RELATIVE SUCCINCTNESS OF REPRESENTATIONS OF LANGUAGES AND SEPARATION OF COMPLEXITY CLASSES

by

Juris Hartmanis

TR 78-349

Department of Computer Science
Cornell University
Ithaca, NY 14853

This research has been supported in part by National Science Foundation grant MCS 78-00418.

RELATIVE SUCCINCTNESS OF REPRESENTATIONS OF LANGUAGES AND SEPARATION OF COMPLEXITY CLASSES

Juris Hartmanis

Department of Computer Science
Cornell University
Ithaca, NY 14853

Abstract

In this paper we study the relative succinctness of different representations of polynomial time languages and investigate what can and cannot be formally verified about these representations. We also show that the relative succinctness of different representations of languages is directly related to the separation of the corresponding complexity classes; for example, PTIME $\neq$ NPTIME if and only if the relative succinctness of representing languages in PTIME by deterministic and nondeterministic clocked polynomial time machines is not recursively bound which happens if and only if the relative succinctness of these representations is not linearly bounded.

This research has been supported in part by National Science Foundation grant MCS 78-00418.

1. Introduction

In this paper we study the relative succinctness of different representations of languages with particular attention to the representation of polynomial time languages. We relate the problems of relative succinctness of representations of languages to the classic problem of separating the corresponding complexity classes and investigate what can and cannot be formally verified about the representations of polynomial time languages.

Let $M_1, M_2, \ldots$ be a standard enumeration of all deterministic multitape Turing machine (Tm's), let $L(M_i)$ denote the language accepted by M_i, let $|x|$ denote the length of x and define

$$T_i(n) = \max\{m \mid m \text{ is number of operations performed by } M_i \text{ on input } x,\ |x| = n\}.$$

Similarly, let $N_1, N_2, \ldots$ be a standard enumeration of all nondeterministic multitape Tm's.

Let the set of polynomial time Tm's be denoted by

$$PM = \{M_i \mid T_i(n) \le kn^k,\ k \ge 1\}$$

and let

$$PTIME = \{L(M_i) \mid M_i \in PM\}.$$

Similarly, let NPTIME denote the set of languages accepted in polynomial time by nondeterministic Turing machines.

The set PTIME consists of all languages accepted by deterministic Tm's in polynomial time and the main object of this paper is to investigate different representations of this family of languages.

A standard way of representing languages in PTIME is by <u>clocked</u> Tm's which are Tm's with standard form subroutines which for input x count up to $k|x|^k$ and at this count override the Tm and reject the input and halt. It is assumed that these clockes are in an easy to recognize form so that the set of clocked Tm's is recursive; denote it by

$$CM = \{M_{\sigma(i,k)} \mid i,k = 1,2,\ldots\}.$$

If we let

$$LANG[CM] = \{L[M_{\sigma(i,k)}] \mid i,k = 1,2,\ldots\} = \{L[M_j] \mid M_j \in CM\}$$

then, clearly

$$PTIME = LANG[CM].$$

In a way, the class of clocked machines, CM, can be viewed as a set of machines for which we can easily supply proofs that they run in polynomial time. A more general class of machines running in polynomial time can be obtained by considering all the machines for which we can prove that they run in polynomial time. To make this idea precise let F be an axiomatizable sound formal mathematical system in which we can prove elementary facts about Turing machines. If S is a formal statement provable in F we write

$$F \vdash [S].$$

Let the class of <u>verified machines</u>, VM, be given by

$$VM = \{M_i \mid F \vdash [M_i \text{ runs in polynomial time}]\}$$

and let

$$\mathrm{LANG}[\mathrm{VM}] = \{L(M_i) \mid M_i \in \mathrm{VM}\}.$$

For a sufficiently powerful sound formal system F we have that

$$\mathrm{LANG}[\mathrm{CM}] = \mathrm{LANG}[\mathrm{VM}] = \mathrm{PTIME}.$$

Finally, we can represent the sets in PTIME by nondeterministic polynomial time Tm's. Let CNP denote the class of <u>clocked nondeterministic machines</u> which accepts sets in PTIME,

$$\mathrm{CNP} = \{N_{\rho(i,k)} \mid L(N_{\rho(i,k)}) \in \mathrm{PTIME}\}.$$

and let VCN denote the class of clocked nondeterministic polynomial time machines for which we can prove in F that they accept sets in PTIME,

$$\mathrm{VCNP} = \{N_{\rho(i,k)} \mid F \vdash [L(N_{\rho(i,k)}) \in \mathrm{PTIME}]\}.$$

