SIAM J. COMPUT.
Vol. 30, No. 6, pp. 1976–1992

OPTIMAL SIMULATIONS BETWEEN UNARY AUTOMATA*

CARLO MEREGHETTI† AND GIOVANNI PIGHIZZINI‡

Abstract. We consider the problem of computing the costs—*in terms of states*—of optimal simulations between different kinds of finite automata recognizing *unary* languages. Our main result is a tight simulation of unary n-state two-way nondeterministic automata by $O(\mathrm{e}^{\sqrt{n \ln n}})$-state one-way deterministic automata. In addition, we show that, given a unary n-state two-way nondeterministic automaton, one can construct an equivalent $O(n^2)$-state two-way nondeterministic automaton performing *both* input head reversals and nondeterministic choices only at the ends of the input tape. Further results on simulating unary one-way alternating finite automata are also discussed.

Key words. formal languages; deterministic, nondeterministic, and alternating finite state automata; unary languages

AMS subject classifications. 68Q10, 68Q45, 68Q68

PII. S009753979935431X

1. Introduction.

Finite automata are probably one of the simplest and most extensively studied models of computation. First of all, their computational power is well established: they exactly define the class of regular languages. Furthermore, it is also well known that several added features, such as nondeterminism, alternation, and two-way motion of the input head, do not increase their computing ability. Equivalences are clearly obtained by simulating different kinds of automata by the original device, the one-way deterministic finite automaton (1dfa) [22]. Here, we are particularly interested in the cost—*in terms of states*—of simulations between automata.

The following are the well-known costs of simulating different automata by 1dfa's (number of states of the best 1dfa simulating any n-state automaton in the class): one-way nondeterministic finite automata (1nfa): $O(2^n)$ [22], one-way alternating finite automata (1afa): $O(2^{2^n})$ [7], two-way deterministic finite automata (2dfa): $O(n^n)$ [22, 24], two-way nondeterministic finite automata (2nfa): $O(2^{n^2})$ [22, 24]. All these bounds are tight. We refer the reader to [3] which is a valuable source for results and references.

There are several open questions concerning automata simulation, the most important being probably the one posed by Sakoda and Sipser in [23]: how many states are necessary and sufficient to simulate 2nfa's (or 1nfa's) by 2dfa's? They conjecture such a cost to be exponential, and Sipser [25] proves that this is exactly the case when 2dfa's are required to be *sweeping*, i.e., to have head reversals only at the ends of the input tape. Berman and Lingas [2] state a lower bound of $\Omega(n^2/\log n)$ for the general problem and provide an interesting connection with the celebrated open problem DLOGSPACE $\stackrel{?}{=}$ NLOGSPACE. More precisely, they show that

*Received by the editors April 20, 1999; accepted for publication (in revised form) July 21, 2000; published electronically March 20, 2001. This work was partially supported by MURST, under the project "Modelli di calcolo innovativi: metodi sintattici e combinatori." A preliminary version of this paper [20] has been presented to the 15*th Annual Symposium on Theoretical Aspects of Computer Science* 1998 *(STACS*98*)*, Paris, France, February 25–27, 1998.
http://www.siam.org/journals/sicomp/30-6/35431.html

†Dipartimento di Informatica, Sistemistica e Comunicazione, Università degli Studi di Milano–Bicocca, via Bicocca degli Arcimboldi 8, 20126 Milano, Italy (mereghetti@disco.unimib.it).

‡Dipartimento di Scienze dell'Informazione, Università degli Studi di Milano, via Comelico 39, 20135 Milano, Italy (pighizzi@dsi.unimi.it).

if DLOGSPACE = NLOGSPACE then for some polynomial p and for all integers m and k-state 2nfa A, there is a $p(mk)$-state 2dfa accepting a subset of $\mathcal{L}(A)$, the language accepted by A. The subset consists of all strings of length no more than m in $\mathcal{L}(A)$. As a consequence of this result, Sipser [25] relates the DLOGSPACE $\stackrel{?}{=}$ NLOGSPACE question also to the existence of sweeping automata with a polynomial number of states for a certain family of regular languages. This might give added evidence that the problem of evaluating how the number of states changes when turning one automaton into another is not only motivated by the investigation on the succinctness of representing regular languages but is also related to fundamental questions in complexity.

It is important to stress that the optimality of such simulations has been established by witness languages built over alphabets of two or more symbols. As a matter of fact, the situation turns out to be quite different whenever we restrict the problems to *unary* automata, i.e., automata with a single letter input alphabet. The problem of evaluating the costs of unary automata simulations was raised in [25] and has led to emphasize some relevant differences with the general case. For instance, we know that $O(\mathrm{e}^{\sqrt{n \ln n}})$ states suffice in order to simulate a unary n-state 1nfa or 2dfa by a 1dfa. Furthermore, a unary n-state 1nfa can be simulated by a 2dfa having $O(n^2)$ many states. All these results and their optimality have been proved in 1986 by Chrobak [8].

In this paper, we further deepen the study of optimal simulations between unary automata. To this aim, we find it useful to consider some techniques from the *sublogarithmic space world* (see, e.g., [11, 12, 27]).

The first part of the paper is devoted to study unary 2nfa's. We closely analyze the structure of their computation paths. In particular, using graph theoretical and number theoretical arguments, we show that, for sufficiently large inputs, it is possible to consider only computation paths in which states are repeated in a very regular way. This allows us to state our main result concerning the optimal simulation of unary 2nfa's by 1dfa's: each unary n-state 2nfa can be optimally simulated by a 1dfa with $O(\mathrm{e}^{\sqrt{n \ln n}})$ states. Note that such a complexity is the same as the above mentioned optimal simulations of unary 1nfa's and 2dfa's by 1dfa's. Thus, we can conclude that the simultaneous elimination of both two-way motion and nondeterminism on unary automata has the same cost as the elimination of either of them.

As another consequence of our analysis of unary 2nfa's, we are able to prove that each unary n-state 2nfa can be simulated by a 2nfa with $O(n^2)$ states which reverses the input head direction and makes nondeterministic decisions *only* when the input head visits the left or the right end of the input. This result can be regarded as a further step toward the solution of the Sakoda–Sipser open problem recalled above, at least for unary inputs.

In the second part of the paper, we study the relationships between unary 1afa's, dfa's and nfa's. This problem was proposed in [8] and partly solved in [3], where it is proved that 2^n states are necessary for 2nfa's to simulate unary n-state 1afa's. Here, we point out an optimal simulation of unary n-state 2nfa's by $O(\sqrt{n \ln n})$-state 1afa's.

By combining our results with those in the literature, we are able to draw an almost complete description of the costs of optimal simulations between the different types of unary automata here considered. We do not explicitly list all the resulting optimal bounds, which can be better understood by observing Figure 1.1.

2. Preliminary notions and results. In this section, we begin by recalling basic notions on finite state automata (subsection 2.1). Then, for the reader's ease of

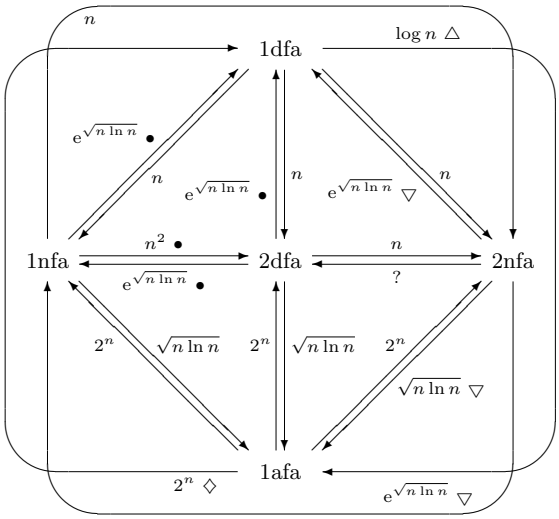

FIG. 1.1. *Costs of the optimal simulations between different kinds of unary automata. An arc labeled $f(n)$ from a vertex x to a vertex y means that a unary n-state automaton of type x can be simulated by a $O(f(n))$-state automaton of type y. Costs marked by "•" are proved in* [8]*, by "◇" in* [7]*, by "△" in* [17]*, by "▽" in this paper. The unmarked costs can be trivially obtained. The arc labeled "?" represents the open question of Sakoda and Sipser* [23].

mind, a brief outline of the main result of this paper—the $O(e^{\sqrt{n \ln n}})$ simulation of 2nfa's by 1dfa's—is provided (subsection 2.2). In the following two subsections, we set up some mathematical tools to study the costs of our simulations. More precisely, we prove some new facts concerning *linear Diophantine equations* (subsection 2.3) and *directed graphs* (subsection 2.4).

