Alternation and the power of nondeterminism

Extended abstract

Ravi Kannan
M.I.T.

Section 1 Introduction and statement of results

While nondeterminism is widely beleived to be more powerful than determinism in various contexts (the most famous being the conjecture that NP strictly contains P), no proof of the added power of nondeterminism is available for any significant issue. The weaker conjecture (than NP strictly contains P) that there is a language accepted by a nondeterministic linear time bounded multitape Turing Machine that cannot be accepted by a deterministic linear time bounded multi-tape TM still seems quite hard (Paul 1982). The aim of this paper is to show how the existance of the polynomial-time hierarchy of Meyer and Stockmeyer(1972) and the related concept of alternation (Chandra, Kozen and Stockmeyer(1981)) can be exploited to prove the power of nondeterminism over determinism in some contexts. It is hoped that this approach may be useful in proving stronger results.

The basic idea is as follows : Under some conditions if nondeterminism is no more powerful than determinism then neither is a fixed number of alternations (see the Chandra, Kozen and Stockmeyer(1981)- the reader is assumed to be familiar with this paper). Using this and the fact that under these conditions, a fixed number of alternations (usually no more than 4) is indeed more powerful than determinism, one proves by contradiction the power of nondeterminism over determinism. Besides TM's the paper deals with various weaker models of computation: one-tape TM's, 2-way finite automaton . A one-tape TM consists of one read only tape and one read-write tape.It my be deterministic or nondeterministic. For the standard definitions of finite automaton the reader is referred to any usual text book, for example Hopcroft and Ullmann (1974). We assume that in all models the input on the read only tape comes with a left and a right end marker.

The main theorems proved in the paper are listed below:

The first theorem basically asserts that there is in general a non-polynomial blow-up of space while going from nondeterminism to determinism.(cf Remark 2, for the relation of this result to Savitch's theorem.)

Theorem 1 : For any positive integer k, there is a fully-space constructable function $S(n)$ with $S(n) \geq (1/2)loglog(n)$ infinetly often and a nondeterministic $S(n)$ space bounded Turing Machine which cannot be simulated by any $O(S(n)^{(log(S(n)))^k})$ space bounded deterministic Turing Machine .

Definition (of "simulated by") : We say that a Turing Machine M (which is a Σ_l machine for some fixed l) which is $S(n)$ space bounded is simulated by a Turing Machine M' if for each w in $\{0,1\}^*$ and each letter w_0 of w and for each t in $\{0,1\}^{S(length(w))}$, M started at w_0 with w on the input tape and with t on its work tape accepts iff M' started at w_0 on w with t in the first $S(length(w))$ cells of its work tape accepts.

Remark 1 : Note that M' may take space more than $S(n)$. For $S(n) \geq log(n)$ and $T(n) \geq S(n)$, it is possible to show that each $S(n)$ space bounded NDTM can be simulated by a $T(n)$ space bounded DTM iff NSPACE($S(n)$) is contained in DSPACE($T(n)$) .Such a result ,that is, containement of language classes implying capability of simulation under our definition does not seem possible to prove for smaller functions. If it was we would have the stronger theorem that concluded that DSPACE($S(n)^{log(S(n))^k}$) does not contain NSPACE($S(n)$).

Remark 2: Savitch's theorem (1970) asserts that if $S(n)$ is at least log(n) and is constructable, then a $S(n)$ space bounded NDTM can be simulated by a

$(S(n)^2)$ space bounded DTM. The functions $S(n)$ of our theorem are therefore obviously less than log(n). Thus our theorem asserts that Savitch's theorem in the above form cannot be extended below log space.

Theorem 2 : For any function $S(n)$ with $S(n) \geq c.loglog(n)$ for some constant c and $\lim(S(n)/\log(n)) = 0$, there is a $O(S(n))$ space bounded nondeterministic Turing machine which cannot be simulated by any $O(S(n))$ space bounded deterministic Turing machine.

Theorem 3 : For each natural number k, there is a 2-way nondeterministic finite automaton with s states which cannot be simulated by any 2-way deterministic finite automaton with $s^{(log(s))^k}$ or less states.

