Simulating Finite Automata with Context-Free Grammars

Michael Domaratzki [a], Giovanni Pighizzini [b], Jeffrey Shallit [c,1]

[a] *Department of Computer Science, Queen's University Kingston, Ontario K7L 3N6, Canada*

[b] *Dipartimento di Scienze dell'Informazione, Università degli Studi di Milano via Comelico 39, 20135 Milano, Italy*

[c] *Department of Computer Science, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada*

Abstract

We consider simulating finite automata (both deterministic and nondeterministic) with context-free grammars in Chomsky normal form (CNF). We show that any unary DFA with n states can be simulated by a CNF grammar with $O(n^{1/3})$ variables, and this bound is tight. We show that any unary NFA with n states can be simulated by a CNF grammar with $O(n^{2/3})$ variables. Finally, for larger alphabets we show that there exist languages which can be accepted by an n-state DFA, but which require $\Omega(n/\log n)$ variables in any equivalent CNF grammar.

Key words: formal languages, context-free grammar, finite automata

1 Introduction

In *descriptional complexity* we are interested in the descriptive power of various computing models, such as deterministic finite automata (DFA's), non-

Email addresses: `domaratz@cs.queensu.ca` (Michael Domaratzki), `pighizzi@dsi.unimi.it` (Giovanni Pighizzini), `shallit@math.uwaterloo.ca` (Jeffrey Shallit).

URLs: `http://www.cs.queensu.ca/home/domaratz/` (Michael Domaratzki), `http://homes.dsi.unimi.it/~pighizzi/home-eng.html` (Giovanni Pighizzini), `http://www.math.uwaterloo.ca/~shallit` (Jeffrey Shallit).

[1] Research supported in part by a grant from NSERC.

deterministic finite automata (NFA's), and context-free grammars (CFG's) [12]. For example, many recent papers have examined the number of states required by deterministic finite automata to simulate various operations on languages (see, e.g., Yu, Zhuang, and Salomaa [19]). This is in sharp contrast to the more familiar *computational complexity*, where we are instead concerned with the time and space used by computing models such as Turing machines as a function of the size of the input.

In this paper we study the descriptional complexity of context-free grammars that simulate finite automata. For both DFA's and NFA's the number of states is a generally-accepted measure of descriptional complexity (e.g., [14,3]), although it can be argued that for NFA's the number of transitions is more suitable. However, for CFG's there is no univerally-agreed-upon measure of descriptional complexity. For example, the following are just three of the many proposed measures of the complexity of a CFG:

(a) the number of variables [9,7];
(b) the number of productions [10];
(c) the sum of the lengths of the productions [15].

For still other proposals, see [11].

Given a CFL L, we may measure its complexity by choosing one of the above measures and computing the minimum over all CFG's G with $L = L(G)$. In this paper we focus on measure (a). As stated it is not completely satisfactory for the descriptional complexity of CFL's; for example, if there are no restrictions on the length of productions then any finite language can be generated by a CFG with a single variable. So instead we restrict our attentions to CFG's in Chomsky normal form (CNF). Recall that a context-free grammar $G = (V, \Sigma, P, S)$ is said to be in Chomsky normal form if every production is of the form $A \rightarrow BC$, or $A \rightarrow a$, where $A, B, C \in V$, and $a \in \Sigma$. This measure of descriptional complexity was previously mentioned by Shallit and Wang [18] and appears in a recent paper of Nederhof and Satta [16]. It is also of interest because it generalizes the well-studied concept of word chains (see § 3).

The standard construction showing that every DFA M (or NFA, for that matter) has an equivalent regular grammar (see, for example, [13, §9.1]) proves that if M has n states and an input alphabet Σ of k symbols, then there is a CNF grammar with $n + k$ variables generating $L(M) - \{\epsilon\}$. We will see that this bound can be significantly improved in the unary case.

