INFORMATION AND CONTROL 52, 2–7 (1982)

On the Recursion-Theoretic Complexity of Relative Succinctness of Representations of Languages

LOUISE HAY

Department of Mathematics, Statistics, and Computer Science, University of Illinois at Chicago, Chicago, Illinois 60680

1. INTRODUCTION

In Hartmanis (1980) a simple proof is given of the fact (originally proved in Valiant (1976)) that the relative succinctness of representing deterministic context-free languages by deterministic vs. nondeterministic pushdown automata is not recursively bounded, in the following sense: there is no recursive function which, for deterministic context-free languages L, can bound the size of the minimal deterministic pushdown automaton accepting L as a function of the size of a nondeterministic pushdown automaton accepting L. It is then stated that "...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." This does not, however, rule out *a priori* the possibility that a *partial* recursive bound might exist, as a function of the *description* of the nondeterministic pushdown automaton rather than of its size; indeed, the proof in Hartmanis (1980) uses the fact that the bounding function is total. It is the purpose of this note to make a case for using *partial* bounding functions in questions of relative succinctness. It will be shown that for the examples considered in Hartmanis (1980) (deterministic context-free languages, unambiguous languages, regular languages), the best possible partial bound as a function of the *description* of the unrestricted automata, while still nonrecursive, has lower recursion-theoretic complexity than the best possible bound as a functions of the *size* of the unrestricted automata. An example will be given of a class of languages for which there exists a partial recursive bound on the size of the restricted automata, again as a function of the *description* of the unrestricted automata, while no (total) recursive bound exists as a function of the *size* of the unrestricted automata.

0019-9958/82/010002–06$02.00/0
Copyright © 1982 by Academic Press, Inc.
All rights of reproduction in any form reserved.

2. Recursion-Theoretic Preliminaries

All notation is taken from Rogers (1967). In particular, if $\{M_i\}_{i \in N}$ is a standard enumeration of all Turing machines, $\mathbf{0}'$ denotes the Turing degree of the "halting problem" $K = \{i: M_i$ halts on input $i\}$. For any set $C \subseteq N$, C' = the halting problem relativized to $C = \{i: M_i^C$ halts on input $i\}$, where M_i^C denotes M_i with oracle set C, $\mathbf{0}''$ denotes the Turing degree of K', Σ_n^0, Π_n^0 denote the n-quantifier levels in the arithmetic hierarchy (Rogers, 1967, Chap. 14), $\leqslant_1$ denotes one–one reducibility, and $\leqslant_T$ Turing reducibility.

The following well-known recursion-theoretic facts will be used:

Lemma 1 (Rogers, 1967, Theorem 14-VIII). *If X is recursively enumerable in $\mathbf{0}'$, then $X \in \Sigma_2^0$.*

Lemma 2. *Let* INF $= \{M_i : L(M_i)$ *is infinite*$\}$. *Then*

(a) (Rogers, 1967, Theorem 13-VIII) INF *is Π_2^0-complete.*

(b) (Rogers, 1967, Theorem 14-VIII) *If X is Π_2^0-complete, $X \notin \Sigma_2^0$.*

Lemma 3. *Let φ be a partial function recursive in a set C (i.e., the graph of φ is recursively enumerable in C). Then φ has a total extension F recursive in C'.*

Proof. To compute $F(x)$, ask, recursively in C', whether $\varphi(x)$ converges. If yes, let $F(x) = \varphi(x)$. If no, let $F(x) = 0$.

Lemma 4 (Fixed-point theorem) (Rogers, 1967, Theorem 11-I). *If f is any recursive function, there exists n such that $L(M_n) = L(M_{f(n)})$.*

3. Relative Bounds

We consider automata of the usual types, with descriptions over some alphabet. For simplicity, we identify the automaton with its description when considering it as part of the domain for a function. It will be assumed that the length function $|A|$ is a recursive function of (the description of) A, and that there are finitely many automata of given length n, which can be effectively found given n.

Let C be a class of languages, with two types of repesentation $\mathscr{R}_1$, $\mathscr{R}_2$. Call a total function F a *bounding relation* between $\mathscr{R}_1$ and $\mathscr{R}_2$ if, for any automaton A of type $\mathscr{R}_1$ that accepts a language in C, there exists an equivalent automaton B of type $\mathscr{R}_2$ such that

$$\text{(a)} \quad |B| \leqslant F(A).$$

Call F a *bounding size relation* if (a) can be replaced by

$$\text{(a}'\text{)} \quad |B| \leqslant F(|A|).$$

Call a partial function φ a *partial bounding relation* between $\mathscr{R}_1$ and $\mathscr{R}_2$ if, for any (description A of an) automaton of type $\mathscr{R}_1$ that accepts a language in C, $\varphi(A)$ is defined and there exists an equivalent automaton B of type $\mathscr{R}_2$ such that

$$\text{(b)} \quad |B| \leqslant \varphi(A).$$

Call φ a *partial bounding size relation* if, for any automaton A of type $\mathscr{R}_1$ that accepts a language in φ, $\varphi(|A|)$ is defined and (b) can be replaced by

$$\text{(b}'\text{)} \quad |B| \leqslant \varphi(|A|).$$

PROPOSITION 5. *There is a bounding relation of given recursion-theoretic complexity (i.e., Turing degree) between $\mathscr{R}_1$ and $\mathscr{R}_2$ if and only if there is a bounding size relation of the same complexity.*

Proof. ($\leftarrow$). If $F$is a bounding size relation, then $G(A) = F(|A|)$ is a bounding relation of the same complexity, since the size function $|A|$ is recursive.

