International Journal of Foundations of Computer Science Vol. 13 No. 1 (2002) 145–159
© World Scientific Publishing Company

UNARY LANGUAGE OPERATIONS, STATE COMPLEXITY AND JACOBSTHAL'S FUNCTION

GIOVANNI PIGHIZZINI*
Dipartimento di Scienze dell'Informazione, Università degli Studi di Milano
via Comelico 39, 20135 Milano, Italy
pighizzi@dsi.unimi.it

JEFFREY SHALLIT†
Department of Computer Science, University of Waterloo
Waterloo, Ontario, Canada N2L 3G1
shallit@graceland.uwaterloo.ca

Received 16 December 2000
Revised 2 April 2001
Communicated by Giovanni Pighizzini

ABSTRACT

In this paper we give the cost, in terms of states, of some basic operations (union, intersection, concatenation, and Kleene star) on regular languages in the unary case (where the alphabet contains only one symbol). These costs are given by explicitly determining the number of states in the noncyclic and cyclic parts of the resulting automata. Furthermore, we prove that our bounds are optimal. We also present an interesting connection to Jacobsthal's function from number theory.

Keywords: Finite automata; formal languages; state complexity; number theory; unary languages.

1. Introduction

Finite automata are one of the first computational models presented in the literature and, certainly, one of the most extensively investigated. However, some problems concerning these simple models are still open and the investigation of some aspects of the finite automata world is only at the beginning. For instance, many complexity results for finite automata are given under the hypothesis that the input alphabet contains at least two symbols. A typical example is the simulation of an n–state nondeterministic finite automaton (NFA) by an equivalent deterministic finite automaton (DFA). The upper bound of 2^n states is provably optimal in the

*Research supported in part by MURST, under the project "Modelli di calcolo innovativi: metodi sintattici e combinatori".

†Research supported in part by a grant from NSERC.

worst case when the input alphabet contains at least two symbols [5, 13, 14], but can be reduced to about $e^{\sqrt{n \log n}}$ in the *unary* case, i.e., where the automata have an input alphabet consisting of a single letter [10, 11, 12, 4].

In this paper we study the *state complexity* of some simple operations on unary regular languages. We recall that the state complexity of a regular language L, written $\mathrm{sc}(L)$, is the number of states in the smallest deterministic finite automaton (DFA) accepting L. Several papers, such as [19], address the question of obtaining good upper bounds on the state complexity of basic operations such as intersection, union, concatenation and Kleene star of languages L' and L'', in terms of the state complexities of L' and L''.

The standard product construction for automata (e.g., [6, pp. 59–60]) easily shows that if $\mathrm{sc}(L') = n'$ and $\mathrm{sc}(L'') = n''$, then $\mathrm{sc}(L' \cap L'') \leq n'n''$. This upper bound can actually be attained for all $n', n'' \geq 1$ provided the underlying alphabet has at least two letters. Indeed, as Yu and Zhuang observe [18], we can let

$$L' = \{x \in (a+b)^* \ : \ |x|_a = n'\} \text{ and } L'' = \{x \in (a+b)^* \ : \ |x|_b = n''\},$$

where $|x|_c$ denotes the number of occurrences of the symbol c in the string x. A similar construction works for unary alphabets provided $\gcd(n', n'') = 1$. However, determining the best upper bound for unary languages when $\gcd(n', n'') > 1$ was stated as an open problem by Yu [17]. In Section 3 of this paper we solve this problem by proving tight bounds for the state complexity of the intersection and of the union of unary regular languages.

In [19], Yu, Zhuang and Salomaa proved that $\mathrm{sc}(L'L'') = n'2^{n''} - 2^{n''-1}$. This result cannot be improved if the input alphabet contains at least three symbols. However, in the unary case, the number of states which are sufficient to recognize $L'L''$ reduces to $n'n''$. This number is also necessary, in the worst case, when n' and n'' are relatively prime. In Section 4 we refine this analysis, obtaining tight bounds even when n' and n'' are not relatively prime. We also explicitly indicate the number of states in the cyclic and in the noncyclic parts of the resulting unary automata. We complete the scenario in Section 5 by presenting some considerations concerning the Kleene star operation. We point out that C. Nicaud [15] has recently investigated the *average* state complexity for the same operations on unary languages.

The estimations presented in the paper are related to an interesting function from number theory due to Jacobsthal. Section 6 is devoted to studying this function and its connections to our results.

2. Preliminary notions and results

In this section, we recall basic notions, notations and facts used in the paper.

Given two integers $a, b \geq 0$, we denote by $\gcd(a, b)$ and by $\mathrm{lcm}(a, b)$, their *greatest common divisor* and their *least common multiple*, respectively. The following result will be crucial in order to evaluate the number of states of unary automata:

Lemma 1 *Suppose a, b are positive integers. Then each number of the form $ax+by$, with $x, y \geq 0$, is a multiple of $\gcd(a, b)$. Furthermore, the largest multiple of $\gcd(a, b)$ that cannot be represented as $ax + by$, with $x, y \geq 0$, is $\mathrm{lcm}(a, b) - (a + b)$.*

Proof. It is well–known that each number $z = ax+by$ is a multiple of $g = \gcd(a, b)$. Let $a' = a/g$ and $b' = b/g$. Then $\gcd(a', b') = 1$. It is also well-known that the largest integer that cannot be represented as $a'x+b'y$, with $x, y \geq 0$, is $a'b' - (a'+b')$ (see, e.g., [3, 19]). Multiplying by g, we get that the largest multiple of g that cannot be written as $ax + by$, with $x, y \geq 0$, is $\mathrm{lcm}(a, b) - (a + b)$. □

Given an alphabet Σ, Σ^* denotes the set of strings on Σ. Given a language $L \subseteq \Sigma^*$, its complement, i.e., the set $\Sigma^* - L$, is denoted as L^c. A language L is said to be *unary* (or *tally*) whenever it can be built over a *single letter* alphabet. In this case, we let $L \subseteq a^*$.

The computational models we will consider in this paper are *one–way deterministic finite automata* (DFA) defined over a one-letter input alphabet $\Sigma = \{a\}$. A unary DFA will be denoted as a 5-tuple $A = (Q, \Sigma, \delta, q_0, F)$, with the usual meaning (see, e.g., [6]). By the pigeonhole principle, it is not difficult to observe that the transition graph of a unary DFA A, with n states, has a "tail" consisting of $\mu \geq 0$ states and a "cycle" of $\lambda \geq 1$ states. Furthermore, if the transition diagram is connected (as we may assume without loss of generality) then $n = \mu + \lambda$. See Figure 1. Following [4], we define the *size* of A to be the pair (λ, μ). Note that

$$L(A) = X \cup a^{\mu}Y = X \cup Z(a^{\lambda})^* \tag{1}$$

where $X = L(A) \cap \{a^x \ : \ 0 \leq x < \mu\}$ is the set of strings accepted by states in the tail, $Y = \{a^x \ : \ a^{x+\mu} \in L(A)\}$ is the set of the strings accepted by restricting A to the cycle, and $Z = \{a^x \ : \ \mu \leq x < \mu + \lambda\}$.