We say a grammar G is in *binary normal form* (BNF) if every production is in one of the following four forms: $A \rightarrow a$, $A \rightarrow \epsilon$, $A \rightarrow B$, or $A \rightarrow BC$, with $A, B, C \in V$ and $a \in \Sigma$. We use the following fact throughout the paper: if $G = (V, \Sigma, P, S)$ is a grammar in BNF, then there exists a grammar

$G' = (V, \Sigma, P', S)$ in Chomsky normal form such that $L(G') = L(G) - \{\epsilon\}$. To see this, note that the usual algorithm [13, §4.4] for removing ϵ-productions and unit productions does not introduce additional variables.

2 Simulation of Unary Automata

In this section we consider simulating unary automata, that is, automata whose input alphabet consists of a single symbol.

Lemma 2.1 *Let T be any subset of $\{\epsilon, a, a^2, \ldots, a^{n-1}\}$. Then there exists a BNF grammar G such that $L(G) = T$, and G has $O(n^{1/3})$ variables.*

Proof. Define $r := \lceil n^{1/3} \rceil$. We can then express an integer i, $0 \leq i < n$, in base r using at most 3 digits, say $i = e_i r^2 + f_i r + g_i$, with $0 \leq e_i, f_i, g_i < r$. We now define some productions, as follows:

$$\begin{array}{ccc}
G_0 \rightarrow \epsilon & F_0 \rightarrow \epsilon & E_0 \rightarrow \epsilon \\
G_1 \rightarrow a & F_1 \rightarrow G_r & E_1 \rightarrow F_r \\
G_2 \rightarrow G_1 G_1 & F_2 \rightarrow F_1 F_1 & E_2 \rightarrow E_1 E_1 \\
G_3 \rightarrow G_2 G_1 & F_3 \rightarrow F_2 F_1 & E_3 \rightarrow E_2 E_1 \\
\vdots & \vdots & \vdots \\
G_{r-1} \rightarrow G_{r-2} G_1 & F_{r-1} \rightarrow F_{r-2} F_1 & E_{r-1} \rightarrow E_{r-2} E_1 \\
G_r \rightarrow G_{r-1} G_1 & F_r \rightarrow F_{r-1} F_1 &
\end{array}$$

If $X \in V$ is a variable in a grammar $G = (V, \Sigma, P, S)$, we abuse notation somewhat by defining $L(X) = \{x \in \Sigma^* \ : \ X \Longrightarrow^* x\}$. It is trivial to prove by induction that

$$\begin{aligned}
L(G_i) &= \{a^i\}, \quad 0 \leq i \leq r; \\
L(F_i) &= \{a^{ir}\}, \quad 0 \leq i \leq r; \\
L(E_i) &= \{a^{ir^2}\}, \quad 0 \leq i < r.
\end{aligned}$$

Now we define the remaining productions.

$$
\begin{aligned}
S &\to E_0S_0 \mid E_1S_1 \mid E_2S_2 \mid \cdots \mid E_{r-1}S_{r-1} \\
S_0 &\to F_iG_j \text{ for all } i,j,\ 0 \le i,j < r, \text{ such that } a^{ir+j} \in T; \\
S_1 &\to F_iG_j \text{ for all } i,j,\ 0 \le i,j < r, \text{ such that } a^{r^2+ir+j} \in T; \\
&\vdots \\
S_{r-1} &\to F_iG_j \text{ for all } i,j,\ 0 \le i,j < r, \text{ such that } a^{(r-1)r^2+ir+j} \in T.
\end{aligned}
$$

The resulting grammar is in BNF, and the total number of variables is $4r+3 = O(n^{1/3})$.