2.1. Finite state automata. Given a set S, $\sharp S$ denotes its cardinality, and 2^S the family of all its subsets. Given an alphabet Σ, Σ^* denotes the set of strings on Σ, with the empty string ϵ. Given a string $x \in \Sigma^*$, $|x|$ denotes its length. A language $\mathcal{L}$ is said to be *unary* (or *tally*) whenever it can be built over a *single letter* alphabet. In this case, we let $\mathcal{L} \subseteq 1^*$.

Let us take a brief look on the computational model of finite automata. For a detailed exposition, we refer the reader to [14]. A *one-way nondeterministic finite automaton* is a 5-tuple $A = (Q, \Sigma, \delta, q_0, F)$, where Q is the finite set of states, Σ is the finite input alphabet, $\delta : Q \times \Sigma \to 2^Q$ is the transition function, $q_0 \in Q$ is the initial state, $F \subseteq Q$ is the set of final states. The transition function δ can be extended to strings in a standard way. The language accepted by A is the set $\mathcal{L}(A) = \{x \in \Sigma^* \mid \delta(q_0, x) \cap F \neq \emptyset\}$. The one-way automaton A is *deterministic* if and only if $\sharp\delta(q, \sigma) = 1$, for any $q \in Q$, and $\sigma \in \Sigma$. (This implies that deterministic automata are assumed to be *complete*.)

It is well known that a one-way automaton can be viewed as a control unit that reads, by an input head, a tape containing input strings stored one symbol per square. At each move, the input head is shifted one square right. This model can be extended to encompass *stationary* and *left* moves of the input head. More precisely, we can define a 2nfa as a 5-tuple $A = (Q, \Sigma, \delta, q_0, F)$ in which Q, Σ, q_0, and F are defined as for 1nfa's, while $\delta : Q \times (\Sigma \cup \{\vdash, \dashv\}) \to 2^{Q \times \{-1, 0, +1\}}$ is the transition function, where

$\vdash, \dashv$ are two special symbols—the left and the right endmarker, respectively—not in Σ. Input strings are stored onto the input tape surrounded by the two endmarkers, the left endmarker being at the 0th square. The computation begins with the input head scanning the 0th square. In a move, A reads an input symbol, changes its state, and moves the input head one position forward or backward or keeps it stationary, depending on whether δ returns $+1$, -1, or 0, respectively. Left and right moves on the left and right endmarker, respectively, are forbidden. A is *deterministic* if and only if $\sharp\delta(q, \sigma) = 1$, for any $q \in Q$, and $\sigma \in \Sigma \cup \{\vdash, \dashv\}$.

Alternating finite automata[1] provide a natural generalization of nondeterministic automata. We now briefly recall their formal definition and refer the reader to [7, 10] for more details. A *one-way alternating finite automaton* is a 5-tuple $A = (Q, \Sigma, g, q_1, F)$, where $Q = \{q_1, \ldots, q_k\}$ is the set of states, Σ is the input alphabet, $F \subseteq Q$ is the set of final states, q_1 is the initial state, and $g : Q \to (\Sigma \times \mathbf{B}^k \to \mathbf{B})$ is a mapping of Q into the set of all mappings of $\Sigma \times \mathbf{B}^k$ into $\mathbf{B} = \{0, 1\}$. To explain the behavior of A, consider, for $i = 1, \ldots, k$, the function $g_i : \Sigma \times \mathbf{B}^k \to \mathbf{B}$ representing the image by g of the state q_i, i.e., $g_i = g(q_i)$. Such functions can be inductively extended to strings to obtain $G_i : \Sigma^* \times \mathbf{B}^k \to \mathbf{B}$ as: $G_i(\epsilon, \mathbf{u}) = u_i$, and $G_i(\sigma x, \mathbf{u}) = g_i(\sigma,\ G_1(x, \mathbf{u}), \ldots, G_k(x, \mathbf{u}))$, for $\sigma \in \Sigma$, $x \in \Sigma^*$, and $\mathbf{u} = (u_1, \ldots, u_k) \in \mathbf{B}^k$. The language accepted by A is the set $\mathcal{L}(A) = \{x \in \Sigma^* \mid G_1(x, \mathbf{f}) = 1\}$, where $\mathbf{f} \in \mathbf{B}^k$ denotes the characteristic vector of the set of final states. Notice that $G_i(\epsilon, \mathbf{f}) = 1$ if and only if q_i is a final state. Moreover, $G_i(\sigma x, \mathbf{f})$ can be computed as follows: a process in the state q_i reads σ from the input tape, and splits into k independent processes computing $G_j(x, \mathbf{f})$, for $j = 1, \ldots, k$. Then, by applying $g_i(\sigma, \_)$ to all these results, we get $G_i(\sigma x, \mathbf{f})$.

It is easy to see that 1nfa's are just special cases of 1afa's up to setting

$$g_i(\sigma, \mathbf{u}) = \bigvee_{q_j \in \delta(q_i, \sigma)} u_j,$$

for $i = 1, \ldots, k$, $\sigma \in \Sigma$, and $\mathbf{u} \in \mathbf{B}^k$.

We call *unary* any automaton that works with a single letter input alphabet.

2.2. Outline of the main result. The proof of the optimal $O(\mathrm{e}^{\sqrt{n \ln n}})$ simulation of 2nfs's by 1dfa's, which is the main result of this paper, is rather long and complex. To help the reader, we emphasize the main ideas here.

Let $\mathcal{C}$ be an accepting computation path of an n-state 2nfa A on input 1^m. Along $\mathcal{C}$, consider the sequence $r_0, r_1, \ldots, r_p$ of states in which the input head scans either of the endmarkers. For $j = 1, \ldots, p$, the following two possibilities arise:

(i) *U-Turn*: in both r_{j-1} and r_j, the input head scans the same endmarker.

(ii) *Left-to-right (right-to-left) traversal:* in r_{j-1} the input head scans the left (right) endmarker, while in r_j it scans the other one.

For sufficiently large inputs, U-Turns can be nondeterministically guessed without moving the input head (by Lemma 3.1). As a consequence, the computation $\mathcal{C}$ can be reduced to a sequence of left-to-right and right-to-left traversals.

Now, the key point is that in each traversal the automaton A can only measure the length of the input 1^m modulo some integer $\ell_i \leq n$. Hence, it can be argued that if there exists a traversal from r_{j-1} to r_j on 1^m, then there also exists a traversal with the same endpoints on $1^{m+\mu\ell_i}$, for every (possibly negative) integer μ exceeding

[1] Automata hereafter defined are sometimes, and perhaps more appropriately, called *boolean automata* (see, e.g., [6, 17]).

a certain value. This enables us to prove that *the language accepted by A forms an ultimately periodic set of period $\ell = \mathrm{lcm}(\ell_1, \ldots, \ell_r)$*, where the ℓ_i's depend on the cycle structure of the transition graph of A and their sum does not exceed n (Theorem 3.5).

By using a number theoretical result that will be recalled in Lemma 2.1, we get that $\ell = O(\mathrm{e}^{\sqrt{n \ln n}})$. So, the number of states in the cycle of a 1dfa A' simulating A is $O(\mathrm{e}^{\sqrt{n \ln n}})$, and this actually turns out to be an upper bound on the number of all states of A'.

As the reader will notice when we get into proof details, the study of path structure in certain weighted digraphs representing automata transitions will be crucial. Such graph theoretical problems are tackled in subsection 2.4 by using some number theoretical tools presented in the next subsection.

2.3. Number theory and Diophantine equations. As usual, we let $\mathbf{Z}^+$ ($\mathbf{Z}^-$) be the set of positive (negative) integers, and $\mathbf{Z}$ be the set of whole integers. Natural numbers are the elements of the set $\mathbf{N} = \mathbf{Z}^+ \cup \{0\}$. The absolute value of z is denoted by $|z|$. For any $z > 0$, $\ln z$ denotes the natural logarithm of z, while $\log z$ is the logarithm of z taken to the base 2. The *greatest common divisor* of integers $a_1, \ldots, a_s$ is denoted by $\gcd(a_1, \ldots, a_s)$. Their *least common multiple* is denoted by $\mathrm{lcm}(a_1, \ldots, a_s)$. Both gcd and lcm are, actually, meant to be taken on $|a_1|, \ldots, |a_s|$. We sometimes write $\gcd(A)$ to denote the greatest common divisor of the integers in the set A.