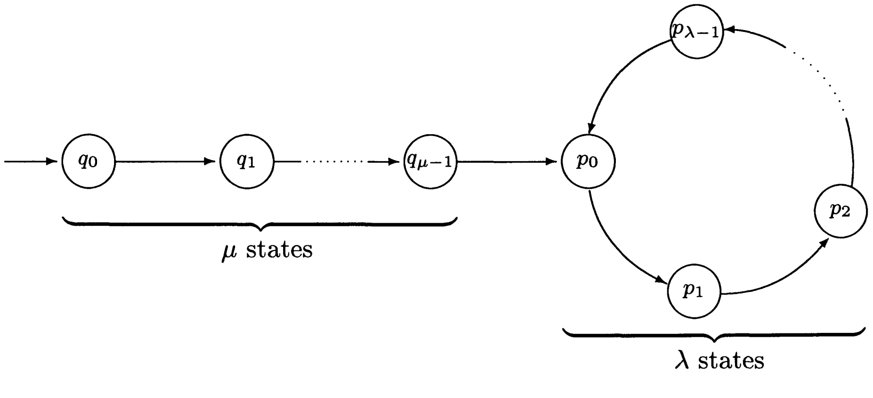

Fig. 1. A unary DFA of size (λ, μ) (final states not indicated).

Throughout the paper we will use the following conventions to denote any unary automaton $A = (Q, \Sigma, \delta, q_0, F)$ of size (λ, μ): the set of states is denoted as $Q = \{q_0, q_1, \ldots, q_{\mu-1}, p_0, p_1, \ldots, p_{\lambda-1}\}$, where $q_0, q_1, \ldots, q_{\mu-1}$ are the states occurring in the tail, and $p_0, p_1, \ldots, p_{\lambda-1}$ are the states of the cycle (with $q_0 = p_0$ when $\mu = 0$); then $\delta(q_i, a) = q_{i+1}$, for $i = 0, \ldots, \mu-2$, $\delta(q_{\mu-1}, a) = p_0$, and $\delta(p_i, a) = p_{(i+1) \text{ MOD } \lambda}$, for $i = 0, \ldots, \lambda-1$. Observing the form of unary DFA's, it is not difficult to conclude that unary regular languages correspond to *ultimately periodic sets* of integers:

Theorem 1 *A unary language L is regular if and only if there are two integers $\mu \geq 0$, $\lambda \geq 1$, such that for any $n \geq \mu$, $a^n \in L$ if and only if $a^{n+\lambda} \in L$.*

Note that, given a unary regular language L, the pair of integers (λ, μ) in Theorem 1, is the size of a DFA which accepts the regular language L. More precisely:

Theorem 2 *Given a unary regular language L and two integers $\lambda \geq 1$, $\mu \geq 0$, the following statements are equivalent:*

(i) L is accepted by a DFA of size (λ, μ);

(ii) for any $n \geq \mu$, $a^n \in L$ if and only if $a^{n+\lambda} \in L$.

A unary DFA is said to be *cyclic* if and only if its transition graph is a directed cycle. Languages accepted by cyclic automata are said to be *cyclic languages.* In other words, a unary language is cyclic if and only if it can be accepted by a DFA of size $(\lambda, 0)$, for some $\lambda \geq 1$. To emphasize the periodicity of L, we say that L is λ*-cyclic.*

In order to show the optimality of our constructions, we now present a condition which characterizes minimal unary DFA's [15, Lemma 1]:

Theorem 3 *A unary DFA $A = (Q, \Sigma, \delta, q_0, F)$ of size (λ, μ) is minimal if and only if both the following conditions are satisfied:*

(i) for any maximal proper divisor d of λ (i.e., $\lambda = \alpha \cdot d$, for some prime number $\alpha > 1$) there exists an integer h, with $0 \leq h < \lambda$, such that $p_h \in F$ if and only if $p_{(h+d) \text{ MOD } \lambda} \notin F$, i.e., $a^{\mu+h} \in L$ if and only if $a^{\mu+h+d} \notin L$;

(ii) $q_{\mu-1} \in F$ if and only if $p_{\lambda-1} \notin F$, i.e., $a^{\mu-1} \in L$ if and only if $a^{\mu+\lambda-1} \notin L$.

Note that condition (i) in Theorem 3 states that the cycle of A cannot be substituted with a shorter one, while condition (ii) states that it is impossible to "roll up" the last state of the tail on the cycle.

Corollary 1 *Given two integers $\mu \geq 0$ and $\lambda \geq 1$, let $L = a^{\mu+\lambda-1}(a^\lambda)^*$. Then, the size of the minimal DFA accepting L is (λ, μ).*

Proof. The language L is accepted by a DFA A of size (λ, μ), whose only final state is $p_{\lambda-1}$. Using Theorem 3, it is easy to prove that this is minimal. □

3. Intersection and union

In this section we evaluate the state complexity of the intersection of unary regular languages, by taking into account not only the total number of states, but also the sizes of the automata. Since any unary DFA accepting a language L can be easily transformed into a DFA of the same size accepting the complement of L, and $L' \cup L'' = (L'^c \cap L''^c)^c$, our results can be immediately extended to the union operation.

Theorem 4 *Let L' and L'' be two languages accepted by unary automata A' and A'' of size (λ', μ') and (λ'', μ''), respectively. The intersection (the union, respectively) of L' and L'' is accepted by a DFA of size* $(\mathrm{lcm}(\lambda', \lambda''), \max(\mu', \mu''))$.

Proof. Write $L' = X' \cup Z'(a^{\lambda'})^*$, as in Eq. (1), and $L'' = X'' \cup Z''(a^{\lambda''})^*$. We have $a^x \in L'$ iff $[x < \mu'$ implies $a^x \in X'$ and $x \geq \mu'$ implies there exists $a^y \in Z'$ such that $x \equiv y \pmod{\lambda'}]$. Similarly, $a^x \in L''$ iff $[x < \mu''$ implies $a^x \in X''$ and $x \geq \mu''$

implies there exists $a^y \in Z''$ such that $x \equiv y \pmod{\lambda''}$]. By the Chinese remainder theorem, there exists a set Z such that if $x \geq \max(\mu', \mu'')$, then $a^x \in L' \cap L''$ iff there exists $u \in Z$ such that $x \equiv u \pmod{\operatorname{lcm}(\lambda', \lambda'')}$. Hence we can accept $L' \cap L''$ using a cycle of $\operatorname{lcm}(\lambda', \lambda'')$ states and a tail of $\max(\mu', \mu'')$ states. □

Now, we prove that the construction given in Theorem 4 is optimal:

Theorem 5 *For any $\mu', \mu'' \geq 0$, $\lambda', \lambda'' \geq 1$, there exist two languages L' and L'' which are accepted by DFA's of size (μ', λ') and (μ'', λ''), respectively, such that the minimal DFA's accepting $L' \cap L''$ and $L' \cup L''$ have both size* $(\operatorname{lcm}(\lambda', \lambda''), \max(\mu', \mu''))$.

Proof. Let $l = \operatorname{lcm}(\lambda', \lambda'')$.

If $\mu' = \mu'' = 0$, then let $L' = (a^{\lambda'})^*$, $L'' = (a^{\lambda''})^*$. Then L' (respectively L'') may be accepted by a DFA A' (respectively A'') of size $(\lambda', 0)$, (respectively $(\lambda'', 0)$). Now $L' \cap L'' = (a^l)^*$. It is easy to see that $(a^l)^*$ may be accepted by a DFA with of size $(l, 0)$, with only a final state, and by the Theorem 3 this is best possible.