Example 2.2 Consider representing the set $T = \{a^2, a^4, a^6, a^{17}, a^{18}, a^{21}, a^{25}\}$ by a grammar in CNF. Here $n = 26$ and $r = 3$. The following BNF grammar generates S:

$$
\begin{array}{ll}
S \to E_0S_0 \mid E_1S_1 \mid E_2S_2 & F_0 \to \epsilon \\
S_0 \to F_0G_2 \mid F_1G_1 \mid F_2G_0 & F_1 \to G_3 \\
S_1 \to F_2G_2 & F_2 \to F_1F_1 \\
S_2 \to F_0G_0 \mid F_1G_0 \mid F_2G_1 & F_3 \to F_2F_1 \\
G_0 \to \epsilon & E_0 \to \epsilon \\
G_1 \to a & E_1 \to F_3 \\
G_2 \to G_1G_1 & E_2 \to E_1E_1 \\
G_3 \to G_2G_1 &
\end{array}
$$

The ϵ-productions, unit productions, and useless symbols may easily be removed to give the following equivalent grammar in CNF:

$$
\begin{array}{ll}
S \to G_1G_1 \mid F_1G_1 \mid F_1F_1 \mid E_1S_1 \mid E_2S_2 \mid E_1E_1 & F_1 \to G_2G_1 \\
S_1 \to F_2G_2 & F_2 \to F_1F_1 \\
S_2 \to G_2G_1 \mid F_2G_1 & E_1 \to F_2F_1 \\
G_1 \to a & E_2 \to E_1E_1 \\
G_2 \to G_1G_1 &
\end{array}
$$

Next, we state a lemma from [17]:

Lemma 2.3 *Let M be a unary DFA with n states. Then there exist integers $t \ge 0$ and $c \ge 1$ with $t + c \le n$, and sets $A \subseteq \{\epsilon, a, a^2, \ldots, a^{t-1}\}$ and $B \subseteq \{\epsilon, a, a^2, \ldots, a^{c-1}\}$ such that $L(M) = A + Ba^t\{a^c\}^*$.*

Now we can prove an upper bound.

Theorem 2.4 *Let M be a unary DFA with n states. Then there exists a context-free grammar G in CNF such that $L(G) = L(M) - \{\epsilon\}$, and G has $O(n^{1/3})$ variables.*

Proof. By Lemma 2.3 we can write $L(M) = A + Ba^t\{a^c\}^*$ for suitable A, B, t, c. By Lemma 2.1, we can construct BNF grammars with $O(n^{1/3})$ variables for the languages A, B, $\{a^t\}$, and $\{a^c\}$.[2] We can now easily combine these BNF grammars to get a BNF grammar for $A+Ba^t\{a^c\}^*$, having $O(n^{1/3})$ variables. Hence a CNF grammar for $L(M)-\{\epsilon\}$ exists with $O(n^{1/3})$ variables.

Remark. Our upper bound can be viewed as a trade-off result, in that we have decreased the number of variables in our grammar to $O(n^{1/3})$ at the cost of a linear increase in the total size of the description.

We now prove a matching lower bound.

Theorem 2.5 *There exist constants c, n_0 such that for all integers $n \geq n_0$ there exists a finite subset $T \subseteq \{a, a^2, \ldots, a^{n-1}\}$ such that any context-free grammar G in CNF with $L(G) = T$ has at least $cn^{1/3}$ variables.*

Proof. Suppose $L(G) = T$, and G has t variables. If G is in CNF then there are $t^3 + t$ possible productions and for each production we can decide whether or not to include it in the grammar. This gives 2^{t^3+t} distinct grammars. But there are 2^{n-1} possible subsets of $\{a, a^2, \ldots, a^{n-1}\}$. It follows that $t^3+t \geq n-1$, and hence $t = \Omega(n^{1/3})$, as desired.

Corollary 2.6 *There exist constants c, n_0 such that for all $n \geq n_0$ there is a unary DFA M of n states, accepting a finite language, such that any CNF grammar G with $L(G) = L(M) - \{\epsilon\}$ has at least $cn^{1/3}$ variables.*

Proof. Use Theorem 2.5 and the fact that any subset of $\{\epsilon, a, a^2, \ldots, a^{n-1}\}$ can be accepted by a DFA containing $\leq n+1$ states.

We now turn to nondeterministic finite automata.