Remark 1 : This is the analog of theorem 1 for constant space. The definition of simulated by is similar-we require the simulating automaton to produce correct results whereever on the input it is started. Again we do not seem to be able to prove the stronger result that there is a **language** accepted by a 2-way nondeterministic finite automaton with s states that cannot be accepted by a 2-way deterministic finite automaton with $s^{log(s)^k}$ or less states. Such a nonpolynomial blow-up in the number of states was conjectured by Sakoda and Sipser (1978) and proved for the special case of sweeping automata by Sipser (1979).

Remark 2 : Whereas one can prove that any language accepted by a 2-way nondeterministic finite automaton can also be accepted by a 2-way deterministic finite automaton, it is not clear whether every 2-way nondeterministic finite automaton can be simulated by a 2-way deterministic finite automaton according to our definition. We conjecture that the statement is true.

Theorem 4: There is a language accepted by a ND multitape TM in linear time which cannot be accepted by a one-tape DTM in O($n^{1.104}$) .

Remark 1 : Note that regular crossing sequence arguments will not prove lower bounds for one-tape TM's. For example, pallindromes can be accepted in linear time by these machines.

Remark 2 : The theorem says in very rough terms that given one more work tape (since 2-tape NDTM's are as powerful as multi-tape ones (Book, Greibach and Wegbreit (1970)), and nondeterminism, we can save a nontrivial amount of time. The power of just one extra tape has been proved only for on-line computations (Paul(1982), Aandera(1974)) and for real time computations on NDTM's by Duris and Galil(1982).

Remark 3 : Note that Theorem 4 is not provable by naive diagonalization since the linear time bounded NDTM does not have enough time to simulate and negate O($n^{1.104}$) time bounded DTM's.

Remark 4 : The curious constant 1.104 is roughly the fourth root of 3/2 and the proof of course makes it clear how it comes into the picture.

Theorem 5: If for some k greater than or equal to 2, every language accepted in time $O(n^k)$ by a detrministic multi tape TM can also be accepted by a one-tape NDTM in time $O(n^{2k-\epsilon})$ for some $\epsilon > 0$ then there exists a language that can be accepted in linear time by a multitape NDTM but not by a multitape DTM.

Remark 1: The conclusion of the theorem would acheive the seperation of nondeterminism from determinism for a powerful model of computation-multitape TM's. The hypothesis gives us a 1-tape NDTM to simulate a multitape DTM. Though the best simulation we know of a multitape by a one-tape machine results in square loss of time, (Hartmanis and Sterns(1965)), here we have two advantages- nondeterminism and one read tape in addition to the work tape. However as the statement of the theorem should make clear the author has not been able to prove the hypothesis.

Section 2 Outline of proofs

First we describe the proof of theorem 3. We use alternating finite automaton described in Chandra, Kozen and Stockmeyer (1981) We consider the familiar "substring matching language " (from the same paper) :

S.M. = { x&w : w is in $\{0,1\}^s$, x is in $\{0,1,2\}^*$ and 2w2 is a substring of x}

where s is a fixed constant.

It is easily seen that any 1-way deterministic finite automaton accepting S.M. must have at least 2^{2^s} states. Hence by a result of Shepherdson(1959) which states that any language accepted by a 2-way deterministic finite automaton with z states can be accepted by a 2-way deterministic finite automaton with z^z states, we have that any 2-way deterministic finite automaton accepting S.M. must have at least $2^{\sqrt{(s)}}$ states. But on the other hand a Σ_2 finite automaton with $O(s)$ states accepts S.M. as follows : It moves to the right existentially guessing at some 2 that 2w2 has begun. It universally branches off to check that the i th bit of the ensuing 0,1 string is equal to the i th bit of the string following &. To do all this it clearly needs only $O(s)$ states. Finally Theorem 3 follows from the translation lemma below :

Translation Lemma : If p(.) is such that any 2-way nondeterministic finite automaton with z states can be simulated by a 2-way deterministic finite automaton with p(z) states or less then each 2-way Σ_2 finite automaton with y states can be simulated by a 2-way deterministic finite automaton with p(p(y)) states.

This lemma makes crucial use of our stronger definition of simulation - starting at an arbitrary point on the input string rather than the usual starting point of the left most end.