Otherwise, at least one of μ', μ'' is non-zero. Without loss of generality, assume $\mu' \geq \mu''$ and hence $\mu' > 0$. Define $L' = a^{\mu'+\lambda'-1}(a^{\lambda'})^*, L'' = a^r(a^{\lambda''})^*$ where $r := (\mu' - 1) \text{ MOD } \lambda''$. It is easy to see that L' (respectively, L'') can be accepted by a DFA A' (respectively, A'') of size (λ', μ') (respectively, (λ'', μ'')). (In fact, L'' can be accepted by a DFA A'' with an empty tail.)

We claim $L' \cap L'' = a^{\mu'+l-1}(a^l)^*$. To see this, note that $a^x \in L'$ iff $x = (\mu' + \lambda' - 1) + k\lambda'$ for some integer $k \geq 0$. Similarly, letting $\mu' - 1 = q\lambda'' + r$ with $0 \leq r < \lambda''$, we have $a^x \in L''$ iff $x = r + j\lambda''$ for some integer $j \geq 0$, i.e., iff $x = (\mu' - 1) + (j - q)\lambda''$. Thus $a^x \in L' \cap L''$ iff $\mu' + \lambda' - 1 + k\lambda' = (\mu' - 1) + (j - q)\lambda''$, which is the case iff $(k+1)\lambda' = (j - q)\lambda''$. But this equation has integer solutions iff $(k+1) = b\lambda''/g$ and $j - q = b\lambda'/g$ for some integer b, where $g = \gcd(\lambda', \lambda'')$. But $k \geq 0$ iff $b \geq 1$. Recalling that $\ell = \lambda'\lambda''/g$, it now follows that

$$L' \cap L'' = \{a^{(\mu'+\lambda'-1)+(b\lambda''/g-1)\lambda'} \ : \ b \geq 1\} = \{a^{\mu'-1+bl} \ : \ b \geq 1\} = a^{\mu'+l-1}(a^l)^*$$

as desired. By Corollary 1, the minimal DFA accepting $L' \cap L''$ has size (l, μ'). □

In Section 6, we will investigate the state complexity of the intersection and union of unary regular languages more deeply, by estimating the function:

$$F(n', n'') = \max_{\substack{1 \leq \lambda' \leq n' \\ 1 \leq \lambda'' \leq n''}} \left(\max(n' - \lambda', n'' - \lambda'') + \operatorname{lcm}(\lambda', \lambda'')\right).$$

4. Concatenation

In this section, we evaluate the optimal size of an automaton accepting the concatenation of the languages accepted by two given unary DFA's. Moreover, we are able to show that, for some subclasses of unary regular languages, this size can be further reduced.

Let us start by observing that two unary regular languages $\mathcal{L}', \mathcal{L}'' \subseteq a^*$, accepted by two unary automata A' and A'' of size (λ', μ') and (λ'', μ''), respectively, according to Equation (1), can be expressed in the form $\mathcal{L}' = X' \cup a^{\mu'}Y'$ and $\mathcal{L}'' = X'' \cup a^{\mu''}Y''$. Hence, the product $\mathcal{L}$ of $\mathcal{L}'$ and $\mathcal{L}''$ can be expressed as:

$$\mathcal{L} = L_0 \cup L_1 \cup L_2 \cup L_3, \tag{2}$$

where $L_0 = X'X''$, $L_1 = a^{\mu'}X''Y'$, $L_2 = a^{\mu''}X'Y''$, and $L_3 = a^{\mu'+\mu''}Y'Y''$.

In order to evaluate the size of a DFA accepting $\mathcal{L} = \mathcal{L}'\mathcal{L}''$, we first compute the sizes of DFA's accepting the languages L_0, L_1, L_2, and L_3. Subsequently, using our result concerning union of two unary regular languages (Theorem 4), we will get the size of a DFA accepting $\mathcal{L}$. We will also show that, in the general case, the size so obtained is optimal.

We observe that the languages L_0, L_1, L_2, and L_3 have very particular forms:

- L_0 is the concatenation of two finite languages;
- L_1 (L_2, respectively) is obtained by concatenating a singleton language with the product of a finite language and a cyclic language;
- L_3 is the concatenation of a singleton language with the product of two cyclic languages.

Hence, to get the sizes of automata accepting L_0, L_1, L_2, and L_3, in Theorems 6 and 8, we study the product of two languages L' and L'' in the following cases:

- one of the two languages L' and L'' is finite;
- both L' and L'' are cyclic.

We also prove the optimality of our results.

4.1. One Language is Finite

Theorem 6 *Given $\lambda', \lambda'' \geq 1$, $\mu', \mu'' \geq 0$, let L' and L'' be unary languages accepted by two DFA's A' and A'' of size (λ', μ') and (λ'', μ''), respectively. If L'' is finite then $L'L''$ is accepted by a DFA of size $(\lambda', \mu' + \mu'' - 1)$.*

Proof. If L'' is finite, then any state on the cycle of A'' should be nonfinal. This implies that the length of any string belonging to L'' is less than μ''. Thus, given an integer $n \geq \mu' + \mu'' - 1$ such that $a^n \in L'L''$, we can find two integers x and y such that $n = x + y$, $a^x \in L'$, $a^y \in L''$, $y < \mu''$, and $x \geq \mu'$. Since L' is accepted by a DFA of size (λ', μ'), this implies that $a^{x+\lambda'} \in L'$, and then $a^{n+\lambda'} \in L'L''$. By using similar arguments, we can also prove that, for any $n \geq \mu' + \mu'' + \lambda' - 1$, $a^{n+\lambda'} \in L'L''$ implies that $a^n \in L'L''$. Thus, in light of Theorem 2, we conclude that $L'L''$ is accepted by an automaton of size $(\lambda', \mu' + \mu'' - 1)$. □

The result presented in Theorem 6 is in fact optimal:

Theorem 7 *Let $\mu' \geq 0$, $\mu'', \lambda', \lambda'' \geq 1$ be integers. The languages*

$$L' = a^{\mu'+\lambda'-1}(a^{\lambda'})^* \text{ and } L'' = a^{\mu''-1},$$

are accepted by two DFA's of size (λ', μ') and (λ'', μ''), respectively. Moreover, the size of the minimal DFA accepting the concatenation of L' and L'' is $(\lambda', \mu'+\mu''-1)$.

Proof. It is enough to observe that $L'L'' = a^{\mu'+\mu''+\lambda'-2}(a^{\lambda'})^*$, and then apply the result presented in Corollary 1. □

4.2. Both Languages are Cyclic

Theorem 8 *Given $\lambda', \lambda'' \geq 1$, let L' and L'' be unary λ'-cyclic and λ''-cyclic languages, resp. Then $L'L''$ is accepted by a DFA of size* $(\gcd(\lambda', \lambda''), \operatorname{lcm}(\lambda', \lambda'') - 1)$.

Proof. According to Equation (1), we can write $L' = Z'(a^{\lambda'})^*$, $L'' = Z''(a^{\lambda''})^*$.

