ON THE SUCCINCTNESS OF DIFFERENT REPRESENTATIONS OF LANGUAGES

by

J. Hartmanis

TR78-346
REVISED

Department of Computer Science
Cornell University
Ithaca, NY 14853

This research has been supported in part by National Science Foundation Grant DCR75-09433.

ON THE SUCCINCTNESS OF DIFFERENT REPRESENTATIONS OF LANGUAGES

J. Hartmanis

Department of Computer Science
Cornell University
Ithaca, NY 14853

Abstract

The purpose of this paper is to give simple new proofs of some interesting recent results about the relative succinctness of different representations of regular, deterministic and unambiguous context-free languages and to derive some new results about how the relative succinctness of representations change when the representations contain a formal proof that the languages generated are in the desired subclass of languages.

This research has been supported in part by National Science Foundation Grant DCR75-09433 and MCS 78-00418.

Introduction

It has been shown recently that there exist dramatic compression of the length of representations of languages in subclasses of context-free languages as we go from restricted to unrestricted representations of these languages [2,4,5]. For example, when we consider the representation of deterministic context-free languages by deterministic versus nondeterministic pushdown automata, then there is no recursive function which can bound the size of the minimal deterministic pushdown automaton as a function of the size of the equivalent minimal nondeterministic pushdown automaton [5]. It is well known that we cannot recursively decide whether a given pushdown automaton has a equivalent deterministic pushdown automaton, but the above result makes a considerably stronger statement: even if we would know (or be given) which pushdown automata describe deterministic languages, we still could not effectively write down the corresponding deterministic pushdown automata because of their enormous size which grows nonrecursively in the size of the nondeterministic pushdown automata. Therefore we see that though nondeterminism is not needed in the description of nondeterministic context-free languages its use in the description permits nonrecursively bounded shortening of infinitely many representations.

Similar results hold for the relative succinctness of the description of unambiguous cfl's by unambiguous and ambiguous cfg's [4], and the description of finite or regular sets by finite automata and pushdown automata [2].

Some of the original proofs of these results are quite hard and they require special results about context-free languages. In the first part of this paper we give a very simple, elementary proof that the relative succinctness of representing deterministic context-free languages by deterministic or nondeterministic pushdown automata is not recursively bounded, and using a result about inherently ambiguous cfl's and Turing machine computations [3], derive an equally simple proof for the representation of unambiguous cfl's by unambiguous or ambiguous context-free grammars. The results about the representation of finite and regular sets can be easily proven by the same methods.

In the representation of deterministic cfl's by deterministic pda's we can easily check whether a given pda is deterministic, on the other hand, for a nondeterministic pda we have no obvious way of verifying that it accepts a deterministic cfl. Therefore the question arises whether the relative succinctness of the two representations is caused by the fact that in one representation we can prove what we are accepting but that no such proofs are possible in the other representation.

Indeed a close inspection of the original proof [2] reveals that it does not hold when we represent dcfl's by dpda's or pda's with attached proofs that they accept deterministic cfl's.

In the second part of this paper we show that our proof techniques furthermore prove that, for example, the relative succinctness results hold for the representation of dcfl's by deterministic pda and nondeterministic pda with attached proofs that they accept deterministic cfl's.

Finally, to gain further insight how the inclusion of formal proofs of correctness in representations of languages affects their succinctness, we consider the representation of finite sets. We show that there is no recursive bound in the relative succinctness of the representation of finite sets by finite automata or Turing machines (even if we attach proofs that the Tm accepts a finite set). On the other hand, we show that the relative succinctness is recursively bounded for the representation of finite sets by finite automata or Turing machines with proofs which explicitly give the cardinality of the finite set accepted.

It follows from the results that the relative succinctness is not recursively bounded for the representation of finite sets by finite automata (or tables) or Turing machines which accept them, but that there is a recursive bound for the representation of finite sets by finite automata (or tables) and Turing machines which list them and halt.

Succinctness Results about CFL's

We first establish notation and summarize some well known facts about context-free languages (cfl's).

We denote pushdown automata (pda) by A_i and deterministic pushdown automata (dpda) by D_j. Let $|A_i|$ denote the length of the description of the automaton A_i over some finite alphabet and $L(A_i)$ the language accepted by A_i. We consider only one-tape Turing machines, denoted by M_i, and for technical reasons we assume (withou any loss of generality) that M_i can halt only after an even number of moves, M_i accepts by halting and that it makes at least two moves before halting, finally assume that M_i cannot print a blank. An instantaneous description of M_i depicts the symbols written on the tape, indicates the tape square scanned by M_i and its state; they are strings of the following form:

$$-\Sigma^*(a,q)\Sigma^*-, -(-,q)\Sigma^*- \text{ or } -\Sigma^*(-,q)-,$$

where - denotes a blank tape square, Σ is the finite alphabet of symbols M_i can print, $a \in \Sigma$ and q is a state of M_i. For Tm M_i $ID_0(x)$ denotes the instantaneous description of the starting configuration on input x and $ID_1(x), ID_2(x), \ldots$ denote the successive instantaneous descriptions of M_i on input x. If $x = a_1a_2\ldots a_n$ then $x^T = a_na_{n-1}\ldots a_2a_1$. Let $VALC[M_i]$ denote the set of valid computations of M_i in which every second instantaneous description is reversed, i.e.