In estimating the simulation costs of unary automata, a crucial role is played by the function

$$F(n) = \max\{\mathrm{lcm}(x_1, \ldots, x_s) \mid x_1, \ldots, x_s \in \mathbf{Z}^+ \text{ and } x_1 + \cdots + x_s = n\}.$$

Evaluating the growth rate of $F(n)$ is known as Landau's problem [15, 16]. Such a problem is related to the study of the maximal order in the symmetric group S_n [26, 28]. Several approximations for $F(n)$ are given in the literature. The best one is contained in [26, Theorem I] from which we can derive the following upper bound that suffices to our purposes.

LEMMA 2.1. $F(n) = O(\mathrm{e}^{\sqrt{n \ln n}})$.

The other arithmetical tool we shall make use of is the theory of linear Diophantine equations, whose principles can be found, for instance, in [21]. We consider equations in the form

$$a_1 x_1 + \cdots + a_s x_s = z, \tag{2.1}$$

where $a_1, \ldots, a_s, z$ are given integers, and $x_1, \ldots, x_s$ are integer variables. It is a very well-known fact that (2.1) has (infinitely many) solutions in integers if and only if $\gcd(a_1, \ldots, a_s)$ divides z.

THEOREM 2.2. *For any given integers $a_1, \ldots, a_s$, the set of integers*

$$\{a_1 x_1 + \cdots + a_s x_s \mid x_1, \ldots, x_s \in \mathbf{Z}\}$$

is exactly the set of all integral multiples of $\gcd(a_1, \ldots, a_s)$.

We are sometimes interested in solving Diophantine equations in *natural numbers*. To this regard, an interesting question, first raised by Frobenius (see [4, 5]), can be stated as follows: given positive integers $a_1, \ldots, a_s$ satisfying $\gcd(a_1, \ldots, a_s) = 1$, what is the greatest number b such that the Diophantine equation $a_1 x_1 + \cdots + a_s x_s = b$ has no solution in natural numbers? For $s = 2$, such b is known to be $(a_1 - 1)(a_2 - 1) - 1$

[4, 5]. For $s \geq 3$, the problem is still open, even though several upper bounds are proved. We will refer to the following one.

THEOREM 2.3 (see [4, 5, 19]). *Let $a_1 < a_2 < \cdots < a_s$ be positive integers with $\gcd(a_1, \ldots, a_s) = 1$. Then, the greatest number b such that the Diophantine equation $a_1x_1 + \cdots + a_sx_s = b$ has no solution in natural numbers does not exceed $(a_1 - 1)(a_s - 1)$.*

More accurate estimations can be found in [9]. However, the bound in Theorem 2.3 will suffice for our purposes. By combining Theorem 2.2 and Theorem 2.3, one easily obtains the following corollary.

COROLLARY 2.4. *Let $a_1, \ldots, a_s$ be positive integers less than or equal to n, and let*

$$\mathcal{X} = \{a_1x_1 + \cdots + a_sx_s \mid x_1, \ldots, x_s \in \mathbf{N}\}.$$

Then, the set $\mathcal{X} \cap \{z \in \mathbf{Z}^+ \mid z > n^2\}$ is exactly the set of all integral multiples of $\gcd(a_1, \ldots, a_s)$ greater than n^2.

In order to consider sets defined not only by positive coefficients, but even by *both* positive and negative coefficients, we generalize Corollary 2.4 as follows.

LEMMA 2.5. *Let $A = \{a_1, \ldots, a_p\}$ and $B = \{b_1, \ldots, b_q\}$ be sets of positive integers less than or equal to n, with $\gcd(A) = \alpha$ and $\gcd(B) = \beta$. Moreover, let*

$$\mathcal{X} = \{a_1x_1 + \cdots + a_px_p \; - \; (b_1y_1 + \cdots + b_qy_q) \mid x_1, \ldots, x_p\,, y_1, \ldots, y_q \in \mathbf{N}\},$$

with $\gcd(A \cup B) = \gcd(\alpha, \beta) = \gamma$.

(a) *If $A = \emptyset$ ($B = \emptyset$), then $\mathcal{X} \cap \{z \in \mathbf{Z}^- \mid z < -n^2\}$ ($\mathcal{X} \cap \{z \in \mathbf{Z}^+ \mid z > n^2\}$) is exactly the set of negative (positive) integral multiples of β (α) smaller than $-n^2$ (greater than n^2).*

(b) *If $A \neq \emptyset$ and $B \neq \emptyset$, then $\mathcal{X}$ is exactly the set of integral multiples of γ.*

Proof. (a) follows trivially from Corollary 2.4. For (b) let us denote by $\mathcal{T}$ the set of integral multiple of γ. It is easy to see that $\mathcal{X} \subseteq \mathcal{T}$. In fact, each number in $\mathcal{X}$ is obtained from the Diophantine equation defining $\mathcal{X}$ itself, and hence it must be a multiple of γ.

Conversely, to show $\mathcal{T} \subseteq \mathcal{X}$, let us first define the sets

$$\mathcal{A} = \{a_1x_1 + \cdots + a_px_p \mid x_1, \ldots, x_p \in \mathbf{N}\},$$
$$\mathcal{B} = \{b_1y_1 + \cdots + b_qy_q \mid y_1, \ldots, y_q \in \mathbf{N}\}.$$

Corollary 2.4 says that $\mathcal{A} \cap \{z \in \mathbf{Z}^+ \mid z > n^2\}$ ($\mathcal{B} \cap \{z \in \mathbf{Z}^+ \mid z > n^2\}$) is exactly the set of positive integral multiples of α (β) greater than n^2. Moreover, it is easy to see that

$$\mathcal{X} = \{h - k \mid h \in \mathcal{A} \text{ and } k \in \mathcal{B}\}. \tag{2.2}$$

So, consider $\xi \in \mathcal{T}$. Since ξ is an integral multiple of $\gamma = \gcd(\alpha, \beta)$, there exist two integers $\overline{x}$ and $\overline{y}$ such that $\xi = \alpha\overline{x} - \beta\overline{y}$. Moreover, it is well known that there are infinitely many solutions of the equation $\xi = \alpha x - \beta y$ that can be obtained from the particular solution $(\overline{x}, \overline{y})$ as follows:

$$\begin{aligned} x &= \overline{x} - \frac{\beta}{\gamma}\, t, \\ y &= \overline{y} - \frac{\alpha}{\gamma}\, t, \qquad t \in \mathbf{Z}. \end{aligned}$$

At this point, it is easy to find $\hat{t} \in \mathbf{Z}$, giving the solution $(\hat{x}, \hat{y})$, such that both $\alpha\hat{x}$ and $\beta\hat{y}$ are greater than n^2, and hence belong to $\mathcal{A}$ and $\mathcal{B}$, respectively. In other words, we have found two integers $h = \alpha\hat{x} \in \mathcal{A}$ and $k = \beta\hat{y} \in \mathcal{B}$ such that $\xi = h - k$. This together with (2.2) shows that $\xi \in \mathcal{X}$ and completes the proof. □

We end this subsection by showing a further property of solutions of Diophantine equations in natural numbers.

LEMMA 2.6. *Let $a_1, \ldots, a_s$ be positive integers less than or equal to n, and let the integer $z \geq 0$. If the equation $a_1x_1 + a_2x_2 + \cdots + a_sx_s = z$ has a solution in natural numbers, then it also has a solution in natural numbers satisfying*

$$a_2x_2 + \cdots + a_sx_s \leq n^2.$$

Proof. We prove a stronger result, namely, that there exists a solution $x_1, \ldots, x_s$ satisfying $x_2 + \cdots + x_s \leq a_1$. Suppose that $(\overline{x}_1, \overline{x}_2, \ldots, \overline{x}_s)$ is a solution in natural numbers, with $\mu = \overline{x}_2 + \cdots + \overline{x}_s > a_1$. Let us form the sequence

$$\underbrace{a_2, a_2, \ldots, a_2}_{\overline{x}_2-\text{times}}\ ,\ \underbrace{a_3, a_3, \ldots, a_3}_{\overline{x}_3-\text{times}}\ ,\ \cdots\ ,\ \underbrace{a_s, a_s, \ldots, a_s}_{\overline{x}_s-\text{times}}\ . \tag{2.3}$$

Now, for $i = 1, \ldots, \mu$, let S_i be the sum of the first i members in (2.3). Since the length of the sequence $S_1, \ldots, S_\mu$ exceeds a_1, there must exists a pair of indices $1 \leq i < j \leq \mu$ satisfying $S_i \equiv S_j \pmod{a_1}$, or equivalently, such that $S_j - S_i$ is a multiple of a_1. This enables us to remove from (2.3) the elements forming $S_j - S_i$, consequently augmenting the value of $\overline{x}_1$ by $(S_j - S_i)/a_1$. By iterating this process, we get the claimed result. □

2.4. Paths in directed graphs. Here, we recall some basic notions and prove—by the mathematics above developed—some results on directed graphs. Our interest in this topic is due to the fact that, as we will see in the next section, computation paths of unary automata can be represented as paths in suitable directed graphs. Few elementary notions of graph theory are required and summarized below. For more details, we refer the reader to any of the standard text on graph theory such as, e.g., [1].