In order to prove that $L'L''$ is accepted by a DFA of size $(\gcd(\lambda', \lambda''), \operatorname{lcm}(\lambda', \lambda'') - 1)$, by Theorem 2, it is enough to show that, for any integer $z \geq \operatorname{lcm}(\lambda', \lambda'') - 1$, $a^z \in L'L''$ holds if and only if $a^{z+\gcd(\lambda',\lambda'')} \in L'L''$. To do this, consider an integer w such that $a^w \in L'L''$. Then $w = z' + z'' + \lambda' i + \lambda'' j$, where $a^{z'} \in Z'$, $a^{z''} \in Z''$, $i, j \geq 0$. If $w \geq \operatorname{lcm}(\lambda', \lambda'') - 1$, then $\lambda' i + \lambda'' j \geq \operatorname{lcm}(\lambda', \lambda'') - (\lambda' + \lambda'') + 1$. Observing that $\lambda' i + \lambda'' j$ is a multiple of $\gcd(\lambda', \lambda'')$, by Lemma 1 this implies that $\lambda' i + \lambda'' j + \gcd(\lambda', \lambda'')$ can be represented as $\lambda' x + \lambda'' y$, for some $x, y \geq 0$. Thus, $a^{w+\gcd(\lambda',\lambda'')} \in L'L''$.

Now, suppose that $w \geq \operatorname{lcm}(\lambda', \lambda'') + \gcd(\lambda', \lambda'') - 1$. Using an argument similar to that in the previous paragraph, it is easy to show that $\lambda' i + \lambda'' j - \gcd(\lambda', \lambda'') = \lambda' x + \lambda'' y$, for some $x, y \geq 0$. Given $z = w - \gcd(\lambda', \lambda'')$, this means that for $z \geq \operatorname{lcm}(\lambda', \lambda'') - 1$, $a^{z+\gcd(\lambda',\lambda'')} \in L'L''$ implies $a^z \in L'L''$.

This completes the proof. □

The result presented in Theorem 8 can be easily extended, using Theorem 6, as follows:

Corollary 2 *Let $L' = a^{\mu'}Y'$ and $L'' = a^{\mu''}Y''$ be two languages such that $Y', Y'' \subseteq a^*$ are λ'-cyclic and λ''-cyclic, respectively. Then L' and L'' are accepted by two automata of size (λ', μ') and (λ'', μ''), respectively, while $L = L'L''$ is accepted by an automaton of size (λ, μ), where $\lambda = \gcd(\lambda', \lambda'')$, and $\mu = \mu' + \mu'' + \operatorname{lcm}(\lambda', \lambda'') - 1$.*

The optimality of the results presented in Theorem 8 and in Corollary 2 is proved in the following result:

Theorem 9 *Let $\mu', \mu'' \geq 0$, $\lambda', \lambda'' \geq 1$ be integers. The languages*

$$L' = a^{\mu'+\lambda'-1}(a^{\lambda'})^* \text{ and } L'' = a^{\mu''+\lambda''-1}(a^{\lambda''})^*,$$

are accepted by two DFA's of size (λ', μ') and (λ'', μ''), respectively. Moreover, the size of the minimal DFA accepting $L'L''$ is $(\gcd(\lambda', \lambda''), \mu' + \mu'' + \operatorname{lcm}(\lambda', \lambda'') - 1)$.

Proof. First of all, we point out that, as observed in Corollary 1, the languages L' and L'' are accepted by two DFA's of size (λ', μ') and (λ'', μ''), respectively.

For the sake of simplicity, we prove the theorem in the case $\mu' = \mu'' = 0$. The extension to the general case is trivial.

By Theorem 8, there exists an automaton A of size $(\gcd(\lambda', \lambda''), \operatorname{lcm}(\lambda', \lambda'') - 1)$, which accepts $L = L'L''$. By using Theorem 3, we now show that A is minimal.

Given $n \geq 0$, the string a^n belongs to L if and only if there are two integers $x, y \geq 0$ such that $n = \lambda' + \lambda'' - 2 + \lambda' x + \lambda'' y$. By Lemma 1, all integers of the form $\lambda' x + \lambda'' y$, with $x, y \geq 0$, are multiple of $\gcd(\lambda', \lambda'')$. Hence, it is easy to conclude that condition (i) of Theorem 3 is satisfied.

Furthermore, given $k \geq 1$, the string $a^{\operatorname{lcm}(\lambda',\lambda'')-2+k\gcd(\lambda',\lambda'')}$ belongs to L if and only if $\lambda' x + \lambda'' y = \operatorname{lcm}(\lambda', \lambda'') + k\gcd(\lambda', \lambda'')$, for some $x, y \geq 1$, i.e., $\lambda' x + \lambda'' y =$

$\mathrm{lcm}(\lambda', \lambda'') - (\lambda' + \lambda'') + k\gcd(\lambda', \lambda'')$, for some $x, y \geq 0$. By Lemma 1, this implies that $a^{\mathrm{lcm}(\lambda',\lambda'')-2} \notin L$, while $a^{\mathrm{lcm}(\lambda',\lambda'')-2+\gcd(\lambda',\lambda'')} \in L$. Hence, condition (ii) of Theorem 3 is also satisfied. This permits us to conclude that A is minimal. □

4.3. The General Case

Using the results so far presented, we can now easily study the general case:

Theorem 10 *Given $\mu', \mu'' \geq 0$, $\lambda', \lambda'' \geq 1$, let $\mathcal{L}'$ and $\mathcal{L}''$ be unary languages accepted by two automata A' and A'' of size (λ', μ') and (λ'', μ''), respectively. Then, the concatenation of $\mathcal{L}'$ and $\mathcal{L}''$ is accepted by a DFA of size (λ, μ), where $\lambda = \mathrm{lcm}(\lambda', \lambda'')$ and $\mu = \mu' + \mu'' + \mathrm{lcm}(\lambda', \lambda'') - 1$.*

Proof. Using the notations introduced at the beginning of this section, we get the sizes of some automata accepting the languages L_0, L_1, L_2, and L_3 in Equality (2).

We note that the languages X', X'', Y', Y'', $a^{\mu'}$, and $a^{\mu''}$ can be accepted by DFA's of size $(1, \mu')$, $(1, \mu'')$, $(\lambda', 0)$, $(\lambda'', 0)$, $(1, \mu' + 1)$ and $(1, \mu'' + 1)$, respectively. Thus:

- L_0 is accepted by a DFA of size $(1, \mu' + \mu'' - 1)$ (Theorem 6);
- L_1 and L_2 are accepted by DFA's of size $(\lambda', \mu' + \mu'' + \lambda' - 1)$ and $(\lambda'', \mu' + \mu'' + \lambda'' - 1)$, respectively (Theorem 6);
- L_3 is accepted by a DFA of size $(\gcd(\lambda', \lambda''), \mu' + \mu'' + \mathrm{lcm}(\lambda', \lambda'') - 1)$ (Corollary 2).

According to Theorem 4, from these four automata we can get a DFA accepting L of size (λ, μ), where $\lambda = \mathrm{lcm}(\lambda', \lambda'')$ and $\mu = \mu' + \mu'' + \mathrm{lcm}(\lambda', \lambda'') - 1$. □

We now study the optimality of the result stated in Theorem 10. First, we show that this result is optimal when $\gcd(\lambda', \lambda'') > 1$. Subsequently, we will consider relatively prime λ' and λ'', and we will see that in this case the number of states in the cyclic part can be further reduced.

Let us start by proving the following result:

Theorem 11 *For any $\mu', \mu'' \geq 2$, $\lambda', \lambda'' \geq 2$, such that $\gcd(\lambda', \lambda'') > 1$, there exist two unary languages $\mathcal{L}'$ and $\mathcal{L}''$ which are accepted by two DFA's A' and A'' of size (λ', μ') and (λ'', μ''), respectively, such that the size of the minimal DFA accepting their concatenation is (λ, μ), with $\lambda = \mathrm{lcm}(\lambda', \lambda'')$ and $\mu = \mu' + \mu'' + \mathrm{lcm}(\lambda', \lambda'') - 1$.*

Proof. If λ'' divides λ' (λ' divides λ'', respectively), then the languages in the proof of Theorem 7 provide the desired examples.