$$VALC[M_i] = \{\#ID_0(x)\#[ID_1(x)]^T\#ID_2(x)\ldots\#[ID_{2k-1}(x)]^T\#ID_{2k}(x)\# \mid x \in \Sigma^* \text{ and } ID_{2k}(x) \text{ is a halting configuration}\}.$$

Let

$$INVALC[M_i] = \Gamma^* - VALC[M_i].$$

It is well known that $INVALC[M_i]$ can be accepted by a nondeterministic pda and therefore it is a cfl [1]. On the other hand, $VALC[M_i]$ is a cfl iff $L(M_i)$ is a finite set, since otherwise for arbitrarily large inputs x the three first instantaneous descriptions must be related and the cfl pumping lemma does not hold. This yields the well known auxiliary result.

<u>Lemma 1</u>: $INVALC[M_i]$ is a deterministic cfl iff $L(M_i)$ is finite.

<u>Proof</u>: If $L(M_i)$ is finite then $INVALC[M_i]$ is a regular set and therefore a dcfl. If $L(M_i)$ is infinite then $VALC[M_i]$ is not a cfl and therefore $INVALC[M_i]$ cannot be a dcfl. ∎

Lemma 2: The set $R = \{A_i \mid L(A_i)$ is not a dcfl$\}$ is not recursively enumerable.

Proof: Since INVALC$[M_i]$ is a deterministic cfl iff $L(M_i)$ is finite, a recursive enumeration of R would yield a recursive enumeration of the set $\{M_i \mid L(M_i)$ is infinite$\}$, which is seen not to be possible by Rice's theorem. ∎

For two representations, such as the representation of deterministic cfl's by deterministic and nondeterministic pda's, we will say that their relative succinctness is not recursively bounded if there does not exist a recursive function F such that for any pda, A, which accepts a deterministic cfl, there exists an equivalent deterministic pda, D, for which $|D| \le F(|A|)$.

Theorem 3: The relative succinctness of representing deterministic cfl's by deterministic and nondeterministic pda's is not recursively bounded.

Proof: If such a recursive F exists then for any pda A we can compute $F(|A|)$ and effectively list the dpda's whose length of description does not exceed $F(|A|)$, say $D_{i_1}, D_{i_2}, \ldots, D_{i_s}$. Then L(A) is a nondeterministic cfl iff none of the D_{i_j}, $1 \le j \le s$, is equivalent to A, but if this is so then we can detect it by comparing the D_{i_j} and A on successive inputs from Σ^*. Therefore the existence of F implies that the set

$$\{A \mid L(A) \text{ is not a dcfl}\}$$

is recursively enumerable, which we know is not the case by Lemma 2. Therefore, F does not exist as was to be shown.

Next we consider the relative succinctness between the representation of unambiguous cfl's by unambiguous and ambiguous cfg's.

We exploit a recent result (which is given in a somewhat different formulation in) [3]. For any Tm, M_i, let

$$A_S(M_i) = \{\#ID_0(x)\#([ID_j]^T\# \; ID_{j+1} \; \#)^* \mid$$
$$ID_{j+1} \text{ follows from } ID_j \text{ by one operation of } M_i,\ x \in \Sigma^*\}$$
$$A_E(M_i) = \{\#(ID_j \; \#[ID_{j+1}]^T\#)^* \; ID_{2k}\# \mid ID_{j+1} \text{ follows from } ID_j \text{ in}$$
$$\text{one operation of } M_i \text{ and } ID_{2k} \text{ is a halting configuration.}\}$$

and define

$$A(M_i) = A_S(M_i) \cup A_E(M_i)$$

It is easily seen that $A(M_i)$ is a context-free language and it links the ambiguity question for $A(M_i)$ to finiteness of sets accepted by the Turing machine M_i.

<u>Theorem 4</u>: $A(M_i)$ is an inherently ambiguous cfl iff $L(M_i)$ is infinite.

<u>Proof</u>: See [3]. ∎

<u>Theorem 5</u>: The relative succinctness of representing unambiguous cfl's by unambiguous and ambiguous cfg's is not recursively bounded.