($\rightarrow$). If F is a bounding relation, let $G(n) = \max\{F(A)\colon |A| = n\}$. Then G has the same complexity as F and is a bounding size relation since $F(A) \leqslant G(|A|)$ for all A. ∎

It is therefore unnecessary to distinguish between bounding relations and bounding size relations *if the functions are total.* For partial functions, however, the situation is different. We still have, just as above.

PROPOSITION 6. *If φ is a partial bounding size relation between $\mathscr{R}_1$ and $\mathscr{R}_2$, then there is a partial bounding relation of the same complexity.*

However, for the other half we have only:

PROPOSITION 7. *If φ is a partial bounding relation between $\mathscr{R}_1$ and $\mathscr{R}_2$ and φ is dominated by total function F of the same recursion-theoretic complexity, then there is a (total) bounding size relation ψ of the same complexity.*

Proof. Assume that φ is a partial bounding relation, and that $\varphi(A) \leqslant F(A)$ for all $A \in$ domain φ. Then as above,

$$\psi(n) = \max\{F(A)\colon |A| = n\}$$

is a bounding size relation ψ of the same complexity. ∎

Note that for bounding *size* relations, if there is suitable "padding" available it may be assumed that all functions are total. There is reason to expect that in general when a partial bounding relation φ is replaced by a bounding size relation the complexity may increase, since (at least in the "naive" replacement) we must take a maximum over all $\varphi(A)$, where $|A| = n$, and for this one must know when $\varphi(A)$ converges.

4. Succinctness Results About Context-Free Languages

Following Hartmanis (1980) we denote pushdown automata by A_i and deterministic pushdown automata by D_j. Theorem 3 of Hartmanis (1980) can now be restated in our terms as follows:

If $C =$ the class of deterministic context-free languages, $\mathscr{R}_1 =$ the class of nondeterministic pushdown automata which accept a deterministic context-free language, and $\mathscr{R}_2 =$ the class of deterministic pushdown automata, then there does not exist a recursive bounding relation between $\mathscr{R}_1$ and $\mathscr{R}_2$.

This leaves open the possibility of a partial recursive *partial* bounding relation between $\mathscr{R}_1$ and $\mathscr{R}_2$. The question is settled as follows:

Theorem 8. *If $\mathscr{C} =$ the class of deterministic context-free languages, $\mathscr{R}_1 =$ the class of nondeterministic pushdown automata accepting a language in $\mathscr{C}$, and $\mathscr{R}_2 =$ the class of deterministic pushdown automata, the following hold for bounding relations between $\mathscr{R}_1$ and $\mathscr{R}_2$:*

(a) *There does not exist a bounding relation recursive in* $\mathbf{0}'$.

(b) *There exists a bounding relation recursive in* $\mathbf{0}''$.

(c) *There does not exist a (partial) recursive partial bounding relation.*

(d) *There exists a partial bounding relation recursive in* $\mathbf{0}'$.

Proof. Part (a) follows from a careful examination of the proof of Theorem 3 of Hartmanis (1980). If $ND = \{A_i : L(A_i)$ is not a deterministic context-free language$\}$, the proof of Lemma 1 of Hartmanis (1980) shows that $INF \leqslant_1 ND$. Now it is easily seen that $ND \in \Pi_2^0$, since

$$\begin{aligned} A_i \in ND &\leftrightarrow \forall j(L(A_i) \neq L(D_j)) \\ &\leftrightarrow \forall j\, \exists w(w \in \Sigma^* \;\&\; [(w \in L(A_i) - L(D_j)) \\ &\qquad \vee (w \in L(D_j) - L(A_i))]). \end{aligned}$$

