Nondeterministic Space is Closed Under Complementation

Neil Immerman*

Computer Science Department
Yale University
New Haven, CT 06520

SIAM Journal of Computing (17:5) (1988), 935-938.

Introduction

In this paper we show that nondeterministic space $s(n)$ is closed under complementation, for $s(n)$ greater than or equal to $\log n$. It immediately follows that the context-sensitive languages are closed under complementation, thus settling a question raised by Kuroda in 1964 [9]. See Hartmanis and Hunt [4] for a discussion of the history and importance of this problem, and Hopcroft and Ullman [5] for all relevant background material and definitions.

The history behind the proof is as follows. In 1981 we showed that the set of first-order inductive definitions over finite structures is closed under complementation [6]. This holds with or without an ordering relation on the structure. If an ordering is present the resulting class is P. Many people expected that the result was false in the absence of an ordering. In 1983 we studied first-order logic, with ordering, with a transitive closure operator. We showed that NSPACE[$\log n$] is equal to (FO + pos TC), i.e. first-order logic with ordering, plus a transitive closure operation, in which the transitive closure operator does not appear within any negation symbols [7]. Now we have returned to the issue of complementation in the light of recent results on the collapse of the log space hierarchies [10, 2, 14]. We have shown that the class (FO + pos TC) is closed under complementation. Our

*Research supported by NSF Grant DCR-8603346.

main result follows. In this paper we give the proof in terms of machines and then state the result for transitive closure as Corollary 3. The question of whether (FO + pos TC) *without ordering* is closed under complementation remains open.

Our work in first-order expressibility led to our proof that nondeterministic space is closed under complementation. However, because first-order expressibility classes are not directly relevant to the proofs in this paper, we omit those definitions here. The interested reader should refer to [7] for all these definitions. Note that the proof of Theorem 3.3 in [7] is more complicated than the proof of Theorem 1, but quite similar to it. The same is true of the proof in [6] that the first-order inductive formulas are closed under complementation.

Results

Theorem 1 *For any $s(n) \geq \log n$,*

$$\mathit{NSPACE}[s(n)] = \mathit{co\text{-}NSPACE}[s(n)]\,.$$

Proof We do this by two lemmas. We will show that counting the exact number of reachable configurations[1] of an NSPACE[$s(n)$] machine can be done in NSPACE[$s(n)$] (Lemma 2). Lemma 1 says that once this number has been calculated we can detect rejection as well as acceptance. Note the similarity between Lemma 1 and a similar result about census functions in [12].

Lemma 1 *Suppose we are given an NSPACE[$s(n)$] machine M, a size $s(n)$ initial configuration, START, and the exact number N of configurations of size $s(n)$ reachable by M from START. Then we can test in NSPACE[$s(n)$] if M rejects.*

Proof Our NSPACE[$s(n)$] tester does the following. It initializes a counter to 0, and a target configuration to the lexicographically first string of length $s(n)$. For each such target either we guess a computation path of M from START to target, and increment both counter and target; or we simply

[1] The *configuration* of a Turing machine is the contents of its work tapes, the positions of its heads, and its state. Note that for $s(n) \geq \log n$, the number of possible configurations is less than $c^{s(n)}$ for some constant c, and thus can be written in $O[s(n)]$ space.

increment target. For each target that we have found a path to, if it is an accept configuration of M then we reject. Finally, if when we are done with the last target the counter is equal to N, we accept; otherwise we reject. Note that we accept iff we have found N reachable configurations, none of which is accepting. (Suppose that M accepts. In this case there can be at most $N-1$ reachable configurations that are not accepting, and our machine will reject. On the other hand, if M rejects then there are N non-accepting reachable configurations. Thus our nondeterministic machine can guess paths to each of them in turn and accept.) That is we accept iff M rejects. □

Lemma 2 *Given START, as in Lemma 1, we can calculate N – the total number of configurations of size $s(n)$ reachable by M from START – in NSPACE[$s(n)$].*

Proof Let N_d be the number of configurations reachable from START in at most d steps. The computation proceeds by calculating N_0, N_1, and so on. By induction on d we show that each N_d may be calculated in NSPACE[$s(n)$]. The base case $d = 0$ is obvious.