<u>Proof</u>: If a recursive bound F exists, then the set

$$AMB = \{G \mid G \text{ cfg and } L(G) \text{ is inherently ambiguous}\}$$

is recursively enumerable. To see this note that we can list for any cfg G all cfg's whose representations are shorter than $F(|G|)$ and then cross off those grammars which are found to be ambiguous or not equivalent to G as we test them on successive strings from Σ^*. L(G) is inherently ambiguous iff eventually all grammars from the list are crossed off. Thus the set AMB is recursively enumerable and therefore, (by Theorem 4) so is the set

$$\{M_i \mid L(M_i) \text{ is infinite}\},$$

which leads to a contradiction. Therefore the recursive bound F does not exist.

By the same method we can give an easy proof for the next result [1].

Theorem 6: The relative succinctness of the representation of cofinite sets by finite automata and pushdown automata is not recursively bounded. Therefore the relative succinctness of the representation of regular sets by finite automata and pushdown automata is also not recursively bounded.

Proof: Similar to the proof of Theorem 3, by using the set

$$R = \{A_i \mid L(A_i) \text{ is not cofinite}\}. \qquad \blacksquare$$

The same reasoning shows that there is no recursive bound between the size of context-free grammars (which generate cfl's whose complements are also cfl's) and the size of the cfg's generating the complements.

Theorem 7: There is no recursive function F such that for any cfg G such that $\Sigma^* - L(G)$ is a cfl, there exists a cfg, G', with $L(G') = \Sigma^* - L(G)$ and $|G'| \leq F(|G|)$.

Proof: Similar to the proof of Theorem 3. ∎

Succinctness Results about Verified Representations

In the representation of deterministic cfl's by deterministic and nondeterministic pda's we can easily verify that a given automaton is indeed deterministic, but for the equivalent nondeterministic pda's we have no fixed way of verifying that it will accept a deterministic cfl. This lack of symmetry in our representations suggests that we should consider only representations by nondeterministic pda's with attached proofs that they accept a deterministic language and add the length of the proof to the length of the representation of the pda.

A close inspection of the original proofs [2,4,5] reveals that they do not extend to the representation with added proofs. On the other hand our proof techniques show that the previous succinctness results can be extended to representations with attached verifications that they accept the desired type of language.

More precisely, let FS be an axiomatizable, sound formal mathematical system which is powerful enough to express and prove elementary facts about Turing machines, context-free languages and pushdown automata. Since FS is axiomatizable we know that we can recursively enumerate the set of provable theorems and soundness assures as that the provable theorems are true. Instead of specifying FS in detail we will describe what must be easily provable in FS.

a) Let $M_{\sigma(r)}$ be a simply and uniformly constructed Tm which for each input x computes and saves the length of x, $|x| = n$; then enumerates all one-tape Tm's up to length r, i.e. $|M_i| \le r$, and simulates in a dowe-tail manner the com putations of this finite set of machines on blank tape. $M_{\sigma(r)}$ halts (and therefore accepts) iff some M_i, $|M_i| \le r$, halts after performing n or more steps. From this constru tion we see that for all r, $r \ge 1$, $M_{\sigma(r)}$ accepts a finite set. We assume that FS is sufficiently powerful that we can prove in FS that $L(M_{\sigma(r)})$ is finite and that the lengt of these proofs is recursively bounded in r.

b) We furthermore assume that there is a simple and uniform construction ρ which yields for each Tm M_i a pda $A_{\rho(i)}$ such that

$$L(A_{\rho(i)}) = \mathrm{INVALC}[M_i]$$

and that it can be proven in FS (by a proof whose length i recursively bounded in i) that:

if $L(M_i)$ is finite then $L(A_{\rho(i)}) = \mathrm{INVALC}[M_i]$ is a deterministic cfl.

From these assumptions it follows that we can prove (easily) in FS that:

$A_{\rho(\sigma(r))}$ accepts a deterministic cfl.

It should be observed that in any logic designed to reason about computations we should be able to formulate and prove easily the above result. Furthermore, to any given sound formal system w can add the above assertions as an axiom scheme to obtain the desired FS.

A nondeterministic pda with an attached proof in FS that it accepts a deterministic cfl is called a verified pda or vpda.

Theorem 8: The relative succinctness of representing dcfl's by dpda's and vpda's is not recursively bounded.

Proof: For r, $r \geq 1$, let $M_{\sigma(r)}$ be a Tm which accepts all inputs up to length N_r, where N_r is the maximal running time before halting achieved by a Tm of size r on blank tape. Let $A_{\rho(i)}$ be a non-deterministic pda which accepts INVALC$[M_i]$. It is assumed that $\sigma(r)$ and $\rho(r)$ are simple enough to compute and that FS is sufficiently rich that there exist short proofs (whose length is recursively bounded in r) that $L[M_{\sigma(r)}]$ is finite and therefore $L[A_{\rho(\sigma(r))}]$ is a deterministic cfl.