Again we note that for sufficiently strong sound formal system F

$$\mathrm{LANG}[\mathrm{CNP}] = \mathrm{LANG}[\mathrm{VCNP}] = \mathrm{PTIME}.$$

For the representations of sets in PTIME by machines from CM and PM we say that the <u>relative succinctness is not recursively bound</u> if there is no recursive function in the size of machines in PM which can bound the size of the minimal equivalent machines in CM. Similarly the definition is extended to other pairs of representations.

In the first part of this paper we show that the relative succinctness of representing sets in PTIME by machines from CM to VM cannot be recursively bounded and that the same is true for representations from VM and PM.

The relative succinctness between deterministic and nondeterministic representations of polynomial time computations is directly related to the classic problem PTIME= NPTIME? The relative succinctness of representing PTIME languages by clocked deterministic polynomial time machines and clocked nondeterministic polynomial time machines (CM and CNP) is recursively bounded iff PTIME = NPTIME. As a matter of fact, we show that PTIME = NPTIME if and only if the relative succinctness of the deterministic and nondeterministic representations can be linearly bounded. This result establishes an interesting relation between the relative succinctness of representations and the separation of the corresponding complexity classes, thus linking the dynamic and static properties of programs. This result can be easily extended to other separation problems including the classic problem about deterministic and nondeterministic context-sensitive languages as well as polynomial time and polynomial tape languages [1,3,4].

In the last part of this paper we consider what can and cannot be proven in F about these representations. We show that for the simple representation of sets in PTIME by the clocked machines, CM, we can prove in F that we get the desired class of languages, namely

$$F \vdash [\mathrm{LANG}[\mathrm{CM}] = \mathrm{PTIME}].$$

On the other hand, though we gain in succinctness of representations by going from CM to verified machines, VM, we loose our ability to prove in F that we are getting the right family of languages. We show that

$$\mathrm{LANG}[\mathrm{VM}] = \mathrm{PTIME}$$

is not provable in F. Thus, allowing the full power of F to select machines for which we can prove in F that they run in polynomial time prevents us from verifying in F that we have defined the right class of languages.

2. Succinctness Results

In this section we show that the relative succinctness between the representation of PTIME languages by machines in CM, VM and PM is not recursively bounded.

Theorem 1: The relative succinctness of representing languages in PTIME by machines in CM and PM is not recursively bounded.

Proof: Let B(r) denote the maximum number of steps performed before halting on blank tape by Tm's of size r. Clearly B(r) is not a recursive function and B(r) cannot be recursively bounded in r.

Below we construct a Tm $M_{p(r)}$ which runs in polynomial time and accepts a finite set not accepted by any clocked machine, $M_{\sigma(i,k)}$ in CM, with

$$|M_{\sigma(i,k)}| < B(r).$$

Since the size of $M_{p(r)}$ will be seen to be recursively related to r we must conclude that the relative succinctness of these two representations is not recursively bounded.

In the construction of $M_{p(r)}$ we use an auxilliary list, L_n, of machines in CM which have been found not to be equivalent to $M_{p(r)}$ by processing inputs up to length n-1.

Description of $M_{p(r)}$: Let $L_1 = \emptyset$, reject the null string and all inputs not in 0*. For $w = 0^n$, $n \geq 1$, lay off $\lceil \log n \rceil$ tape squares and in this amount of tape try to find the maximal N of number steps performed by a Tm of size r before halting; clearly $N \leq B(r)$ and eventually (for sufficiently large n) an N, N = B(r) will be found. (Note that this computation takes a polynomial number of steps.) If no N is found (on the available tape) w is rejected. If a value is found let N be the maximum. Reconstruct the list L_n, and try to find the smallest $M_{\sigma(i,k)}$ in CM such that

$$M_{\sigma(i,k)} \not\in L_n \text{ and } |M_{\sigma(i,k)}| < N.$$

If no such $M_{\sigma(i,k)}$ can be found on the available tape then reject input and set $L_{n+1} = L_n$. If $M_{\sigma(i,k)}$ is found then simulate $M_{\sigma(i,k)}$ on 0^n and do the opposite and set $L_{n+1} = L_n \cup \{M_{\sigma(i,k)}\}$.