Hence ND is Π_2^0-complete, which by Lemma 2(b) implies $ND \notin \Sigma_2^0$. Now the proof of Theorem 3 of Hartmanis (1980) actually shows that if F is any (total) bounding relation between $\mathscr{R}_1$ and $\mathscr{R}_2$, then ND is recursively enumerable in F. If there exists such an F recursive in $\mathbf{0}'$, this would make ND recursively enumerable in $\mathbf{0}'$ and hence by Lemma 1 would imply $ND \in \Sigma_2^0$, which is a contradiction. Part (c) now follows easily from Lemma 3, since a partial recursive (i.e., recursive in $\mathbf{0}$) partial bounding relation would have a total extension recursive in $\mathbf{0}'$; such an extension would be a bounding relation, contradicting part (a). For part (d), define φ as follows: given a nondeterministic pushdown automaton A, ask for each j in turn whether $L(D_j) = L(A)$, i.e., whether

$$(\forall w \in \Sigma^*)(w \in L(D_j)) \leftrightarrow w \in L(A)).$$

This is a 1-quantifier question, hence recursive in $\mathbf{0}'$. If $L(A)$ is a deterministic context-free language, the answer will be yes for some j_0, and we can then define $\varphi(A) = |D_{j_0}|$. Part (b) now follows by Lemma 3 (or, more directly, since $ND \in \Pi_2^0$ is recursive in $\mathbf{0}''$, to define $F(A)$, ask recursively in $\mathbf{0}''$ whether $A \in ND$. If yes, let $F(A) = 0$; if no, apply the algorithm above used for $\varphi(A)$ to compute $F(A)$). ∎

Note that the function φ exhibited in the proof of part (d) provides a "natural" example (i.e., one occurring in nature) of a partial function recursive in $\mathbf{0}'$ which has no total extension recursive in $\mathbf{0}'$ (and in fact, by Proposition 7, which is not dominated by any total function recursive in $\mathbf{0}'$).

Exactly analogous results hold for the complexity of the relative succinctness of the representation of unambiguous context-free languages by unambiguous and ambiguous context-free grammars (Schmidt and Szymanski, 1977) and for the representation of regular sets by finite automata and pushdown automata (Meyer and Fischer, 1971) this is seen by applying the above analysis to the proofs of Theorems 5 and 6 of Hartmanis (1980).

5. Succinctness Results About k-Element Sets

We now show that there exists a partial recursive succinctness bound for the representation of k-element sets by finite automata and by unrestricted Turing machines, but that there does not exist a total recursive succinctness bound (and hence none as a function of the *size* of the Turing machine representation). More precisely, we have

Theorem 9. *For $k \geqslant 1$, let C_k = the class of k-element sets over the alphabet Σ, $\mathscr{R}_1$ = the class of all Turing machines, $\mathscr{R}_2$ = the class of finite automata (or tables) accepting exactly k elements of Σ^*. The following then holds:*

(a) *there is a partial recursive partial bounding relation between* $\mathscr{R}_1$ *and* $\mathscr{R}_2$; *but*

(b) *there is no (total) recursive bounding relation between* $\mathscr{R}_1$ *and* $\mathscr{R}_2$.

Proof. For part (a), define φ as follows: Let $\{S_i\}_{i \in N}$ be some effective one-to-one enumeration of the k-element subsets of Σ^*, and let $\{P_i\}_{i \in N}$ be an enumeration of all finite automata. Let f be a recursive function such that $P_{f(i)}$ is a finite automaton that accepts S_i and nothing else. If M is any Turing machine, dovetail computations of M on elements of Σ^* until M has accepted exactly k elements, say S_i. If this ever happens, define $\varphi(M) = |P_{f(i)}|$. This evidently defines a partial recursive partial bounding relation, since $P_{f(i)}$ is equivalent to M if $L(M) = S_i$. For part (b), let F be any (total) recursive function; we shall effectively construct a Turing machine M which accepts a k-element set but such that $F(M)$ does not bound the minimal finite automaton accepting $L(M)$. Given any Turing machine M_n, construct a Turing machine M' as follows: Let S' = the first k elements in some enumeration of Σ^* which are not in $\bigcup_i \{L(P_i): |P_i| \leqslant F(M_n)\}$. S' can be found effectively from n, and so can a Turing machine M' which accepts exactly S'; hence there is a recursive function h such that $M' = M_{h(n)}$. By the fixed-point theorem (Lemma 4) there exists some n such that $L(M_n) = L(M_{h(n)})$, and M_n is evidently the desired Turing machine M. ∎

RECEIVED: June 15, 1981; REVISED: March 22, 1982

REFERENCES

HARTMANIS, J. (1980), On the succinctness of different representations of languages, *SIAM J. Comput.* **9**, 114–120.

MEYER, A. R. AND FISCHER, M. J. (1970), Economy of descriptions by automata, grammars and formal systems, "Conference Record, IEEE 12th Annual Symposium on Switching and Automata Theory," 188–190.

ROGERS, JR., H. (1967), "Theory of Recursive Functions and Effective Computability," McGraw-Hill, New York.

SCHMIDT, E. H., AND SZYMANSKI, T. G. (1977), Succinctness of descriptions of unambiguous context-free languages, *SIAM J. Comput.* **6**, 547–553.

VALIANT, L. G. (1976), A note on the succinctness of descriptions of deterministic languages, Inform. Contr. **32**, 139–145.