Theorem 2.7 *Let M be a unary NFA with n states. Then there exists a context-free grammar G in CNF such that $L(G) = L(M) - \{\epsilon\}$, and G has $O(n^{2/3})$ variables.*

Proof. We use a result of Chrobak [4] which says that every unary NFA with n states is equivalent to an NFA in a certain normal form (called Chrobak normal form), which has the following properties: there is a "tail" of $O(n^2)$ states, ending in a single nondeterministic state which leads to a number of different cycles, and the total number of states in all the cycles is bounded above by n. See Figure 1 for an illustration.

[2] Actually, using the "binary method", we can generate the languages $\{a^c\}$ and $\{a^t\}$ using context-free grammars in BNF having only $O(\log n)$ variables; see [5].

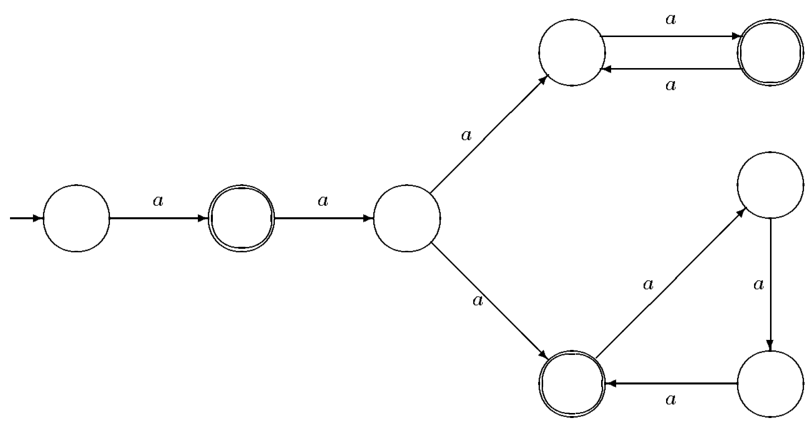

Fig. 1. An NFA in Chrobak normal form

Thus it follows that

$$L(M) = A \cup a^t \left(\bigcup_{1 \le i \le s} B_i \{a^{c_i}\}^* \right)$$

for some sets $A \subseteq \{\epsilon, a, \ldots, a^{t-1}\}$ with $t = O(n^2)$, and $B_i \subseteq \{\epsilon, a, \ldots, a^{c_i - 1}\}$, for some integers $s, c_1, \ldots, c_s > 0$, such that $c_1 + \cdots + c_s \le n$.

We now describe a set of variables and productions which can be used to generate the set of strings corresponding to the cycles of the automaton, namely, the set $\bigcup_{1 \le i \le s} B_i \{a^{c_i}\}^*$.

To this end, we define $r := \lceil n^{1/3} \rceil$ and, exactly as in the proof of Lemma 2.1, we introduce the variables E_i, F_i, G_i, $i = 0, \ldots, r$, and the corresponding productions, in such a way that

$$\begin{aligned} L(G_i) &= \{a^i\}, \quad 0 \le i \le r; \\ L(F_i) &= \{a^{ri}\}, \quad 0 \le i \le r; \\ L(E_i) &= \{a^{r^2 i}\}, \quad 0 \le i < r. \end{aligned}$$

Now we consider the ith cycle, whose length is c_i, and we define $r_i := \lceil c_i / r^2 \rceil$. First, we describe a set of variables and productions useful to generate the set B_i. More precisely, we introduce the variables

$$S^{(i)}, S_0^{(i)}, \ldots, S_{r_i - 1}^{(i)},$$

with the productions:

$$S^{(i)} \to E_0 S_0^{(i)} \mid E_1 S_1^{(i)} \mid E_2 S_2^{(i)} \mid \cdots \mid E_{r_i-1}^{(i)} S_{r_i-1}^{(i)} \quad \text{and}$$
$$S_h^{(i)} \to F_k G_j \text{ for all } k, j, h,\ 0 \le k, j < r,\ 0 \le h < r_i, \text{ such that } a^{hr^2 + kr + j} \in B_i.$$

It is easy to verify that $L(S^{(i)}) = B_i$.