Let $G = (V, E)$ be a directed graph or *digraph*, with V the set of vertices, and $E \subseteq V \times V$ the set of arcs. An *oriented path* $\mathcal{P}$ in G is a sequence of vertices $\mathcal{P} = v_0, v_1, \ldots, v_n$ where, for $i = 1, \ldots, n$, $(v_{i-1}, v_i) \in E$. Since we will be dealing with digraphs only, the attribute "oriented" will always be intended. The *length* of $\mathcal{P}$ is the number of arcs $\mathcal{P}$ consists of, and is denoted by $|\mathcal{P}|$. The path $\mathcal{P}$ is a *cycle (*or *closed path)* if $v_0 = v_n$. We call *elementary* (or, sometimes, *simple*) any cycle in which no vertex is encountered more than once (except, of course, the initial vertex which is also the terminal vertex).

A *subgraph* of G is any digraph $G' = (V', E')$ satisfying $V' \subseteq V$ and $E' \subseteq E \cap (V' \times V')$. The subgraph G' is said to be *induced by* V', whenever $E' = E \cap (V' \times V')$. The digraph G is *strongly connected* if there exists a path between any two vertices. A *strongly connected component* of G is a subgraph which is strongly connected and not contained in any other strongly connected subgraph.

The following two results concern length and structure of paths in digraphs.

LEMMA 2.7. *Let G be a digraph with n vertices, and let $\mathcal{P}$ be a path of length x which*

(a) *starts from vertex v_1 and ends in vertex v_2,*
(b) *visits* all *vertices in G.*

Then, there also exists a path $\mathcal{P}_0$ of length x_0 which satisfies (a) *and* (b) *and such that*

(c) $x_0 \leq n^2$,

(d) *there exist integers $x_1, x_2, \ldots, x_s \geq 0$ such that $x = x_0 + a_1x_1 + a_2x_2 + \cdots + a_sx_s$, where $a_1, a_2, \ldots, a_s$ are the lengths of the elementary cycles in G.*

Proof. We can factorize the sequence of the $x+1$ vertices visited along $\mathcal{P}$ as a concatenation of n subsequences of vertices $\sigma_1, \sigma_2, \ldots, \sigma_n$, each one starting from a *different* vertex of G. This can be done since $\mathcal{P}$ visits *all* vertices, as required in (b).

Suppose that $x > n^2$. Then, there exists $1 \leq j \leq n$ such that σ_j contains more than n vertices, i.e., $\sigma_j = v_{j_1}v_{j_2}\ldots v_{j_m}$ with $m > n$. A simple pigeonhole argument shows that there must exist $1 \leq h < k \leq m$ such that $v_{j_h} = v_{j_k}$. This clearly means that, when passing through σ_j, our path $\mathcal{P}$ presents a cycle beginning and ending in $v_{j_h} = v_{j_k}$. Call σ'_j the sequence obtained from σ_j after the elimination of the part corresponding to such a cycle, namely, $\sigma'_j = v_{j_1}v_{j_2}\ldots v_{j_h}v_{j_{k+1}}\ldots v_{j_m}$. The sequence $\sigma_1\cdots\sigma_{j-1}\,\sigma'_j\,\sigma_{j+1}\cdots\sigma_n$ defines a new path which satisfies (a) and (b) and whose length is strictly less than the length of $\mathcal{P}$. By iterating such a path-compression, we eventually obtain a path $\mathcal{P}_0$ of length x_0 satisfying (a), (b), and (c).

In conclusion, it is not hard to see that the whole process can be carried on by eliminating *elementary cycles*. Thus, one can easily express the length x of $\mathcal{P}$ as in (d), where, for $i = 1, \ldots, s$, x_i denotes the number of times elementary cycles of length a_i have been deleted. □

THEOREM 2.8. *Let G be a digraph with n vertices, and let $\mathcal{P}$ be a path from vertex v_1 to vertex v_2. Then, there also exists a path $\mathcal{P}'$ from v_1 to v_2 with the same length as $\mathcal{P}$ which consists of*

(a) *an initial path $\mathcal{P}_1$,*

(b) *an elementary cycle $\mathcal{P}_2$, with $1 \leq |\mathcal{P}_2| \leq n$, which is repeated λ times,*

(c) *a final path $\mathcal{P}_3$,*

satisfying $|\mathcal{P}_1| + |\mathcal{P}_3| \leq 2n^2$. (Hence, notice that $|\mathcal{P}'| = |\mathcal{P}| = |\mathcal{P}_1| + \lambda\,|\mathcal{P}_2| + |\mathcal{P}_3|$.)

Proof. Without loss of generality, we suppose that $\mathcal{P}$ visits all vertices of G. Otherwise, we consider the subgraph of G induced by the set of vertices visited along $\mathcal{P}$. The new path $\mathcal{P}'$ can be obtained by suitably "reorganizing"—in the light of our result on Diophantine equations in Lemma 2.6—the way of $\mathcal{P}$ through its elementary cycles.

The first step amounts to eliminate some of the elementary cycles in $\mathcal{P}$ to obtain the path $\mathcal{P}_0$, as explained in Lemma 2.7. The total number of arcs eliminated in this step can be expressed as

$$|\mathcal{P}| - |\mathcal{P}_0| = a_1x_1 + a_2x_2 + \cdots + a_sx_s \tag{2.4}$$

for some integers $x_1, x_2, \ldots, x_s \geq 0$, as one may easily check from Lemma 2.7(d). Now, notice that the a_i's in (2.4) are positive integers not exceeding n, being lengths of elementary cycles in a digraph of n vertices. Hence, from Lemma 2.6, we know that there exist integers $\overline{x}_1, \overline{x}_2, \ldots, \overline{x}_s \geq 0$ which are solutions for (2.4), and such that

$$a_2\overline{x}_2 + \cdots + a_s\overline{x}_s \leq n^2. \tag{2.5}$$

All this leads us to construct the path $\mathcal{P}'$ by "padding" $\mathcal{P}_0$ with $\overline{x}_i$ consecutive repetitions of an elementary cycle of length a_i for $i = 1, \ldots, s$. Let $\mathcal{P}_2$ be an elementary cycle of length a_1, and let $\mathcal{P}_1$ ($\mathcal{P}_3$) be the part of $\mathcal{P}'$ which precedes (follows) the $\overline{x}_1$ consecutive repetitions of $\mathcal{P}_2$. Then, we have

- $1 \leq |\mathcal{P}_2| = a_1 \leq n$,

- $\lambda = \overline{x}_1$,
- $|\mathcal{P}_1| + |\mathcal{P}_3| = x_0 + a_2\overline{x}_2 + \cdots + a_s\overline{x}_s \leq 2n^2$, since $x_0 \leq n^2$, from Lemma 2.7(c), and by considering inequality (2.5). □

Let us now turn to *weighted digraphs*. In particular, we are interested in digraphs whose arcs have weights from the set $\{-1, 0, +1\}$. As usual, the *weight of a path* is the sum of weights of the arcs in the path. The following theorem states a periodicity property on cycle weights in strongly connected weighted digraphs.