Now, suppose that λ' does not divide λ'', and λ'' does not divide λ'. Consider the languages:

$$\mathcal{L}' = a^{\mu'+\lambda'-1}(a^{\lambda'})^* \cup a^{\mu'-2} \text{ and } \mathcal{L}'' = a^{\mu''+\lambda''-1}(a^{\lambda''})^* \cup a^{\mu''-2}.$$

It is not difficult to describe two automata A' and A'' of size (λ', μ') and (λ'', μ'') accepting $\mathcal{L}'$ and $\mathcal{L}''$, respectively. From these automata, according to Theorem 10, an automaton A of size (λ, μ) accepting $\mathcal{L}$ can be obtained. We observe that a state p_x on the cycle of A, with $0 \leq x < \lambda$, is final if and only if there is an integer $k \geq 1$ such that either $x = kg - 1$, or $x = k\lambda' - 2$, or $x = k\lambda'' - 2$, where $g = \gcd(\lambda', \lambda'')$.

In order to show that A is minimal, we prove that both conditions *(i)* and *(ii)* of Theorem 3 are satisfied.

Consider a maximal proper divisor d of λ. Then, either λ' divides d, or λ'' divides d. Suppose that λ' divides d, i.e., $d = \beta\lambda'$, for some $\beta \geq 1$, and consider $h = \lambda'' - 2$. Then $h + d = \lambda'' - 2 + \beta\lambda'$. So, $p_{(h+d) \text{ MOD } \lambda} \in F$ if and only if there exists an integer $k \geq 1$ such that either *(a)* $\lambda'' - 2 + \beta\lambda' = kg - 1$, or *(b)* $\lambda'' - 2 + \beta\lambda' = k\lambda' - 2$, or *(c)* $\lambda'' - 2 + \beta\lambda' = k\lambda'' - 2$.

Since g is greater than 1 and divides both λ' and λ'', the equality *(a)*, which reduces to $\lambda'' + \beta\lambda' = kg + 1$ cannot hold, for any integer k. Also equality *(b)* cannot hold since it implies that λ' divides λ''. Finally, equality *(c)* reduces to $\beta\lambda' = (k-1)\lambda''$; thus, it implies that $\beta\lambda'$, namely d, is a multiple of both λ' and λ'', i.e., a multiple of λ. This is a contradiction. Thus, we are able to conclude that $p_{(h+d) \text{ MOD } \lambda} \notin F$, while $p_h \in F$. The case of λ'' which divides λ' can be managed in a similar way. This permits us to conclude that condition *(i)* of Theorem 3 holds. Using Lemma 1, it is possible to verify that $a^{\mu'+\mu''+\text{lcm}(\lambda',\lambda'')-2} \notin \mathcal{L}$, while $a^{\mu'+\mu''+\text{lcm}(\lambda',\lambda'')+\lambda-2} \in \mathcal{L}$. Hence, condition *(ii)* of Theorem 3 also holds. This implies that A is minimal. □

Theorem 11 shows the optimality of the result stated in Theorem 10, for all $\mu', \mu'', \lambda', \lambda''$ such that $\gcd(\lambda', \lambda'') > 1$, with few exceptions for small μ', μ''.

We now consider the case of relatively prime λ' and λ''. The number of states in the cyclic part of the minimal DFA accepting the product of $\mathcal{L}'$ and $\mathcal{L}''$ is less than $\text{lcm}(\lambda', \lambda'')$. In particular, if both languages are infinite, then this number reduces to $\gcd(\lambda', \lambda'') = 1$, while if $\mathcal{L}''$ is finite it reduces to λ':

Theorem 12 *Let $\mathcal{L}'$ and $\mathcal{L}''$ be unary languages accepted by two automata A' and A'' of size (λ', μ'), (λ'', μ''), respectively, with $\mu', \mu'' \geq 0$, $\lambda', \lambda'' \geq 1$, such that* $\gcd(\lambda', \lambda'') = 1$.

If both $\mathcal{L}'$ and $\mathcal{L}''$ are infinite, then their concatenation is accepted by an automaton A of size $(1, \mu' + \mu'' + \lambda'\lambda'' - 1)$; if $\mathcal{L}''$ is finite, then the concatenation of $\mathcal{L}'$ and $\mathcal{L}''$ is accepted by an automaton of size $(\lambda', \mu' + \mu'' - 1)$.

These results are optimal, with the only exception being the trivial case $\mathcal{L}'' = \emptyset$.

Proof. Suppose that both $\mathcal{L}'$ and $\mathcal{L}''$ are infinite. The concatenation $\mathcal{L}$ of $\mathcal{L}'$ and $\mathcal{L}''$ can be expressed as in Equality (2). Since $\gcd(\lambda', \lambda'') = 1$, any string a^x with $x \geq \mu' + \mu'' + \text{lcm}(\lambda', \lambda'') - 1$ belongs to $a^{\mu'+\mu''}Y'Y''$ and then to $\mathcal{L}$. Hence, it is possible to conclude that a cycle of length 1 is sufficient. The optimality is a consequence of Theorem 9. When $\mathcal{L}''$ is finite, the result is an immediate consequence of Theorem 6 and of Theorem 7. □

4.4. Particular Cases

The construction of an automaton A accepting the concatenation of the languages $\mathcal{L}'$ and $\mathcal{L}''$ accepted by two given unary DFA's A' and A'' (Theorem 10) is based on Equality (2). When one or both noncyclic parts of $\mathcal{L}'$ and $\mathcal{L}''$ are empty, some of the languages on the right side are empty. Thus, evaluating in this cases the size of the resulting automata, one can easily get the following result:

Theorem 13 *Let $\mathcal{L}'$ and $\mathcal{L}''$ be unary languages accepted by two automata A' and A'' of size (λ', μ') and (λ'', μ''), respectively. The concatenation of $\mathcal{L}'$ and $\mathcal{L}''$ is accepted by a DFA of size (λ, μ), where $\mu = \mu' + \mu'' + \mathrm{lcm}(\lambda', \lambda'') - 1$ and:*

- *(i) if the initial path of A' does not contain any final state, then λ can be taken equal to λ';*
- *(ii) if the initial path of A'' does not contain any final state, then λ can be taken equal to λ'';*
- *(iii) if both the initial paths do not contain any final state, then λ can be taken equal to* $\gcd(\lambda', \lambda'')$.

We point out that the results stated in Theorem 13 are optimal when $\gcd(\lambda', \lambda'') > 1$. For statement (iii), this is a consequence of Theorem 9. This also implies the optimality of (ii) when $\lambda'' = \gcd(\lambda', \lambda'') = 2$. On the other hand, for $\lambda'' > 2$, the optimality of (ii) is given in the following result:

Theorem 14 *Given $\mu', \lambda' \geq 2$, $\lambda'' > 2$, $\mu'' \geq 0$ such that* $\gcd(\lambda', \lambda'') > 1$, *consider the languages*

$$\mathcal{L}' = a^{\mu'+\lambda'-1}(a^{\lambda'})^* \cup a^{\mu'-2} \quad \mathcal{L}'' = a^{\mu''+\lambda''-1}(a^{\lambda''})^*.$$

The languages $\mathcal{L}'$ and $\mathcal{L}''$ can be accepted by two automata A' and A'' of size (λ', μ') and (λ'', μ''), respectively. Furthermore, the minimal DFA accepting the concatenation $\mathcal{L} = \mathcal{L}'\mathcal{L}''$ has size $(\lambda'', \mu' + \mu'' + \mathrm{lcm}(\lambda', \lambda'') - 1)$.