As a second step, we consider the cycle length c_i. Let $j, k, h \geq 0$ be the integers such that $hr^2 + kr + j = c_i$. We introduce two variables $T^{(i)}$ and $T'^{(i)}$ with the productions $T^{(i)} \rightarrow E_h T'^{(i)}$ and $T'^{(i)} \rightarrow F_k G_j$, where $hr^2 + kr + j = c_i$. Then $L(T^{(i)}) = \{a^{c_i}\}$.

Finally, we introduce a further variable $U^{(i)}$ with the productions $U^{(i)} \rightarrow S^{(i)} \mid T^{(i)}U^{(i)}$. From the previous discussion, it is not difficult to conclude that $L(U^{(i)})$ is the language accepted by the i'th cycle, i.e.,

$$L(U^{(i)}) = B_i \{a^{c_i}\}^*.$$

Now we compute the number of variables introduced so far. The number of variables E_i, F_i, and G_i is $O(n^{1/3})$. Furthermore, for the ith cycle, we have introduced at most $r_i + 4$ variables. Thus, the total number is

$$\sum_{1 \leq i \leq s} (r_i + 4) = O(s) + \sum_{1 \leq i \leq s} r_i = O(s) + \#\{i \mid r_i = 1\} + \sum_{\substack{1 \leq i \leq s \\ r_i > 1}} r_i,$$

where $\#T$ denotes the cardinality of a set T. Observe that we may assume that each of the cycle lengths is distinct, for otherwise we could simply consolidate cycles of equal lengths. Thus, $s = O(n^{1/2})$. Furthermore, $\#\{i \mid r_i = 1\} \leq s$.

By definition, $r_i > 1$ iff $c_i \geq r^2 = (\lceil n^{1/3} \rceil)^2$. Since $\sum_{1 \leq i \leq s} c_i \leq n$, the number of cycles of length at least r^2 is bounded by $r = \lceil n^{1/3} \rceil$. Hence

$$\sum_{\substack{1 \leq i \leq s \\ r_i > 1}} r_i = \sum_{\substack{1 \leq i \leq s \\ c_i \geq r^2}} \left\lceil \frac{c_i}{r^2} \right\rceil \leq r \left\lceil \frac{c_i}{r^2} \right\rceil \leq n^{1/3} \left(\frac{n}{\lceil n^{1/3} \rceil^2} \right) = O(n^{2/3}).$$

By Lemma 2.1, the languages A and $\{a^t\}$ can be generated with BNF grammars having $O(n^{2/3})$ variables. By the above remarks, we can generate the language $\bigcup_{1 \leq i \leq s} B_i \{a^{c_i}\}^*$ with a BNF grammar having $O(n^{2/3})$ variables. It follows that the same upper bound holds for a CNF grammar for $L(M) - \{\epsilon\}$.

3 The case of larger alphabets

Now we turn to the case of a fixed size, non-unary alphabet. As mentioned above, the standard construction for showing that any DFA M (or NFA) has an equivalent regular grammar [13, §9.1] gives an upper bound of $n + k$ variables on the size of a context-free grammar in CNF accepting $L(M) - \{\epsilon\}$.

In this section we obtain a lower bound. Our lower bound actually holds for the more specific case where the language consists of a single word.

Lemma 3.1 *There exists a constant* c *such that for all* $m \geq 1$ *there exists a language* L_m *accepted by a DFA with* $2^m + m + 1$ *states (or by an NFA with* $2^m + m$ *states) such that the smallest number of variables in any context-free grammar in CNF generating* L_m *is* $> c2^m/m$*.*

Proof. As is well-known, for all m there exists a string w_m of length 2^m+m-1 over $\{0,1\}$ such that every string of length m appears as a subword of w_m. These strings are sometimes called *de Bruijn* words [8,6]. Let $L_m = \{w_m\}$. Then clearly L_m can be accepted by a DFA with $2^m + m + 1$ states or an NFA with $2^m + m$ states.

We now argue that at least $c2^m/m$ variables are needed to generate L_m.