THEOREM 2.9. *Let $G = (V, E)$ be a strongly connected weighted digraph with n vertices, and let $\{a_1, \ldots, a_s\}$ be the set of weights of all the elementary cycles in G, with $\alpha = \gcd(a_1, \ldots, a_s)$. For each $v \in V$, let $\mathcal{X}_v$ be the set of weights of all the cycles containing v. If there exists at least one $a_i > 0$ ($a_i < 0$), then the set $\mathcal{X}_v$ and the set of positive (negative) integral multiples of α coincide on the elements greater than $2n^2$ (less than $-2n^2$).*

Proof. Since each cycle in G is the sum of elementary cycles, $x \in \mathcal{X}_v$ implies that

$$x = a_1x_1 + \cdots + a_sx_s, \tag{2.6}$$

where, for $i = 1, \ldots, s$, the value $x_i \geq 0$ is the number of times an elementary cycle of weight a_i is passed through to form our cycle of weight x. For what we have recalled in subsection 2.3, this means that x must be an integral multiple of α.

Conversely, if x is an integral multiple of α, with $|x| > n^2$, then, by Lemma 2.5, it can be expressed as in (2.6) for some $x_1, \ldots, x_s \geq 0$. However, such a solution does not necessarily describe a cycle in G. To see this, consider the following example: Suppose we have a digraph with three elementary cycles of weights a_1, a_2, a_3, in which the first and the last cycle are connected *only by the cycle of weight a_2*. Any solution with $x_1, x_2, x_3 > 0$ certainly defines (at least) one cycle in such a digraph, where the ith elementary cycle is passed through x_i times, $i = 1, 2, 3$. On the other hand, a solution with $x_1, x_3 > 0$ *and* $x_2 = 0$ does not define any cycle at all, since we cannot traverse both the first and the third elementary cycle without entering (since, $x_2 = 0$) the second one.

To deal with such kind of connectivity problems, we consider a cycle $\mathcal{C}_0$ of weight x_0 which visits *all* vertices of G. From Lemma 2.7(c), $\mathcal{C}_0$ consists of at most n^2 arcs, and this clearly implies that $|x_0| \leq n^2$. Now, notice that x_0 can be expressed as in (2.6) for some $x_1, \ldots, x_s \geq 0$. Hence, $x_0 = k_0\alpha$ for some $k_0 \in \mathbf{Z}$.

Assume $x = k\alpha$ with $|x| > 2n^2$ for $k \in \mathbf{Z}$, and set $z = x - x_0 = (k - k_0)\alpha$. We have that $|z| = |x - x_0| \geq |x| - |x_0| > n^2$. Since $|a_i| \leq n$ for $i = 1, \ldots, s$, from Lemma 2.5 we get $z = a_1z_1 + \cdots + a_sz_s$, for some $z_1, \ldots, z_s \geq 0$. This enables us to "pad" the cycle $\mathcal{C}_0$ with an amount of z_i elementary cycles of weight a_i for $i = 1, \ldots, s$, in order to obtain a cycle of weight x in G. □

3. Computation paths of 2nfa's and simulation by 1dfa's. In this section, we focus on unary 2nfa's. By using results in subsection 2.4, we show that any unary language $\mathcal{L}$ accepted by a 2nfa with n states *forms an ultimately periodic set of period* $\ell = O(\mathrm{e}^{\sqrt{n \ln n}})$. More precisely, we prove that, for any integer $m > 5n^2$, *the string* 1^m *belongs to* $\mathcal{L}$ *if and only if* $1^{m+\ell}$ *belongs to* $\mathcal{L}$. This immediately leads to a $O(\mathrm{e}^{\sqrt{n \ln n}})$ simulation of unary n-state 2nfa's by 1dfa's. The optimality of such a simulation is a direct consequence of a result in [8].

To state our results, we need some tools to examine computations on unary inputs. Specifically, we refer to three properties that Geffert proved in [11] for two-way nondeterministic Turing machines accepting unary languages in sublogarithmic space. We

are now going to reformulate such properties within the realm of unary 2nfa's. Their validity in this new form comes straightforwardly by observing that finite automata can be clearly seen as Turing machines working in sublogarithmic space. Moreover, we improve the last of these three results. Such an improvement will be particularly useful in the next section where we show that, by just squaring the number of states, it is possible to consider 2nfa's with a very particular structure.

From now on, we will always refer to a *unary 2nfa A with n states*.

The first lemma [11, Lemma 3] says that each computation path of A beginning and ending at the same input square and visiting neither of the endmarkers (*U-Turn*) can be replaced by an equivalent computation path not moving the input head "too far," precisely, no more than n^2 positions to the right (or left).

LEMMA 3.1 (U-Turn). *Suppose there exists a computation path of A on input 1^m where*

(a) *the first state is q_1 with the input head at a position i,*
(b) *the last state is q_2 with the input head at the same position,*
(c) *the input head never moves to the left (right) of the ith position, nor does it visit the right (left) endmarker.*

Then, there also exists a computation path on input 1^m satisfying (a), (b), (c) *and where the input head never moves farther than n^2 positions to the right (left) of the ith position.*

The second lemma [11, Lemma 4] states that we can freely "shift" any computation path that never visits the endmarkers to any position of the input tape, provided that we are sufficiently far from the endmarkers. To our purposes, we give this lemma in a slightly different form. (Recall that the left and right endmarker occupy positions 0 and $m+1$, respectively, on the input tape.)

LEMMA 3.2 (position independence). *Suppose there exists a computation path Π of A on input 1^m beginning in the state q_1 with the input head at a position i, ending in the state q_2 with the input head at a position j, and such that*

(a) $1 \le i < j \le m+1$,
(b) *the left endmarker is never visited along Π,*
(c) *the jth position is reached at the last move only.*

Then, there also exists a computation path beginning in the state q_1 with the input head at a position $i+h$, with $h \in \mathbf{Z}$, ending in the state q_2 with the input head at the position $j+h$ which is reached at the last move only, and satisfying (b), *provided that*

$$n^2 \le i+h < j+h \le m+1.$$

A similar result can be stated for computation paths moving toward left, i.e., with the initial (final) position i (j) satisfying $0 \le j < i \le m$.

Proof. Without loss of generality, from Lemma 3.1, we can assume that Π never visits more than n^2 positions to the left of the ith position. Since the input is unary, we can shift Π within the position bounds stated in the theorem. Notice that the right endmarker can possibly be reached only at the end of the path. □

The third lemma [11, Theorem 1] defines the structure of the computation paths that traverse the entire input, from one endmarker to the other. We provide a new proof of this lemma for unary 2nfa's which is different from the one given in [11]. Our proof makes use of the results on the structure of paths in digraphs in subsection 2.4. As a consequence, we obtain a quadratic estimation of the parameters s_1 and s_2 involved in the lemma; this is to be compared with the $O(n^4)$ upper bound in

[11].[2] Though not essential in the unary 2nfa's by 1dfa's simulation, this improvement will be crucial in the quadratic simulation of unary 2nfa's by 2nfa's having both head reversals and nondeterminism at the endmarkers only (Theorem 4.1).

In what follows, by *loop of length* ℓ, we mean a computation path of the automaton A beginning in a state p with the input head at a position i, and ending in the same state with the input head at the position $i + \ell$.

LEMMA 3.3 (dominant loop). *Suppose the input* 1^m *is traversed from left to right by a computation path* Π *of* A *beginning at the left endmarker in the state* q_1*, ending at the right endmarker in the state* q_2*, and such that the endmarkers are visited only in states* q_1 *and* q_2*. Then,* 1^m *can also be traversed by a computation path beginning in the state* q_1*, ending in the state* q_2*, and in which* A

(a) *having traversed the left endmarker and* s_1 *positions,*
(b) *gets into a loop (called* dominant loop*) of length* ℓ*, which starts from a state* p *and is repeated* λ *times,*
(c) *then traverses the remaining* s_2 *input squares, and finally gets the right endmarker,*

for some s_1*,* s_2*,* ℓ *satisfying*

$$1 \le \ell \le n,$$
$$s_1 + s_2 \le 3n^2.$$

Notice that $m = s_1 + \lambda\ell + s_2$*. The same result holds for computation paths traversing inputs from right to left.*

Proof. If $m \le 3n^2$, the result follows trivially. Hence, suppose $m > 3n^2$. Define the digraph $G = (Q, E)$, where Q is the set of states of A, and $(p, q) \in E$ if and only if there exists a computation path of A which starts from p with the input head at a position $n^2 \le i \le m$, reaches q with the input head at the position $i + 1$, and never visits either the endmarkers or the $(i + 1)$th input square in the intermediate steps.