Inductive step. Given N_d we show how to calculate N_{d+1}. As in Lemma 1 we keep a counter of the number of $d+1$ reachable configurations, and we cycle through all the target configurations in lexicographical order. For each target we do the following: Cycle through all N_d configurations reachable in at most d steps, again we find a path of length at most d for each reachable one, and if we don't find all N_d of them then we will reject. For each of these N_d configurations check if it is equal to target, or if target is reachable from it in one step. If so then increment the counter, and start on target+1. If we finish visiting all N_d configurations without reaching target, then just start again on target+1 without incrementing the counter. When we've completed this algorithm for all targets our counter contains N_{d+1}. Since N is bounded above by $c^{s(n)}$ for some constant c, the space needed is O[$s(n)$].

To complete the proof of the lemma and the theorem note that N is equal to the first N_d such that $N_d = N_{d+1}$. □

Remark: In our original statement of Theorem 1 we made the assumption that $s(n)$ is space constructible. However, the standard definition of a nondeterministic Turing machine having space complexity $s(n)$ is that, "... no sequence of choices enables it to scan more than $s(n)$ cells ...," [5]. Thus, the above proof works even if $s(n)$ is not space constructible. We just let $s(n)$ increase as needed.

The following corollary is immediate:

Corollary 1 *The class of context sensitive languages is closed under complementation.*

Proof Kuroda showed in 1964 that CSL = NSPACE$[n]$ [9]. □

The k^{th} level of the log space alternating hierarchy (Σ_kALOG) is defined to be the set of problems accepted by alternating log space Turing machines that make at most $k-1$ alternations and begin in an existential state. Recently Lange, Jenner, and Kirsig [10] showed that this hierarchy collapsed to the second level, Σ_2ALOG. This result was then extended by several authors [2, 14] who showed that the log space oracle hierarchy collapses to L^{NL}. Here L=DSPACE$[\log n]$, and NL = NSPACE$[\log n]$. The logspace oracle hierarchy is given by $\Sigma_1\mathrm{OLOG} = NL$, and $\Sigma_{k+1}\mathrm{OLOG} = NL^{\Sigma_k\mathrm{OLOG}}$. In the case of the polynomial time hierarchy, the oracle and alternating hierarchies are identical, but they appeared to be different in the log space case. We knew that the logspace oracle hierarchy is equal to (FO + TC). This, together with the above results, led us to expect Theorem 1. The following is again immediate.

Corollary 2 *The Log Space Alternating Hierarchy and the Log Space Oracle Hierarchy both collapse to NSPACE$[\log n]$.*

In [7] we showed that NL is equal to (FO + pos TC). In Theorem 3.3 of [7] we also showed that any problem in NL may be expressed in the form $\mathrm{TC}[\phi](\overline{0}, \overline{\max})$ where ϕ is a quantifier free first-order formula, and 0 and max are constant symbols. It now follows that the same is true for the class (FO + TC).

Corollary 3

1. *NSPACE$[\log n]$ = (FO + pos TC) = (FO + TC) .*
2. *Any formula in (FO + TC) may be expressed in the form $TC[\phi](\overline{0}, \overline{\max})$ where ϕ is a quantifier free first-order formula.*

Michael Fischer has observed that one can now diagonalize nondeterministic space and thus easily prove a tight hierarchy theorem for nondeterministic space. Although Corollary 4 is not new, our techniques give a much simpler proof than was previously known. (See Chapter 12 in [5] for the old proof.)

Corollary 4 *For any tape constructible* $s(n) \geq \log n$*,*

$$\lim_{n \to \infty} \frac{t(n)}{s(n)} = 0$$

implies

$$\mathit{NSPACE}[t(n)] \neq \mathit{NSPACE}[s(n)] \ .$$

Conclusions and Directions for Future Work

Most of the interesting questions concerning the power of nondeterminism remain open. We still do not know whether nondeterministic space is equal to deterministic space, or whether Savitch's Theorem [15] is optimal. It is interesting to consider whether our proof method can be extended to answer these questions, or to tell us anything new about nondeterministic time.