It is seen that $M_{p(r)}$ runs in polynomial time and that $|M_{p(r)}|$ is recursively bounded in r, but that the size of the smallest equivalent $M_{\sigma(i,k)}$ is not recursively bounded in r. This completes the proof. ■

For sufficiently powerful axiomatizable formal systems F in which we can prove that $M_{p(r)}$ runs in polynomial time, namely

$$M_{p(r)} \in VM,$$

we can prove our next result:

<u>Lemma 2</u>: The relative succinctness of representing sets in PTIME by machines in CM and VM is not recursively bounded.

It turns out that also the representations VM and PM have nonrecursive succinctness.

<u>Theorem 3</u>: The relative succinctness of representing sets in PTIME by machines in VM and PM is not recursively bounded.

<u>Proof</u>: By a construction similar to the one in the proof of Theorem 1, recalling that VM is recursively enumerable. ■

3. Nondeterministic Representations

In this section we consider the relative succinctness of representing languages in PTIME by deterministic and nondeterministic clocked polynomial time Turing machines. We show that the relative succinctness of these representations is directly linked to the separation of the corresponding complexity classes, PTIME and NPTIME. In the following proof we exploit a technique used in [2].

Theorem 4: If PTIME $\neq$ NPTIME then the relative succinctness of representing sets in PTIME by machines in CM and CNP is not recursively bounded.

Proof: If a recursive succinctness bound G exists for these representations, then we can recursively enumerate the set

$$R = \{N_{\rho(i,k)} \mid L(N_{\rho(i,k)}) \in \text{NPTIME} - \text{PTIME}\}$$

by listing all the $M_{\sigma(j,t)}$ with

$$|M_{\sigma(j,t)}| < G[|N_{\rho(i,k)}|]$$

and checking on successive inputs whether $M_{\sigma(j,t)}$ and $N_{\rho(i,k)}$ disagree, if $N_{\rho(i,k)}$ disagrees with all listed $M_{\sigma(j,t)}$ then $N_{\rho(i,k)}$ has no deterministic polynomial time equivalent machine and it accepts a set in NPTIME - PTIME. But, if NPTIME $\neq$ PTIME then set R is not recursively enumerable, as shown in the next lemma, and therefore a recursive succinctness bound G does not exist. ∎

Lemma 5: If NPTIME $\neq$ PTIME then the set

$$R = \{N_{\rho(i,k)} \mid L(N_{\rho(i,k)}) \in \text{NPTIME} - \text{PTIME}\}$$

is not recursively enumerable.

<u>Proof</u>: To show that R is not recursively enumerable let $A \in$ NPTIME - PTIME. Construct $N_{\delta(r)}$ so that the input x is accepted iff M_r on blank tape has not halted in $|x|$ steps and $x \in A$. Clearly $N_{\delta(r)}$ operates in nondeterministic polynomial time and accepts a set not in PTIME iff M_r does not halt on blank tape. Therefore, a recursive enumeration of the set R would yield a recursive enumeration of

$$\{M_r \mid M_r \text{ does not halt on blank tape}\}$$

which is a contradiction. ∎

The next result extends the previous theorem to representations of sets in PTIME by machines from VM and CNP.

<u>Theorem 6</u>: If PTIME $\neq$ NPTIME then relative succinctness of the representation of sets in PTIME by machines in VM and CNP is not recursively bounded.

<u>Proof</u>: Similar to the proof of Theorem 4. ∎

The results in Theorems 4 and 6 can easily be extended to the representations of sets in PTIME by machines from CM and VCNP.

From the previous results we see that if PTIME $\neq$ NPTIME then the use of nondeterministic algorithms to describe deterministic programs can yield recursively unbound succinctness. Therefore, even if we know that a nondeterministic polynomial time algorithm has an equivalent deterministic polynomial time algorithm we may not be able to use the deterministic polynomial time algorithm because of its immense size. Clearly, an equivalent deterministic algorithm which does not have to run in polynomial time can be effectively computed from the nondeterministic algorithm and its length will be recursively related to the length of the nondeterministic algorithm

So far it is not known whether there exists nonrecursive succinctness between the representation of sets in PTIME by machines from PM and CNP.