Since we are sufficiently (namely, more than n^2 positions) far from the left endmarker, and since computations take place on unary inputs, Lemma 3.2 implies that the digraph G does not depend on the input length. For the same reasons, it is also easy to see that for any path in G of length d starting from the state q' and ending in the state q'', there exists a computation path of A on input 1^m beginning in q' with the input head at a position i, ending in q'' with the input head at the position $i + d$, provided that $n^2 \le i < i + d \le m + 1$.

Conversely, for any computation path of A on input 1^m beginning in the state q' with the input head at a position i, ending in q'' with the input head at the position j, and satisfying conditions (a), (b), and (c) in Lemma 3.2, one can easily find a path in G of length $j - i$ joining q' to q''.

Let us subdivide the computation path Π into

- Π_{in} which is the part of Π ending when the (n^2+1)th input square is reached for the first time,
- Π_{fin} which is the remaining part of Π.

For the correspondence between paths in G and computations in A above described, we can associate with Π_{fin} a path $\mathcal{P}$ in G. As shown in Theorem 2.8, $\mathcal{P}$ can be rearranged into a path $\mathcal{P}'$ of the same length as $\mathcal{P}$ made of

[2] At the time the conference version of this paper [20] was completed, the authors learned from Viliam Geffert [13] that he too had obtained, by different arguments, a quadratic upper bound on s_1 and s_2.

- an initial path $\mathcal{P}_1$,
- an elementary cycle $\mathcal{P}_2$, with $1 \leq |\mathcal{P}_2| \leq n$, which is repeated λ times,
- a final path $\mathcal{P}_3$,

satisfying $|\mathcal{P}_1| + |\mathcal{P}_3| \leq 2n^2$. Now, again for the correspondence paths-computations, we can associate with $\mathcal{P}_1$, $\mathcal{P}_2$, and $\mathcal{P}_3$, three computation paths Π_1, Π_2, and Π_3, respectively; in particular, Π_2 is a loop—the *dominant loop*—of length $1 \leq \ell \leq n$.

In conclusion, the input 1^m can be traversed by a computation path of A which starts with Π_{in}, continues with Π_1 which is followed by λ consecutive repetitions of the loop Π_2, and ends with Π_3. Let s be the total amount of cells visited outside the dominant loop (without counting the endmarkers), i.e., during Π_{in}, Π_1, and Π_3. It is easy to see that

$$s \leq n^2 + |\mathcal{P}_1| + |\mathcal{P}_3| \leq 3n^2. \qquad \square$$

From the previous lemma, we learn that loops play an important role in the structure of computations of unary 2nfa's. By using Theorem 2.9, we can show that possible loop lengths follow a very regular pattern which is directly predictable from the structure of unary 2nfa's themselves.

To this purpose, let us consider the weighted digraph $\mathcal{A}$ consisting of the digraph representing the transition diagram of our automaton A after removing transitions on the endmarkers, and in which we set weights $+1$, -1, or 0 to arcs depending on whether they represent transitions where the input head is moved right, left, or kept stationary, respectively. It is straightforward that *any cycle of weight ℓ in $\mathcal{A}$ represents a computation loop of length ℓ in the automaton A*. Hence, by taking into account Theorem 2.9, we get the following lemma.

LEMMA 3.4. *Let p be any given state of A, and let α be the greatest common divisor of weights of the elementary cycles in the strongly connected component of $\mathcal{A}$ containing p. Let $\mathcal{X}_p^+$ ($\mathcal{X}_p^-$) be the set of integers $x > 2n^2$ ($x < -2n^2$) such that the automaton A, starting from the state p with the input head at the ith input square, reaches the $(i+x)$th input square in the same state p, without visiting the endmarkers. If $\mathcal{X}_p^+ \neq \emptyset$ ($\mathcal{X}_p^- \neq \emptyset$), then it is exactly the set of all integral multiples of α greater than $2n^2$ (less than $-2n^2$).*

The following result is crucial in order to obtain our simulations.

THEOREM 3.5. *There exists a set of positive integers $\{\ell_1, \ldots, \ell_r\} \subseteq \{1, \ldots, n\}$ satisfying $\ell_1 + \cdots + \ell_r \leq n$, such that, for any $m \geq n$, if the input 1^m can be traversed from left to right by a computation path of A beginning in the state q_1, ending in the state q_2, and where the endmarkers are visited on the first and last move only, then there exists an index $i \in \{1, \ldots, r\}$ such that, for any $\mu > \frac{5n^2 - m}{\ell_i}$, there is also a computation path from q_1 to q_2 which traverses from left to right the input*

$$1^{m+\mu\ell_i}.$$

The same result holds for computation paths traversing inputs from right to left.

Proof. Again, consider the weighted digraph $\mathcal{A}$ associated with the automaton A. Suppose $\mathcal{A}$ has r strongly connected components and, for $i = 1, \ldots, r$, denote by ℓ_i the greatest common divisor of weights of the elementary cycles in the ith strongly connected component. Clearly, ℓ_i cannot exceed n, and since strongly connected components partition the set of vertices of $\mathcal{A}$, we get $\ell_1 + \cdots + \ell_r \leq n$.

Given the left-to-right traversing of input 1^m, let s_1, s_2, ℓ, λ, p, be as in Lemma 3.3, so to have the "decomposition" $m = s_1 + \lambda\ell + s_2$, and p being the state the dominant

loop starts from. Let us suppose that p is contained in the ith strongly connected component of $\mathcal{A}$. By Lemma 3.4, the set of integers $x > 2n^2$, for which there exists a path from p to itself visiting neither of the endmarkers and moving the input head x positions right, is exactly the set of the multiples of ℓ_i greater than $2n^2$. This leads us to conclude that, for integers $\eta > \frac{2n^2}{\ell_i}$, there exists a computation from the state q_1 to the state q_2 that completely traverses inputs $1^{s_1+\eta\ell_i+s_2}$, and visits the endmarkers on the first and last move only. It is enough, in fact, to use the original traverse of 1^m, and substitute the λ repetitions of the dominant loop with a single loop of length $\eta\ell_i$.

Now, by choosing $\mu > \frac{5n^2-m}{\ell_i}$ and suitably setting $\eta = \mu + \frac{\lambda\ell}{\ell_i}$, we get

$$\eta > \frac{5n^2-m}{\ell_i} + \frac{\lambda\ell}{\ell_i} = \frac{5n^2-(s_1+\lambda\ell+s_2)+\lambda\ell}{\ell_i} = \frac{5n^2-(s_1+s_2)}{\ell_i} \geq \frac{2n^2}{\ell_i},$$

where the last inequality is obtained by recalling that $s_1 + s_2 \leq 3n^2$, as stated in Lemma 3.3. Thus, for such η's, we obtain a complete traversing of inputs

$$1^{s_1+\eta\ell_i+s_2} = 1^{s_1+\left(\mu+\frac{\lambda\ell}{\ell_i}\right)\ell_i+s_2} = 1^{m+\mu\ell_i},$$

provided that $\mu > \frac{5n^2-m}{\ell_i}$. □

The technique used in the proof of the following theorem can be regarded as a refinement of the well-known $n \to n + n!$ pumping technique [18]. The result in Theorem 3.5 enables us to suitably pump and compress unary strings in order to show that unary languages accepted by n-state 2nfa's form ultimately periodic sets of period $O(\mathrm{e}^{\sqrt{n \ln n}})$.

THEOREM 3.6. *Let $\mathcal{L}$ be a unary language accepted by a n-state 2nfa. Then, there exists a constant $\ell = O(\mathrm{e}^{\sqrt{n \ln n}})$ such that, for any integer $m > 5n^2$,*

$$1^m \in \mathcal{L} \text{ if and only if } 1^{m+\ell} \in \mathcal{L}.$$

Proof. Given our 2nfa A, we let $\{\ell_1, \ldots, \ell_r\}$ be the set defined in Theorem 3.5 and let $\ell = \mathrm{lcm}(\ell_1, \ldots, \ell_r)$. Since we have observed that $\ell_1 + \cdots + \ell_r \leq n$, as a consequence of Lemma 2.1 we have that $\ell = O(\mathrm{e}^{\sqrt{n \ln n}})$.

Compression. Suppose $1^{m+\ell} \in \mathcal{L}$, and let $\mathcal{C}$ be an accepting computation path of A on such an input. Along $\mathcal{C}$, consider $r_0, r_1, \ldots, r_p$, the sequence of all states in which the input head scans either of the endmarkers. For $j = 1, \ldots, p$, the following two possibilities arise:

(i) In both r_{j-1} and r_j, the input head scans the same endmarker. By Lemma 3.1 (U-Turn), we can suppose that this part of the computation can be accomplished by moving no more than n^2 positions away from the endmarker, and hence it can take place on the input 1^m as well, m being greater than $5n^2$.

