NONDETERMINISTIC SPACE IS CLOSED UNDER COMPLEMENT

Neil Immerman

YALEU/DCS/ TR 552
July 1987

Nondeterministic Space is Closed Under Complement

Neil Immerman*

Computer Science Department
Yale University
New Haven, CT 06520

July 21, 1987

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

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 complement [6]. This holds with or without an ordering relation on the structure. With an ordering 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 oracle hierarchies [10,1,14]. We have shown that the class (FO + pos TC) is closed under complement. Our main result follows. In this paper we give the proof in terms of machines and then state the result for transitive closure as Corollary 2. The question of whether (FO + pos TC) *without ordering* is closed under complement remains open.

---

*Research supported by NSF Grant DCR-8603346.

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

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

Proof We do this by two lemmas. We will show that counting the exact number of reachable configurations 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 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 cases $d = 0$ and $d = 1$ are 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}. Note that we have only used $O[s(n)]$ space.

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

The following is immediate:

Corollary 1 *cf. [1,10,14] The Log Space Alternating Hierarchy and the Log Space Oracle Hierarchy both collapse to NSPACE[$\log n$].*

Several corollaries and extensions of Theorem 1 have been observed by other authors. Sam Buss and Steve Cook independently showed that NL^* = NL [2]. Mike Fischer observed that one can now diagonalize nondeterminsitic space and thus easily prove a tight hierarchy theorem for nondeterministice space [4]. Martin Tompa showed that LOG(CFL) is also closed under complement [16].

In [7] we have shown that NSPACE[$\log n$] is equal to (FO + pos TC). Any problem in this class may be expressed in the form $TC[\varphi](\overline{0}, \overline{max})$ 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 2 *1. NSPACE[$\log n$] = (FO + pos TC) = (FO + TC) .*

2. Any formula in (FO + TC) may be expressed in the form $TC[\varphi](\overline{0}, \overline{max})$ where φ is a quantifier free first-order formula.

In [7] we also showed similar results for Symmetric Log Space using a symmetric transitive closure operator (STC), cf. [11,13]. It is easy to see that the proof of Theorem 1 remains true for Symmetric Space and thus,

Corollary 3 *1. For all constructible $s(n) \geq \log n$, SYM-SPACE[$s(n)$] = co-SYM-SPACE[$s(n)$].*

2. The Symmetric Log Space Alternating Hierarchy and the Symmetric Log Space Oracle Hierarchy both collapse to SYM-SPACE$[\log n]$.

3. SYM-SPACE$[\log n]$ = *(FO + pos STC) = (FO + STC)* .

4. Any formula in (FO + STC) may be expressed in the form $STC[\varphi](\overline{0}, \overline{\max})$ *where φ is a quantifier free first-order formula.*

Of course most of the interesting questions concerning the power of nondeterminism remain open. We still don't 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.

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

References

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

[2] S. R. Buss, "NL = NL*," personal communication (1987).

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

[4] M. J. Fischer, "An Easy Proof of the Tight Hierarchy Theorem for Nondeterministic Space," personal communication (1987).

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

[6] N. Immerman, "Relational Queries Computable in Polynomial Time," *14th ACM STOC Symp.,* (1982), 147-152. Also appeared in revised form in *Information and Control,* 68 (1986), 86-104.

[7] N. Immerman, "Languages Which Capture Complexity Classes," *15th ACM STOC Symp.*, (1983) 347-354. Also appeared in revised form in *SIAM J. Comput.* **16**, No. 4 (1987), 760-778.

[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] M. Tompa, "LOG(CFL) is Closed Under Complementation," personal communication (1987).