The proof of Theorem 1 is harder but uses the same ideas. Here we let s vary. If n is the length of the whole input string, we let w be of length roughly log(n). Then it is shown that a 1-way DTM accepting the language requires $O(n)$ space and hence a 2-way DTM requires $O(log(n))$ space. But again a Σ_2 TM can accept the language in roughly $O(loglog(n))$ space. The proof of this is complicated by the fact that log log(n) is not space-constructible (to my knowledge). This difficulty is circumvented by using a known function that is at most c.log log(n) for all n and is infinitely often at least log log(n)/2. (see for example Hopcroft and Ullmann(1979) - problem 12.14c). The detailes are left to the final paper. Finally, the proof of Theorem 1 is completed by a translation argument similar to the one above. We caution that padding arguments - a natural proof technique cannot be used for such low space. A careful statement of the translation lemma leads to a proof without using padding.

Theorem 2 is proved using the same line of argument but with a different translation lemma. Unfortunately it is not clear how to extend these arguments to log space.

To prove Theorem 4, we use the alternating TM's of Chandra, Kozen and Stockmeyer(1981). We refer to a machine that makes at most i-1 alternations starting with an existential state as a Σ_i machine. To prove Theorem 4 , we first prove Theorem 6 below :

Theorem 6: If L is a language accepted by a one-tape DTM in time $O(n^2)$, then L is accepted by a multitape Σ_4 machine in time $O(n^{4/3}log(n))$.

Proof: There are two main ideas involved in the proof of this theorem. The first one is that the n^2 work tape cells used by the one-tape DTM can be divided into blocks of roughly n cells each so that the machine crosses a block bondary at most n out of the n^2 steps it takes. (This idea has been used in Paterson(1972) and Hopcroft and Ullmann(1968).) We can thus guess existantially $O(n)$ length crossing sequence and having done this, one can universally branch off to check that the crossing sequence guessed is correct on each of the blocks. Now since we are confining attention to only one block, we have to simulate a $O(n^2)$ time bounded and $O(n)$ space bounded machine. This can be done efficiently (i.e., in time less than $O(n^2)$) by a Σ_2 machine using the time-alternation trade-off technique found in Kannan(1981). The proof has to carefully balance the block size and the time-alternation trade-off to acheive the stated result.

Ackowledgements : I wish to thank Merrick Furst and Mike Sipser for useful discussions.

References

S.O.Aandera, "On k tape vs k-1 tape real time computation" in Complexity of Computation SIAM-AMS Proceedings Ed. R.M.Karp (1974)

R.V.Book, S.A.Greibach and B.Wegbreit, "Time and tape bounded Turing acceptors and AFL's" JCSS 4:6 (1970)

A.Chandra, D.Kozen and L.Stockmeyer, "Alternation JACM Vol.28 No.1 (1981)

P.Duris and Z.Galil, "Two tapes are better than one for nondeterministic machines" STOC(1982)

J.Hartmanis and L.Sterns, "On the computational complexity of algorithms" Trans. AMS 117(1965)

F.C.Hennie, "One tape off-line Turing machine computations" Information and Control 8:6, pp553-578

J.Hopcroft and J.D.Ullmann, "Introduction to Automata theory, languages and computation " Addison-Wesley (1979)

J.Hopcroft and J.D.Ullmann, "Relation between time and tape complexities "JACM 15(13) (1968)

R.Kannan, "Towards seperating nondeterminism from determinism" STOC(1981)

A.Meyer and L.Stockmeyer, "The equivalence problem for regular expressions with squaring requires exponential space" STOC (1972)

M.S.Patterson, "Tape bounds for time bounded Turing machines" JCSS 6 (1972) pp116-124

W.Paul, "On-line simuation of k+1 tapes by k tapes requires nonlinear time" FOCS(1982)

W.Paul and R.Reischuk,"On alternation II" Acta Informatica 14 (1980)

W.J.Sakoda and M.Sipser, " Nondeterminism and the size of two-way finite automata" STOC(1978)

J.Savitch," Relationship between nondeterministic and deterministic tape complexities" JCSS 4:2 (1970) pp177-192

J.C.Shepherdson, "The reduction of two-way automata to one-way automata" IBM Journal of research and developement (1959) pp198-200

M.Sipser," Lower bounds on the size of sweeping automata" STOC(1979)

Supported by NSF Grant No. MCS 81-05557