(i) In r_{j-1} the input head scans one of the two endmarkers, while in r_j it scans the other. By Theorem 3.5, for some $\ell_i \in \{\ell_1, \ldots, \ell_r\}$, we can replace the computation path from r_{j-1} to r_j on input $1^{m+\ell}$ with a computation path from r_{j-1} to r_j on input $1^{m+\ell+\mu\ell_i}$, provided that $\mu > \frac{5n^2-(m+\ell)}{\ell_i}$. Choose $\mu = \mu_i = -\frac{\ell}{\ell_i}$. Since $m > 5n^2$, it is easy to verify that $\mu_i > \frac{5n^2-(m+\ell)}{\ell_i}$. Thus, we get a computation path on the input $1^{m+\ell+\mu_i\ell_i} = 1^m$ which begins and ends in the states r_{j-1} and r_j, respectively, and visits the endmarkers on the first and last move only.

From (i) and (ii), it is easy to conclude that A accepts the input 1^m as well.

Pumping. The proof of the converse is similar: choose $\mu = \mu_i = \frac{\ell}{\ell_i}$ at point (ii). □

As an immediate consequence of Theorem 3.6 we are able to state our main result of this section.

COROLLARY 3.7. *Each unary n-state 2nfa can be simulated by a $O(e^{\sqrt{n \ln n}})$-state 1dfa.*

Proof. Given a unary n-state 2nfa accepting the language $\mathcal{L}$, the state diagram of an equivalent 1dfa consists of a path of no more than $5n^2$ states joined to a single elementary cycle involving $\ell = O(e^{\sqrt{n \ln n}})$ states. The path takes care of the nonperiodic part of $\mathcal{L}$ (i.e., strings of length not exceeding $5n^2$), while the cycle accounts for the periodic part (i.e., strings of length greater than $5n^2$). Final states are placed onto the path by inspecting membership of all strings of length not exceeding $5n^2$. Moreover, the property "$1^m \in \mathcal{L}$ if and only if $1^{m+\ell} \in \mathcal{L}$" stated in Theorem 3.6 allows us to set final states onto the cycle by just testing membership for strings $1^{5n^2+1}, 1^{5n^2+2}, \ldots, 1^{5n^2+\ell}$. □

It is possible to show that this simulation cost cannot be improved. Actually, a stronger result can be stated, which proves the optimality of all $O(e^{\sqrt{n \ln n}})$ bounds in Figure 1.1.

THEOREM 3.8 (see [8, Theorem 6.1]). *For any integer n, there exists a unary 2dfa with n states such that any equivalent 1nfa requires $\Omega(e^{\sqrt{n \ln n}})$ states.*

The trivial simulations of cost n in Figure 1.1 are optimal. In fact, for fixed $n > 0$, consider the single word language $\mathcal{L}_n = \{1^{n-1}\}$. Such a language is clearly accepted by a (minimum) 1dfa with n states. On the other hand, any 2nfa for $\mathcal{L}_n$ cannot have less than n states [3].

For the optimality of the quadratic simulation of unary 1nfa's by 2dfa's we refer the reader to [8, Theorem 6.2]. Thus, to complete Figure 1.1, we are left to examine unary one-way alternation versus determinism and nondeterminism. This is precisely the subject matter of section 5. Before that, we briefly show how to use the results so far proved to confine *both reversals and nondeterminism* to the endmarkers on unary 2nfa's by just paying a quadratic increase of the number of states.

4. Bringing reversals and nondeterminism at the endmarkers.

THEOREM 4.1. *Each unary n-state 2nfa A can be simulated by a $O(n^2)$-state 2nfa A' which performs both input head reversals and nondeterministic choices only when the input head scans the endmarkers.*

Proof. Without loss of generality, and by adding one more state, we can assume that A accepts with the input head parked on the left endmarker. We informally describe how A' simulates the computation of A on a given input 1^m.

In a first scan, A' deterministically checks whether $m \leq 5n^2$ and, if so, accepts if and only if $1^m \in \mathcal{L}(A)$. We can assume that at the end of this phase, which clearly requires $O(n^2)$ states, the input head is still positioned on the left endmarker.

Otherwise, if $m > 5n^2$, the second part of the simulation starts from the initial state of A. We briefly explain what happens when A' simulates the behavior of A from a given state q_1 and with the input head scanning the left endmarker. (A symmetrical simulation for computations of A from the right endmarker can be trivially stated.) A' nondeterministically chooses one of the following operations:

(a) simulation of a "U-Turn,"
(b) simulation of a left-to-right traversal of the input.

Since $m > 5n^2$, by Lemma 3.1 the simulation (a) can take place in one step, regardless the length of the input. Hence, each U-Turn from the left endmarker can be "embedded" in the transition function of A'. More precisely, it can be replaced by one stationary move. This part of the simulation does not require new states.

Now, we describe how to perform simulation (b). (In the following, a mod b denotes the remainder of the integer division of a by b.) Let $\{\ell_1, \ldots, \ell_r\}$ be the set introduced in Theorem 3.5 with respect to the automaton A. Given a ℓ_i, with $i \in \{1, \ldots, r\}$, two integers m', m'' satisfying

- $5n^2 < m' < m''$, and
- $m' \bmod \ell_i = \varrho = m'' \bmod \ell_i$,

and two states q_1, q_2 of A, it is not hard to prove that

> there exists a computation path of A from q_1 to q_2, scanning the input $1^{m'}$ from left to right, and touching the endmarkers on the first and last moves only if and only if there exists an analogous computation path on $1^{m''}$.

In fact, write $m' = h'\ell_i + \varrho$, $m'' = h''\ell_i + \varrho$ for some $0 < h' \leq h''$ and $0 \leq \varrho < \ell_i$. This yields $m' = m'' + \mu\ell_i$, where $\mu = h' - h'' = \frac{m' - \varrho - m'' + \varrho}{\ell_i} > \frac{5n^2 - m''}{\ell_i}$, by recalling that $m' > 5n^2$. Hence, by Theorem 3.5, from the left-to-right scan of $1^{m''}$, we can obtain a similar computation path for $1^{m'}$. The proof of the converse is similar. As a consequence of this observation, we get that, given the starting state q_1 of A and the input length $m > 5n^2$, the set of states reachable by A after traversing from left to right the input 1^m depends only on q_1 and on the set of pairs

$$\{(\ell_i\,,\ m \bmod \ell_i) \mid i = 1, \ldots, r\}.$$

Thus, in simulating a left-to-right traversal of A on 1^m starting from the state q_1 with the input head on the left endmarker, A' performs the following steps:

(i) nondeterministically selects an ℓ_i, with $i \in \{1, \ldots, r\}$,
(ii) scans the input tape, counting the input length m modulo ℓ_i,
(iii) when the input head reaches the right endmarker, nondeterministically selects one of the states associated with the starting state q_1 and the pair $(\ell_i\,,\ m \bmod \ell_i)$.

A' is easily seen to have both input head reversals and nondeterministic choices at the endmarkers only. Furthermore, during such a simulation from q_1, A' must remember the chosen ℓ_i, and the value m mod ℓ_i; this information can be clearly maintained in $\ell_1 + \cdots + \ell_s$ many states. Globally, $\ell_1 + \cdots + \ell_s$ more states are needed for each possible (starting) state, and since $\ell_1 + \cdots + \ell_s \leq n$, as observed in Theorem 3.5, we conclude that this part of the simulation requires $O(n^2)$ states. This completes the proof. □

5. Some remarks on 1afa's versus other models. We end the paper by briefly discussing the cost of simulating unary 1afa's with deterministic and nondeterministic automata, and the cost of the converse simulations. We begin with the simulations between 1afa's and 1dfa's:

- In [7, section 5.1], it is shown that for any n-state 1afa accepting a language $\mathcal{L}$ on an arbitrary alphabet there exists a 2^n-state 1dfa accepting $\mathcal{L}^R = \{x^R \mid x \in \mathcal{L}\}$, where x^R denotes the reversal of the string x. Such a result is easily seen to directly define the cost of simulating unary 1afa's with 1dfa's, since $x = x^R$ for unary strings.
- Conversely, in [17, Theorem 1] it is shown that for any n-state 1dfa accepting $\mathcal{L}$ there exists a $\lceil \log n \rceil$-state 1afa for $\mathcal{L}^R$.