Proof. (outline) An automaton A of size $(\lambda'', \mu' + \mu'' + \mathrm{lcm}(\lambda', \lambda'') - 1)$, accepting $\mathcal{L}$, can be obtained, from A' and A'', according to Theorem 10.

To show that A is minimal, we observe that a state p_x on the cycle of A is final if and only if there is an integer $k \geq 1$, such that either $x = k\lambda'' - 2$ or $x = kg - 1$, where $g = \gcd(\lambda', \lambda'')$.

Let d be a proper divisor of λ''. If d is a multiple of g, then we consider $h = \lambda'' - 2$. We observe that $p_h \in F$, while $p_{(h+d) \,\mathrm{MOD}\, \lambda''} \in F$ if and only if there exists an integer $k \geq 1$ such that either *(a)* $\lambda'' - 2 + d = k\lambda'' - 2$ or *(b)* $\lambda'' - 2 + d = kg - 1$. *(a)* implies that d is a multiple of λ'', while *(b)* implies that $g > 1$ divides 1. Thus, $p_{(h+d) \,\mathrm{MOD}\, \lambda''} \notin F$. On the other hand, if d is not a multiple of g, then we consider $h = \lambda'' - 1$. Also in this case $p_h \in F$, while $p_{(h+d) \,\mathrm{MOD}\, \lambda''} \in F$ if and only if there exists an integer $k \geq 1$ such that either *(a)* $\lambda'' - 1 + d = k\lambda'' - 2$ or *(b)* $\lambda'' - 1 + d = kg - 1$. Using the hypothesis that $\lambda'' > 2$, it is possible to show that condition *(a)*, which reduces to $(k-1)\lambda'' = d + 1$, cannot hold. Since g divides λ'' and kg, but does not divides d, condition *(b)*, which reduces to $\lambda'' + d = kg$, cannot hold. Thus, we conclude that $p_{(h+d) \,\mathrm{MOD}\, \lambda''} \notin F$. Hence, condition *(i)* of Theorem 3 holds. Now, observe that $p_{\lambda''-1} \in F$, while, using Lemma 1, it is possible to verify that $a^{\mu'+\mu''+\mathrm{lcm}(\lambda',\lambda'')-2} \notin \mathcal{L}$, i.e., $q_{\mu'+\mu''+\mathrm{lcm}(\lambda',\lambda'')-2} \notin F$. Hence, condition *(ii)* holds also. This permits us to conclude that the automaton A is minimal. □

Finally, we point out that the optimality of Theorem 13*(i)*, for $\gcd(\lambda', \lambda'') > 1$, can be done in a similar way.

We conclude this section by summarizing, in Table 1 and in Table 2, the results we have proved concerning the state complexity of the concatenation of two languages $\mathcal{L}'$ and $\mathcal{L}''$ accepted by two automata A' and A'' of size (λ', μ') and (λ'', μ''), respectively.

Table 1. State complexity of the concatenation, *when* $\gcd(\lambda', \lambda'') > 1$. X' and X'' denote the languages accepted by the initial paths of A' and A'', respectively, i.e, $X' = \{a^x \in \mathcal{L}' \ : \ x < \mu'\}$ and $X'' = \{a^x \in \mathcal{L}'' \ : \ x < \mu''\}$.

	$X'' \neq \emptyset$	$X'' = \emptyset$
$X' \neq \emptyset$	$(\mathrm{lcm}(\lambda', \lambda''), \mu' + \mu'' + \mathrm{lcm}(\lambda', \lambda'') - 1)$ upper bound: Th. 10 lower bound: Th. 11	$(\lambda'', \mu' + \mu'' + \mathrm{lcm}(\lambda', \lambda'') - 1)$ upper bound: Th. 13 lower bound: Th. 9 and Th. 14
$X' = \emptyset$	$(\lambda', \mu' + \mu'' + \mathrm{lcm}(\lambda', \lambda'') - 1)$ upper bound: Th. 13 lower bound: Th. 9 and Th. 14	$(\gcd(\lambda', \lambda''), \mu' + \mu'' + \mathrm{lcm}(\lambda', \lambda'') - 1)$ upper bound: Th. 13 lower bound: Th. 9

Table 2. State complexity of the concatenation, *when* $\gcd(\lambda', \lambda'') = 1$.

	$\#L' = \infty$	$\#L' < \infty$
$\#L'' = \infty$	$(1, \mu' + \mu'' + \lambda'\lambda'' - 1)$ Th. 12	$(\lambda'', \mu' + \mu'' - 1)$ upper bound: Th. 6 lower bound: Th. 7
$\#L'' < \infty$	$(\lambda', \mu' + \mu'' - 1)$ upper bound: Th. 6 lower bound: Th. 7	$(1, \mu' + \mu'' - 1)$ upper bound: Th. 6 lower bound: trivial

5. Kleene Star

In this section, we present some short considerations concerning the state complexity of the Kleene star operation in the unary case.

First of all, we recall the following result [19, Th. 5.3]:

Theorem 15 *If L is a unary regular language accepted by an n–state DFA, then L^* is accepted by a DFA with $(n-1)^2 + 1$ states. Furthermore, for any $n \geq 1$ this result cannot be improved.*

We observe that if L is accepted by an automaton of size (λ, μ), then the cycle in the minimal DFA accepting L^* cannot have more than λ states, i.e., the size (λ^*, μ^*) of the minimal automaton accepting L^* verifies $\lambda^* \leq \lambda$.

We now analyze some limit situations.

First, we suppose that $\mu = 0$, i.e., L is λ–cyclic. If $L = (a^\lambda)^*$, then $L = L^*$ and $(\lambda^*, \mu^*) = (\lambda, 0)$. Otherwise, let k be an integer such that $a^k \in L$ and $0 < k < \lambda$. Any string which has length of the form $kx + \lambda y$, with $x \geq 1$ and $y \geq 0$, or $x = y = 0$, belongs to L^*. Hence, the length of the loop is $\lambda^* \leq \gcd(\lambda, k) \leq k$. In particular, when $L = a^k(a^\lambda)^*$, by using Lemma 1, it is not difficult to conclude that $\lambda^* = \gcd(\lambda, k)$ and $\mu^* = \mathrm{lcm}(k, \lambda) - \lambda + 1$. For $k = \lambda - 1$ this reduces to $\lambda^* = 1$ and $\mu^* = (\lambda - 1)^2$, which exactly matches the number of states given in Theorem 15.