If there exists a recursive bound F between $|A_{\rho(\sigma(r))}|$ and the shortest equivalent dpda, then we can list all the dpda's

$$D_{i_1}, D_{i_2}, \ldots, D_{i_s}, \text{ such that } |D_{i_j}| \leq F[|A_{\rho(\sigma(r))}|],\ 1 \leq j \leq s.$$

From this list of dpda's we can effectively construct a list of dpda's which accept the complements of these languages. From this new list we can effectively select the dpda's which accept finite sets and compute the longest string accepted by these dpda's. Clearly the length of this string is bigger than N_r and therefore N_r is recursively bounded in r, which is a contradiction. ■

By exploiting the fact that we can recursively enumerate the vpda's we can prove the next result.

Corollary 9: There is no recursive succinctness bound between the representation of dcfl's by verified pda's and pda's.

By assuming that we can easily prove in FS relations between $A(M_i)$ and ambiguous cfl's we obtain the next result.

Corollary 10: There is no recursive succinctness bound between the representation of unambiguous cfl's by unambiguous cfg's and ambiguous cfg's with proofs that they accept unambiguous cfl's, nor is there a recursive bound between the representation of unambiguous cfl's by ambiguous cfg's with proofs that they accept unambiguous cfl's and cfg's.

Representation of Finite Sets

The situation changes drastically if we consider representation of finite sets and finite sets of known size.

Theorem 11: a) There is no recursive succinctness bound for the representation of finite sets by finite automata and by Tm's with proofs that they accept finite sets.

b) There is a recursive succinctness bound for the representation of finite sets by finite automata (or tables) and Tm's with proofs which explicitly give the size of the finite set accepted.

c) There is a recursive bound for the relative succinctness of representing finite subsets by finite automata (or lists) and Tm's with proofs that they print a list and halt.

Proof: a) Let $M_{\sigma(r)}$ be the Tm constructed for the proof of Theorem 8 and recall that we have assumed that our formal system FS is sufficiently rich to prove, by proofs whose length is recursively bounded in r, that $L(M_{\sigma(r)})$ is finite. Therefore the length of $M_{\sigma(r)}$ plus the length of the proof in FS that $L(M_{\sigma(r)})$ is finite is recursively bounded in r. On the other hand, since $L(M_{\sigma(r)})$ is finite the number of states of any finite automaton accepting $L(M_{\sigma(r)})$ must be no less than the length of the longest string in $L(M_{\sigma(r)})$, which by construction of $M_{\sigma(r)}$ is not recursively bounded in r. Therefore, the relative succinctness of these two representations cannot be recursively bounded.

b) The relative succinctness bound F can be constructed as follows. For n construct all proofs of "M_i accepts a set of size k", i,k = 1,2,..., such that $|M_i|$ plus the length of the proof is less or equal to n. For the M_i with such proofs let k_n be the cardinality of the largest set accepted and a_n the length of the longest string accepted. Clearly k_n and a_n are effectively computable and

$$F(n) = a_n \cdot k_n + 2$$

is such a recursive bound.

c) For any n we can effectively list the finite set of Tm's, $M_{i_1}, M_{i_2}, \ldots, M_{i_k}$, such that

$$|M_{i_j}| + |\text{proof that } M_{i_j} \text{ prints a list and halts}| \le n.$$

Therefore we can run all the Tm's on this list, which are guaranteed to halt because FS is sound, and determine the length of the longest string printed, n_m. Clearly n_m is recursively computable from n, by the above procedure, and, furthermore, the size of the largest minimal finite automaton accepting the sets $L(M_{i_1})$, $L(M_{i_2})$,..., $L(M_{i_k})$ is recursively bounded in n_m. Therefore the size of the finite automata representation of these sets is recursively bounded to n and therefore to the size of the Tm representation with proofs. ∎

References

1. Hartmanis, J. "Context-Free Languages and Turing Machine Computations", Proceedings of Symposia in Applied Mathematics Vol. 19 Mathematical Aspects of Computer Science, pp. 42-51, Amer. Math. Soc. 1967.

2. Meyer, A.R. and M.J. Fischer. "Economy of Description by Automat Grammars and Formal Systems", Conference Record, IEEE 12th Annua Symposium on Switching and Automata Theory (1971), 188-190.

3. Reedy, H. and W.J. Savitch. "The Turing Degree of the Inherent Ambiguity Problem for Context-Free Languages", Theoretical Computer Science, Vol. 1 (1975), 77-91.

4. Schmidt, E.H. and T.G. Szymanski. "Succinctness of Descriptions of Unambiguous Context-Free Languages", SIAM J. Computing, Vol. (1977), 547-553.

5. Valiant, L.G. "A Note on the Succinctness of Descriptions of Deterministic Languages", Information and Control, Vol. 32 (1976 139-145.