{if } k \geq t/2. \end{cases}$$

states are sufficient for a $1\mathrm{DFA}(k)$ M' to accept the language $L(M)$. The $1\mathrm{DFA}(k)$ can effectively be constructed from M.

The basic idea is to let each head of the $1\mathrm{DFA}(k)$ M' simulates the behavior of $\lceil t/k \rceil$ cycles of M.

From One Head NFA to k Head DFA Lower Bound

Theorem

Let $k \geq 1$ be a constant. For any integer $m \geq 1$ there is an integer $n > m$ and a unary n-state $\mathrm{NFA}\ M$, such that $c_2 \cdot \sqrt[k]{e^{\frac{\sqrt{2n}}{\sqrt{c_1 \ln(\sqrt{2n})}}}}$ states are necessary for any $\mathrm{1DFA}(k)$ to accept the language $L(M)$, where $c_1, c_2 > 0$ are two constants.

Summary

$$\begin{tikzcd}[column sep=large, row sep=large]
\text{DFA} & & \text{NFA} \arrow[ll, bend right, "e^{\Theta(\sqrt{n\cdot\log(n)})}"'] \arrow[dl, bend left] \\
& \text{1DFA}(k) \arrow[ul, bend left] \arrow[ur, bend left] &
\end{tikzcd}$$

$$n \cdot F(n)^{k-1} \leq \cdot \leq n \cdot F(t \cdot n)^{k-1}$$

$$n^k \leq \cdot \leq n^{2k}$$

$$c_2 \cdot \sqrt[k]{e^{\frac{\sqrt{2n}}{\sqrt{c_1 \ln(\sqrt{2n})}}}} \leq \cdot \leq n^2 - 2 + \left(n - \frac{t^2+t}{2}\right)^{\left\lceil \frac{t}{k} \right\rceil}$$

Computational Complexity

Theorem

Let $k \geq 1$ be an integer. Then for $1\mathrm{DFA}(k)$ accepting unary languages the problems of testing emptiness, universality, finiteness, inclusion, and equivalence are LOGSPACE-complete.

- It has been shown that the language $L(M)$ can be represented as union of some languages accepted by DFA, so that each DFA has at most $2^{k-1}kn^k + n^k$ states.
- The idea for the Turing machine M is to simulate the given $1\mathrm{DFA}(k)$ M' successively on all inputs of length at most $2^{k-1}kn^k + n^k$ until some input is accepted or all inputs tested are rejected.

Computational Complexity

Lemma

Let $k \geq 1$ and M be an n-state $\mathrm{1DFA}(k)$. Then there exists an n-state $\mathrm{1DFA}(k)$ M' accepting the complement of $L(M)$. The $\mathrm{1DFA}(k)$ M' can effectively be constructed from M.

- In order to decide non-universality, it has to be decided whether $\overline{L(M)}$, is non-empty.
- Infiniteness can similarly be tested as emptiness.
- To decide whether or not $L(M_1)$ is included in $L(M_2)$, for two $\mathrm{1DFA}(k)$ M_1 and M_2, one can decide whether $L(M_1) \cap \overline{L(M_2)}$ is empty and closely the same for equivalence.

Open Questions

- What does the reduction of only one head cost?
- Can we tighten the bound for converting a NFA to $1\mathrm{DFA}(k)$?
- What about simulating a nondeterministic one-way multihead finite automaton by DFAs or NFAs?

Thank you for your attention