Now, we suppose that $\mu > 0$ and $\lambda = 1$. If $p_0 \in F$ then all strings of length greater than $\mu - 1$ belong to L. Thus L^* is accepted by an automaton of size $(1, \mu^*)$, with $\mu^* \leq \mu$. On the other hand, if $p_0 \notin F$ then L is finite. This case was analyzed in [17], where it was proved that L^* is accepted by an automaton with at most

$n^2 - 7n + 13$ states, where $n = \mu + \lambda = \mu + 1$ (this result, which is optimal, holds for $n \geq 3$). We can suppose that $a^{\mu-1} \in L$, otherwise the size of the DFA accepting L can be reduced. If $L = \{a^{\mu-1}\}$, then L^* is accepted by an automaton of size $(\mu - 1, 0)$. If $L = \{a^s, a^{\mu-1}\}$, with $0 \leq s < \mu - 1$, then, using Lemma 1, it can be shown that L^* is accepted by an automaton of size (λ^*, μ^*), with $\lambda^* = \gcd(\mu - 1, s)$ and $\mu^* = \mathrm{lcm}(\mu - 1, s) - \mu - s + 2$. In particular, when $s = \mu - 2$, we get $\lambda^* = 1$ and $\mu^* = \mu^2 - 5\mu + 6$ (note that for $n = \lambda + \mu$, $\lambda^* + \mu^*$ is exactly $n^2 - 7n + 13$).

As pointed out in [17], the reader can verify that the size obtained in the last case is an upper limit for the case of L containing three or more words.

6. Jacobsthal's function

As observed in Section 3, to understand the state complexity of the intersection of regular languages, we need to estimate the function

$$F(n', n'') = \max_{\substack{1 \leq \lambda' \leq n' \\ 1 \leq \lambda'' \leq n''}} (\max(n' - \lambda', n'' - \lambda'') + \mathrm{lcm}(\lambda', \lambda'')).$$

This in turn suggests studying the somewhat simpler and more natural function

$$G(n', n'') = \max_{\substack{1 \leq \lambda' \leq n' \\ 1 \leq \lambda'' \leq n''}} \mathrm{lcm}(\lambda', \lambda''),$$

which is also related to the state complexity of the concatenation.

To the best of our knowledge, neither F nor G has been studied previously, although both functions are closely related to the *Jacobsthal function* $g(n)$, which is defined to be the least integer r such that every set of r consecutive integers contains at least one integer relatively prime to n [8].

Below we show an interesting connection between this problem and state complexity for intersection of unary languages. First, however, we state two known upper bounds on this function. The first is an explicit bound due to Kanold [9]:

Theorem 16 *Let $\omega(n)$ denote the number of distinct prime factors of the positive integer n. Then $g(n) \leq 2^{\omega(n)}$ for all integers $n \geq 1$.*

The second bound is due to Iwaniec [7]:

Theorem 17 *There exists a constant c_1 such that $g(n) \leq c_1(\log n)^2$ for all integers $n \geq 1$.*

First we obtain a lower bound on G (and hence F):

Theorem 18 *Let $n' \leq n''$. Then there exists a constant c_1 such that $F(n', n'') \geq G(n', n'') \geq n'n'' - c_1(\log n')^2 n'$.*

Proof. By Iwaniec's theorem, there exists k with $0 \leq k \leq c_1(\log n')^2$ such that $\gcd(n', n'' - k) = 1$. Hence $G(n', n'') \geq n'(n'' - k) \geq n'(n'' - c_1(\log n')^2)$. □

Carl Pomerance has kindly pointed out (personal communication) that the lower bound of Theorem 18 can be improved in the case where n' and n'' do not differ much in size, as follows. We use a result of Adhikari and Balasubramanian [1]:

Theorem 19 *If x, y are positive integers $\leq N$, then there exist integers a, b with $a = O(\log\log\log N)$ and $b = O((\log N)/(\log\log N))$ such that* $\gcd(x - a, y - b) = 1$.

Using this theorem, we obtain the following:

Theorem 20

(a) If $n' \le n'' \le \frac{n' \log n'}{(\log\log n')(\log\log\log n')}$, *then there exists a constant* c_2 *such that* $F(n', n'') \ge G(n', n'') \ge n'n'' - c_2 \frac{\log n'}{\log\log n'} n'$.

(b) If $\frac{n' \log n'}{(\log\log n')(\log\log\log n')} \le n'' \le \frac{n' (\log n')^2}{\log\log\log n'}$ *then there exists a constant* c_3 *such that* $F(n', n'') \ge G(n', n'') \ge n'n'' - c_3(\log\log\log n')n''$.

Next, we find a upper bound on F. First we prove the following lemma.

Lemma 2 *Let* n', n'' *be fixed positive integers. The quantity*

$$Q(\lambda', \lambda'') := \max(n' - \lambda', n'' - \lambda'') + \operatorname{lcm}(\lambda', \lambda'')$$

is maximized ($1 \le \lambda' \le n'$, $1 \le \lambda'' \le n''$) *only if* $\gcd(\lambda', \lambda'') = 1$.

Proof. Assume not. Then $Q(\lambda', \lambda'')$ is maximized for some λ', λ'' with $\gcd(\lambda', \lambda'') = g > 1$. Assume without loss of generality that $n' \ge n''$. For $n' < 11$ the theorem can be verified by a simple computer program. Hence assume $n' \ge 11$.

We have $\max(n' - \lambda', n'' - \lambda'') < n'$ and $\operatorname{lcm}(\lambda', \lambda'') = \frac{\lambda'\lambda''}{g} \le \frac{n'^2}{2}$, so $Q(\lambda', \lambda'') < n' + \frac{n'^2}{2}$. By Theorem 16 we know there exists a k, $1 \le k \le 2^{\omega(n')}$ such that $\gcd(n', n' - k) = 1$. Since $Q(\lambda', \lambda'')$ is a maximum, we have $Q(\lambda', \lambda'') \ge n'(n' - k) + k > n'(n' - 2^{\omega(n')})$. Putting the inequalities for Q together, we get

$$n'(n' - 2^{\omega(n')}) < n' + \frac{n'^2}{2},$$

and so $n' - 2^{\omega(n')} < 1 + \frac{n'}{2}$. Thus $n' < 2(2^{\omega(n')} + 1)$.

However, we claim that $n' > 2(2^{\omega(n')} + 1)$ for $n' \ge 11$. For $11 \le n' \le 141$ this follows by an explicit calculation. Otherwise $n' \ge 142$. We now use a theorem of Robin [16] which states $\omega(n) \le t(n)$ where

$$t(n) := \frac{\log n}{\log\log n} + 1.45743 \frac{\log n}{(\log\log n)^2}.$$

Since $n' \ge 142$, we have $\log\log n' > 1.6$ and so

$$t(n') < \frac{\log_2 n'}{1.6 \log_2 e} + 1.45743 \frac{\log_2 n'}{2.56 \log_2 e} < .83 \log_2 n'.$$

We thus obtain $2(2^{\omega(n')+1}) < 2(2^{t(n')} + 1) < 2(n'^{.83} + 1)$ and it is easily verified that $2(n'^{.83} + 1) < n'$ for $n' \ge 70$. This contradiction completes the proof. □

Remark. Ming-wei Wang points out (personal communication) that a slightly weaker result is much easier to prove: namely, that Q achieves its maximum at some (λ', λ'') with $\gcd(\lambda', \lambda'') = 1$ (as opposed to "only if"). For if $\gcd(\lambda', \lambda'') > 1$, then write $\lambda' = 2^{e_1} 3^{e_2} \cdots p_k^{e_k}$ and $\lambda'' = 2^{f_1} 3^{f_2} \cdots p_k^{f_k}$ where p_i is the k'th prime and p_k is the largest prime dividing either λ' or λ''. Let $d' = \prod_{\substack{1 \le i \le k \\ e_i \le f_i}} p_i^{e_i}$ and $d'' = \prod_{\substack{1 \le i \le k \\ e_i > f_i}} p_i^{f_i}$. Then $\operatorname{lcm}(\lambda'/d', \lambda''/d'') = \operatorname{lcm}(\lambda', \lambda'')$, and hence we have $Q(\lambda'/d', \lambda''/d'') \ge Q(\lambda', \lambda'')$. However $\gcd(\lambda'/d', \lambda''/d'') = 1$.