This leads to the following equivalence which is basically contained in Corollary 1 and Corollary 2 of [17].

THEOREM 5.1. *The class of unary languages accepted by n-state 1afa's is exactly the class of unary languages accepted by 2^n-state 1dfa's.*

As an immediate consequence, one also gets the following corollary.

COROLLARY 5.2. *Let $\mathcal{L}$ be a unary language for which the minimum 1dfa has n states. Then, every 1afa accepting $\mathcal{L}$ requires at least $\lceil \log n \rceil$ states.*

The simulation costs between unary 1afa's and 1dfa's above stated are optimal, as pointed out in the following theorem.

THEOREM 5.3. *For any integer n, there exists a unary n-state 1afa (1dfa) such that each equivalent 1dfa (1afa) requires not less than 2^n ($\lceil \log n \rceil$) states.*

Proof.

1afa by 1dfa. We again refer to the single word language $\mathcal{L}_{2^n} = \{1^{2^n-1}\}$ considered at the end of section 3. Such a language can be accepted by a 1dfa with 2^n states and hence, according to Corollary 5.2, by a 1afa with n states. On the other hand, any 2nfa for $\mathcal{L}_{2^n}$ requires not less than 2^n states [3].

1dfa by 1afa. The minimum 1dfa for $\mathcal{L}_n = \{1^{n-1}\}$ has n state. Thus, the results follows from Corollary 5.2. □

Yet, we have the following theorem.

THEOREM 5.4. *Each unary n-state 1afa can be simulated by a 2^n-state 1nfa, 2dfa, or 2nfa. Such a simulation cost is tight in all the three cases.*

Proof. It is enough to recall that simulating unary 1afa's with 1dfa's costs 2^n states, and that 1afa's are exponentially more succinct than 2nfa's (Theorem 5.3 [*1afa by 1dfa*]). □

Thus, as already pointed out in [3], Theorem 5.4 says that 1afa's and 2dfa's are not polynomially equivalent regarding their state complexity, as conjectured in [8].

Our results in section 3 allow us to show that simulating 2nfa's (and hence 1nfa's and 2dfa's) by 1afa's takes a sublinear amount of states.

THEOREM 5.5. *Each unary 2nfa, 1nfa, and 2dfa with n states can be simulated by a $O(\sqrt{n \ln n})$-state 1afa.*

Proof. Clearly, it suffices to show the bound for 2nfa's. By Corollary 3.7, a unary n-state 2nfa can be simulated by a $O(e^{\sqrt{n \ln n}})$-state 1dfa which in turn, by Theorem 5.1, can be simulated by a 1afa with $O(\sqrt{n \ln n})$ states. □

The simulations by 1afa's in Theorem 5.5 are optimal.

THEOREM 5.6. *For any integer n, there exists a unary 1nfa and a unary 2dfa with n states such that each equivalent 1afa requires $\Omega(\sqrt{n \ln n})$ states.*

Proof. By [8, Theorem 4.5], for any n there exists a unary n-state 1nfa A such that each 1dfa recognizing $\mathcal{L}(A)$ requires $\Omega(e^{\sqrt{n \ln n}})$ states. By Corollary 5.2, this implies that each 1afa for $\mathcal{L}(A)$ must have $\Omega(\sqrt{n \ln n})$ states. Furthermore, the language $\mathcal{L}(A)$ is accepted even by a unary n-state 2dfa [8, Theorem 5.2]. □

With these results, we complete proving the results summarized in Figure 1.1, section 1.

Acknowledgments. The authors wish to thank anonymous referees for helpful comments and remarks. Moreover, the authors kindly acknowledge Viliam Geffert for stimulating discussions.

REFERENCES

[1] C. BERGE, *Graphs and Hypergraphs*, North-Holland, Amsterdam, 1985.

[2] P. Berman and A. Lingas, *On the Complexity of Regular Languages in Terms of Finite Automata*, Tech. Report 304, Polish Academy of Sciences, 1977.
[3] J.-C. Birget, *State-complexity of finite-state devices, state compressibility and incompressibility*, Math. Systems Theory, 26 (1993), pp. 237–269.
[4] A. Brauer, *On a problem of partitions*, Amer. J. Math., 64 (1942), pp. 299–312.
[5] A. Brauer and J.E. Shockley, *On a problem of Frobenius*, J. Reine Angew. Math., 211 (1962), pp. 215–220.
[6] J.A. Brzozowski and E. Leiss, *On equations for regular languages, finite automata, and sequential networks*, Theoret. Comput. Sci., 10 (1980), pp. 19–35.
[7] A. Chandra, D. Kozen, and L. Stockmeyer, *Alternation*, J. ACM, 28 (1981), pp. 114–133.
[8] M. Chrobak, *Finite automata and unary languages*, Theoret. Comput. Sci., 47 (1986), pp. 149–158.
[9] P. Erdős and R.L. Graham, *On a linear diophantine problem of Frobenius*, Acta Arith., 37 (1980), pp. 321–331.
[10] A. Fellah, H. Jürgensen, and S. Yu, *Constructions for alternating finite automata*, Internat. J. Comput. Math., 35 (1990), pp. 117–132.
[11] V. Geffert, *Nondeterministic computations in sublogarithmic space and space constructibility*, SIAM J. Comput., 20 (1991), pp. 484–498.
[12] V. Geffert, *Tally version of the Savitch and Immerman-Szelepcsényi theorems for sublogarithmic space*, SIAM J. Comput., 22 (1993), pp. 102–113.
[13] V. Geffert, *private communication*, 1997.
[14] J. Hopcroft and J. Ullman, *Introduction to Automata Theory, Languages, and Computation*, Addison–Wesley, Reading, MA, 1979.
[15] E. Landau, *Uber die maximalordung der permutationen gegebenen grades*, Archiv. der Math. und Phys., 3 (1903), pp. 92–103.
[16] E. Landau, *Handbuch der lehre von der verteilung der primzahlen.* I, Teubner, Leipzig, Berlin, 1909.
[17] E. Leiss, *Succinct representation of regular languages by boolean automata*, Theoret. Comput. Sci., 13 (1981), pp. 323–330.
[18] P. Lewis II, R. Stearns, and J. Hartmanis, *Memory bounds for recognition of context free and context sensitive languages*, in IEEE Conference Record on Switching Circuit Theory and Logical Design, 1965, pp. 191–202.
[19] U. Liubicz, *Bounds for the optimal determinization of nondeterministic autonomic automata*, Sibirskii Matemat. Journal, 2 (1964), pp. 337–355 (in Russian).
[20] C. Mereghetti and G. Pighizzini, *Optimal simulations between unary automata*, in 15th Annual Symposium on Theoretical Aspects of Computer Science, Lecture Notes in Comput. Sci. 1373, Springer-Verlag, Berlin, 1998, pp. 139–149.
[21] I. Niven and H.S. Zuckerman, *An Introduction to the Theory of Numbers*, John Wiley & Sons, New York, 1980.
[22] M. Rabin and D. Scott, *Finite automata and their decision problems*, IBM J. Res. Develop, 3 (1959), pp. 114–125. Also in E.F. Moore, *Sequential Machines*, Addison–Wesley, Reading, MA, 1964, pp. 63–91.
[23] W. Sakoda and M. Sipser, *Nondeterminism and the size of two–way finite automata*, in Proceedings 10th ACM Symposium on Theory of Computing, 1978, pp. 275–286.
[24] J.C. Shepherdson, *The reduction of two-way automata to one-way automata*, IBM J. Res. Develop, 3 (1959), pp. 198–200. Also in E.F. Moore, *Sequential Machines*, Addison–Wesley, Reading, MA, 1964, pp. 92–97.
[25] M. Sipser, *Lower bounds on the size of sweeping automata*, J. Comput. System Sci., 21 (1980), pp. 195–202.
[26] M. Szalay, *On the maximal order in S_n and S_n^**, Acta Arith., 37 (1980), pp. 321–331.
[27] A. Szepietowski, *Turing Machines with Sublogarithmic Space*, Lecture Notes in Comput. Sci. 843, Springer-Verlag, New York, 1994.
[28] P. Turan, *Combinatorics, partitions, group theory*, in Colloquio Internazionale sulle Teorie Combinatorie, B. Serge, ed., Accademia Nazionale dei Lincei, Rome, 1976, pp. 181–200.