Next we show that the nonrecursive succinctness given by Theorems 4 and 6 occurs iff PTIME ≠ NPTIME and that, furthermore, PTIME = NPTIME iff the relative succinctness can be bounded linearly.

__Lemma 7:__ If PTIME = NPTIME then there exists a recursive mapping F which maps every $N_{\rho(i,k)}$ onto an equivalent $M_{\sigma(j,t)}$ and there are two constants c_1 and c_2 such that

$$c_1|N_{\rho(i,k)}| + c_2 \geq |F(N_{\rho(i,k)})| = |M_{\sigma(j,t)}|.$$

__Proof:__ Note that the language

$$U = \{N_{\rho(i,k)} \# w(\#^{|N_{\rho(i,k)}|n^k}) \mid N_{\rho(i,k)} \text{ accepts } w\}$$

is a complete language for NPTIME [4]. If PTIME = NPTIME then U is accepted by some $M_{\sigma(j,t)} = M_{i_0}$ and therefore for every $N_{\rho(i,k)}$ we can write down a deterministic polynomial time machine $F(N_{\rho(i,k)})$ which for any input w first writes down

$$N_{\rho(i,k)} \# w(\#^{|N_{\rho(i,k)}|n^k})$$

and then starts M_{i_0} on this input. It is easily seen that

$$L[F(N_{\rho(i,k)})] = L(N_{\rho(i,k)})$$

and that for $c_2 = |M_{i_0}|$ and a $c_1 > 0$

$$c_1|N_{\rho(i,k)}| + c_2 \geq |F(N_{\rho(i,k)})| = |M_{\sigma(j,t)}|,$$

as was to be shown (the constant c_1 can be computed easily if we fix a representation of Tm's). ■

Combining these results we get a direct relation between the PTIME = NPTIME? problem and the problem of succinctness of deterministic representation of sets in PTIME.

<u>Theorem 8</u>: PTIME ≠ NPTIME iff the relative succinctness of representing languages in PTIME by deterministic and nondeterministic clocked polynomial time machines (CM and CNP) is not recursively bounded and this happens iff the relative succinctness is not linearly bounded.

<u>Proof</u>: Follows from Theorem 4 and Lemma 7. ■

This result (Theorem 8) can easily be extended to other separation problems such as deterministic and nondeterministic context-sensitive languages, PTIME and PTAPE and many others [1,3,4].

The above theorem could give further insights into the classic separation problems. First of all, if we could show for arbitrarily large B that the size of a deterministic equivalent of some nondeterministic machines $N_{\rho(i,k)}$ must exceed

$$c_1|N_{\rho(i,k)}| + B$$

then we would have shown that PTIME ≠ NPTIME. Similarly, if we could show that B in the above equation must be atleast of a given size then we would have a lower bound for the size of the polynomial time machine (which may not exist) M_{i_0} which recognizes the set U (Lemma 7); if B is very large then so must be M_{i_0}, indicating that it will be very hard to find.

4. An Observation about Optimal Algorithms

By techniques similar to the ones used in the previous section we can easily prove that for tape or time bounded computations a "slight" increase in the resource bound permits a recursively unbounded shortening of the representations. We state just one special case of this general result, let TAPE[t(n)] denote the set of languages accepted on t(n) TAPE (1].

Theorem 9: The relative succinctness of representing languages in TAPE[n^2] by machines using n^2-tape and machines using $n^{2+\epsilon}$-tape, $\epsilon > 0$, is not recursively bounded.

This result has some interesting implications for optimization of algorithms. It is well known that we can not recursively decide whether, for example, algorithms running on n^3-tape have equivalent algorithms running on n^2-tape. At the same time, there is a feeling that for any particularly important n^3-tape algorithm by hard work and cleverness we will find either a proof that a faster algorithm does not exist or find a faster algorithm. The above result shows that this may not always be the case. There are n^3-tape algorithms for which equivalent n^2-tape algorithms exist and we may even be able to prove that they exist (!), but we cannot ever obtain them because of their immense size. It is not the lack of cleverness or or the weakness of our formal mathematical system (which we are willing to change) which prevents us from using the fast algorithms for these computations, it is their enormous physical size which makes them inaccessible to us.

Similarly, if PTIME $\neq$ NPTIME and we use nondeterministic machines (programs) to specify deterministic polynomial time computations we may not be able to ever write down the equivalent deterministic polynomial time algorithm because of their size.