A *word chain* is a straight-line program to generate a word, where every instruction is of the form $A_i := a$, where $a \in \Sigma$ is a single letter, or $A_i := A_j A_k$, where $j, k < i$. The length of a word chain is the number of instructions.

It is easy to see that every n-variable CNF grammar $G = (V, \Sigma, P, S)$, with no useless symbols, generating $\{w\}$ corresponds to a word chain of length $n + |\Sigma|$ generating w [18].

Now a known result on word chains [1] says that a word chain of length $c2^m/m$ is needed to generate w_m. Our lower bound follows.

Corollary 3.2 *There exist constants* c, n_0 *such that for all* $n \geq n_0$ *there exists a DFA* M_n *having* n *states such that any CNF grammar* G *with* $L(G) = L(M_n)$ *has at least* $cn/(\log n)$ *variables.*

Results on word chains also imply that the $cn/(\log n)$ bound is tight for languages consisting of a single word [2].

4 Acknowledgments

We are grateful to the referees for several suggested improvements, including a simplification of the proof of Theorem 2.5.

References

[1] I. Althöfer. Tight lower bounds for the length of word chains. *Inform. Process. Lett.* **34** (1990), 275–276.

[2] J. Berstel and S. Brlek. On the length of word chains. *Inform. Process. Lett.* **26** (1987/88), 23–28.

[3] J.-C. Birget. Intersection and union of regular languages and state complexity. *Inform. Process. Lett.* **43** (1992), 185–190.

[4] M. Chrobak. Finite automata and unary languages. *Theoret. Comput. Sci.* **47** (1986), 149–158.

[5] J. Currie, H. Petersen, J. M. Robson, and J. Shallit. Separating words with small grammars. *J. Automata, Languages, and Combinatorics* **4** (1999), 101–110.

[6] N. G. de Bruijn. A combinatorial problem. *Proc. Konin. Neder. Akad. Wet.* **49** (1946), 758–764.

[7] J. Goldstine, J. K. Price, and D. Wotschke. A pushdown automaton or a context-free grammar — which is more economical? *Theoret. Comput. Sci.* **18** (1982), 33–40.

[8] I. J. Good. Normal recurring decimals. *J. London Math. Soc.* **21** (1946), 167–169.

[9] J. Gruska. On a classification of context-free languages. *Kibernetika* **3** (1967), 22–29.

[10] J. Gruska. Some classifications of context-free languages. *Inform. Control* **14** (1969), 152–179.

[11] J. Gruska. Descriptional complexity of context-free languages. In *Proc. Math. Found. Computer Sci.*, pp. 71–83, 1973.

[12] J. Gruska. Descriptional complexity (of languages): a short survey. In A. Mazurkiewicz, editor, *Proc. 5th Symposium, Mathematical Foundations of Computer Science 1976*, Vol. 45 of *Lecture Notes in Computer Science*, pp. 65–80. Springer-Verlag, 1976.

[13] J. E. Hopcroft and J. D. Ullman. *Introduction to Automata Theory, Languages, and Computation.* Addison-Wesley, 1979.

[14] 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 Sciences*, pp. 263–267. 1973.

[15] A. R. Meyer and M. J. Fischer. Economy of description by automata, grammars, and formal systems. In *Proc. 12th Annual Symposium on Switching and Automata Theory*, pp. 188–191, 1971.

[16] M. J. Nederhof and G. Satta. IDL-Expressions: A compact representation for finite languages in generation systems. Manuscript, 2002.

[17] G. Pighizzini and J. Shallit. Unary language operations, state complexity, and Jacobsthal's function. To appear, *Internat. J. Found. Comput. Sci.*, 2002.

[18] J. Shallit and M.-w. Wang. Automatic complexity of strings. *J. Automata, Languages, and Combinatorics* **6** (2001), 537–551.

[19] S. Yu, Q. Zhuang, and K. Salomaa. The state complexities of some basic operations on regular languages. *Theoret. Comput. Sci.* **125** (1994), 315–328.