Soon after we proved Theorem 1, Tompa et. al. [1] gave two extensions: they proved that LOG(CFL) – the set of problems log space reducible to a context free language – is closed under complementation, and they showed that Symmetric Log Space (cf. [11, 13]) is contained in ZPLP, "... the class of errorless probabilistic Turing machines running in $O[\log n]$ space and polynomial expected time." We suggest the following open problems:

1. Is (FO without $\leq$ + pos TC) closed under complementation?

2. Is Symmetric Log Space, equivalently (FO + pos STC), closed under complementation?

3. Is NL equal to a complexity class that was previously known to be closed under complementation, e.g., L, AC^1, or DSPACE$[\log^2 n]$?

4. In the proof of Theorem 1 we made use of the linear space compression theorem, Theorem 12.1 in [5]. Our actual construction multiplies the space bound by about eight. It is interesting to ask how much this can be reduced. Note in particular that if we could complement $\log n$ times, while only increasing the space bound by a constant factor, then it would follow that NL = AC^1.

Acknowledgements Thanks to Sam Buss, Mike Fischer, Lane Hemachandra, Steve Mahaney, and Joel Seiferas who contributed comments and corrections to this paper.

References

[1] A. Borodin. S.A. Cook, P.W. Dymond, W.L. Ruzzo, and M. Tompa, "Two Applications of Complementation via Inductive Counting," this volume.

[2] S. R. Buss, S.A. Cook, P. Dymond, and L. Hay, "The Log Space Oracle Hierarchy Collapses," in preparation.

[3] S.A. Cook, "A Taxonomy of Problems with Fast Parallel Algorithms," *Information and Control* **64** (1985), 2-22.

[4] J. Hartmanis and H.B. Hunt, III, "The LBA Problem," *Complexity of Computation,* (ed. R. Karp), *SIAM-AMS Proc. 7* (1974), 1-26.

[5] John E. Hopcroft and Jeffry D. Ullman, *Introdution to Automata Theory, Languages, and Computation,* Addison-Wesley (1979).

[6] N. Immerman, "Relational Queries Computable in Polynomial Time," *Information and Control,* 68 (1986), 86-104. A preliminary version of this paper appeared in *14th ACM STOC Symp.,* (1982), 147-152.

[7] N. Immerman, "Languages That Capture Complexity Classes," *SIAM J. Comput.* **16**, No. 4 (1987), 760-778. A preliminary version of this paper appeared in *15th ACM STOC Symp.,* (1983) 347-354.

[8] N. Immerman, "Expressibility as a Complexity Measure: Results and Directions," *Second Structure in Complexity Theory Conf.* (1987), 194-202.

[9] S.Y. Kuroda, "Classes of Languages and Linear-Bounded Automata," *Information and Control* **7** (1964), 207-233.

[10] K.J. Lange, B. Jenner, and B. Kirsig, "The Logarithmic Hierarchy Collapses: $A\Sigma_2^L = A\Pi_2^L$," *14th International Colloquium on Automata Languages, and Programming* (1987).

[11] H. Lewis and C. H. Papadimitriou, "Symmetric Space Bounded Computation," *ICALP* (1980). Revised version appeared in *Theoret. Comput. Sci.* **19** (1982),161-187.

[12] S. R. Mahaney, "Sparse Complete Sets for NP: Solution of a Conjecture of Berman and Hartmanis," *J. Comput. Systems Sci.* **25** (1982), 130-143.

[13] J. Reif, "Symmetric Complementation," *JACM* **31**, No. 2, April (1984), 401-421.

[14] U. Schöning and K.W. Wagner, "Collapsing Oracles, Census Functions, and Logarithmically Many Queries," Report No.140 (1987), Mathematics Institute, Univ. Augsburg.

[15] W.J. Savitch, "Relationships Between Nondeterministic and Deterministic Tape Complexities," *J. Comput. System Sci.* **4** (1970), 177-192.

[16] Róbert Szelepcsényi, "The Method of Forcing for Nondeterministic Automata," *Bull. European Association Theor. Comp. Sci.* (Oct. 1987), 96-100.

[17] Seinosuke Toda, "Σ_2SPACE(n) is Closed Under Complement," *JCSS* **35**, No.2 (1987), 145-152.