It is not clear whether any such functions with "short" inefficient algorithms whose optimal algorithms are impractically long are of practical interest. On the other hand, we may be able to prove for some "natural" computation that it has fast algorithms but that their length must exceed a large bound, as it was done in our proofs for the "unnatural" sets constructed to show the existence of recursively unbounded relative succinctness for these representations.

5. Provable Properties of Representations

In a sound and sufficiently powerful axiomatizable formal system F we can prove for every "clocked" TM $M_{\sigma(i,k)}$ in CM that it runs in polynomial time, namely

$$F \vdash [T_{\sigma(i,k)}(n) \leq kn^k].$$

Furthermore, for M_i such that $T_i(n) \leq kn^k$ we can prove in F that

$$L(M_{\sigma(i,k)}) = L(M_i).$$

Therefore, we can prove in F that the clocked machines accept exactly the family of languages PTIME.

Theorem 10: $F \vdash$ [LANG[CM] = PTIME].

Proof: As outlined above. ∎

On the other hand, we now show that the corresponding statement for verified machines, LANG[VM] = PTIME, is not provable in F. This proof was suggested by Neil Immerman of Cornell University and utilizes the fact that the consistency of F cannot be proven in F.

Theorem 11: It is not provable in F that

LANG[VM] = PTIME.

Proof: Let PT(M) be the predicate in F which asserts that M runs in polynomial time. Let TH(x) be the predicate in F which asserts that x is a theorem in F, i.e. there exists a y which proves x, PR(x,y). Let CONSIST(F) be a formal assertion in F that F is consistent. If F is sufficiently powerful then we can formulate diagonalization in F and prove that there are computations which cannot be computed in polynomial time, i.e.

$$1) \quad F \vdash [(\exists M_0)[L(M) = L(M_0) \Rightarrow \neg PT(M)]].$$

We will show that if we can prove in F that LANG[VM] = PTIME then we can prove in F that some statement in F is not a theorem, which proves in F that F is consistent, i.e. $F \vdash$ [CONSIST(F)], which is forbidden by Goedel's second incompleteness theorem.

Note that

LANG[VM] = PTIME

is equivalent to the two statements that

PTIME $\subseteq$ LANG[VM] and LANG[VM] $\subseteq$ PTIME, which can be formulated in F in the following way:

2) $PT(M) \Rightarrow (\exists M')[L(M') = L(M) \text{ and } TH[PT(M')]]$ and $TH[PT(M)] \Rightarrow PT(M)$.

If

$$F \vdash [LANG[VM] = PTIME]$$

then from 2) and the fact that

$$F \vdash [A \text{ and } B] \Rightarrow F \vdash [B]$$

we get

$$F \vdash [TH[PT(M)] \Rightarrow PT(M)]$$

from which we get by taking the contrapositive

3) $F \vdash [\neg PT(M) \Rightarrow \neg TH[PT(M)]]$.

From 1) we have that

4) $F \vdash [\neg PT(M_0)]$.

Using the fact that

$$[F \vdash [A] \text{ and } F \vdash [A \Rightarrow B]] \Rightarrow F \vdash [B]$$

and 3) and 4) we get

$$F \vdash [\neg TH(M_0)].$$

But this asserts that we can prove in F that something is not a theorem in F which is equivalent to

$$F \vdash \text{CONSIST}[F].$$

But it is impossible to prove the consistency of F in F for a sound F. Therefore,

$$\text{LANG}[VM] = \text{PTIME}$$

is not provable in F.

References

1. Aho, A.V., J.E. Hopcroft and J.D. Ullman. "The Design and Analysis of Computer Algorithms," Addison-Wesley, Reading, Massachusetts, 1974.

2. Hartmanis, J. "On the Succinctness of Different Representations of Languages", Technical Report TR78-346, Department of Computer Science, Cornell University, Ithaca, NY 14853.

3. Hartmanis, J. and H.B. Hunt, III. "The LBA Problem and Its Importance in the Theory of Computing", SIAM-AMS Proceedings Volume 7 (1974), pp. 1-26.

4. Hartmanis, J. and J. Simon. "On the Structure of Feasible Computations", Advances in Computers Volum 14, Morris Rubinoft and Marshall C. Yovits, eds. 1-43. Acadmic Press, New York 1976.