Remark. Note that $F(n', n'') = \max_{\substack{1 \le \lambda' \le n' \\ 1 \le \lambda'' \le n''}} \max(n' - \lambda', n'' - \lambda'') + \mathrm{lcm}(\lambda', \lambda'')$ does not necessarily achieve its maximum at the same pair (λ', λ'') which maximizes $G(n', n'') = \max_{\substack{1 \le \lambda' \le n' \\ 1 \le \lambda'' \le n''}} \mathrm{lcm}(\lambda', \lambda'')$. For example, $F(148, 30) = 4295$, which is uniquely achieved at $(\lambda', \lambda'') = (143, 30)$, while $G(148, 30) = 4292$, which is uniquely achieved at $(\lambda', \lambda'') = (148, 29)$.

We can now prove our upper bound.

Theorem 21 *There exist a constant c_4 and infinitely many distinct pairs n', n'' with $n'' < n'$ such that $G(n', n'') \le F(n', n'') \le n'n'' - c_4\sqrt{\frac{\log n'}{\log\log n'}}n'$.*

Proof. Let $d \ge 1$ be a fixed integer. Let $S_d = \{(i, j) \ : \ i, j \ge 0 \text{ and } i + j < d\}$. For each pair $(i, j) \in S_d$, choose a distinct prime $q_{i,j}$ from the set $\{p_1, p_2, \dots, p_v\}$, where p_i denotes the i'th prime and $v = d(d+1)/2$. By the Chinese remainder theorem, we can find n', n'' such that $q_{i,j} \mid n' - i$ and $q_{i,j} \mid n'' - j$ for all pairs $(i, j) \in S_d$. Furthermore, we may choose n' and n'' such that $K \le n'' < 2K$, $2K \le n' < 3K$, where $K := \prod_{1 \le i \le v} p_i$. By the prime number theorem (e.g., [2]), we have $K = e^{(1+o(1))v \log v}$. Hence there exists a constant c_5 such that $d \ge c_5\sqrt{\frac{\log n'}{\log\log n'}}$.

It follows that $\gcd(n' - i, n'' - j) > 1$ for all pairs $(i, j) \in S_d$. By Lemma 2, we know that F cannot achieve its maximum when $(\lambda', \lambda'') \in S_d$.

It follows that $F(n', n'') \le \max_{b+\lambda'=d}((n' - b)(n'' - \lambda') + d)$. But

$$\max_{b+\lambda'=d}((n' - b)(n'' - \lambda') + d) \le n'n'' - dn'' + d^2/4 + d.$$

Hence $F(n', n'') \le n''(n' - d) + d^2/4 + d$. Since $n'' \ge n'/3$, the result follows. □

Remark. This result suggests defining a function $S(n)$ to be the least positive integer r such that there exists an integer m, $0 \le m \le r$, with $\gcd(r - i, m - j) > 1$ for $0 \le i, j < n$. By an argument similar to that given above, we know that $S(n) < e^{(1+o(1))2n^2 \log n}$. The following table gives the first few values of $S(n)$:

Table 3. First few values of $S(n)$.

n	$S(n)$	m
1	2	0
2	21	15
3	1310	1276

It is possible to prove through brute force calculation that $450000 < S(4) \le 172379781$. The upper bound follows from the fact that if

$$(x, y) = (172379781, 153132345),$$

then we have $\gcd(x - i, y - j) > 1$ for $0 \le i, j < 4$.

7. Acknowledgments

The second author would like to thank Carl Pomerance and Ming-wei Wang for their helpful suggestions, and Jean-Eric Pin for pointing out the work of Nicaud.

References

1. S. D. Adhikari and R. Balasubramanian. On a question regarding visibility of lattice points. *Mathematika*, 43:155–158, 1996.
2. E. Bach and J. Shallit. *Algorithmic Number Theory.* MIT Press, 1996.
3. A. Brauer. On a problem of partitions. *Amer. J. Math.*, 64:299–312, 1942.
4. M. Chrobak. Finite automata and unary languages. *Theoretical Computer Science*, 47:149–158, 1986.
5. Yu. L. Ershov. On a conjecture of V. A. Uspenskii. *Algebra i Logika* 1:(4):45–48, 1962. (In Russian).
6. J. E. Hopcroft and J. D. Ullman. *Introduction to Automata Theory, Languages, and Computation.* Addison-Wesley, 1979.
7. H. Iwaniec. On the problem of Jacobsthal. *Dem. Math.*, 11:225–231, 1978.
8. E. Jacobsthal. Über Sequenzen ganzer Zahlen, von denen keine zu n teilerfremd ist. I-III. *Norske Vid. Selsk. Forh. Trondheim*, 33:117–139, 1960.
9. H.-J. Kanold. Über eine zahlentheoretische Funktion von Jacobsthal. *Math. Annalen*, 170:314–326, 1967.
10. Yu. I. Lyubich. Estimates for optimal determinization of nondeterministic autonomous automata. *Sibirskii Matemat. Journal*, 5:(2):337–355, 1964. (In Russian).
11. Yu. I. Lyubich. Estimates of the number of states that arise in the determinization of a nondeterministc autonomous automaton. *Doklady Akad. Nauk SSSR*, 155:(1):41–43, 1964. In Russian. English translation in *Soviet Mathematics* 5:345–348, 1964.
12. R. Mandl. Precise bounds associated with the subset construction on various classes of nondeterministic finite automata. In *Proc. 7th Princeton Conference on Information and System Science*, 263–267, 1973.
13. A. R. Meyer and M. J. Fischer. Economy of description by automata, grammars, and formal systems. In *Conference Record 1971 Twelfth Annual Symposium on Switching and Automata Theory*, IEEE Computer Society, pp. 188–191, 1971.
14. F. R. Moore. On the bounds for state-set size in the proofs of equivalence between deterministic, nondeterministic, and two-way finite automata. *IEEE Trans. Computers*, 20:1211–1214, 1971.
15. C. Nicaud. Average state complexity of operations on unary automata. In M. Kutylowski, L. Pacholski, and T. Wierzbicki, editors, *Proc. 24th Symposium, Mathematical Foundations of Computer Science 1999*, Vol. 1672 of *Lecture Notes in Computer Science*, pp. 231–240. Springer-Verlag, 1999.
16. G. Robin. Estimation de la fonction de Tchebychef Θ sur le k-ième nombre premier et grandes valeurs de la fonction $\omega(n)$ nombre de diviseurs premiers de n. *Acta Arith.*, 42:367–389, 1983.
17. S. Yu. State complexity of regular languages. In *International Workshop on Descriptional Complexity of Automata, Grammars and Related Structures, Preproceedings*, pp. 77–88. Department of Computer Science, Otto-von-Guericke University of Magdeburg, July 1999.
18. S. Yu and Q. Zhuang. On the state complexity of intersection of regular languages. *SIGACT News*, 22(3):52–54, Summer 1991.
19. S. Yu, Q. Zhuang, and K. Salomaa. The state complexity of some basic operations on regular languages. *Theoret. Comput. Sci.*, 125:315–328, 1994.