JOURNAL OF COMPUTER AND SYSTEM SCIENCES 23, 151–165 (1981)

Pebbling with an Auxiliary Pushdown

NICHOLAS PIPPENGER

IBM Research Laboratory, San Jose, California 95193

Received December 2, 1980; revised March 16, 1981

A pebble game on graphs is introduced which bears the same relationship to the ordinary pebble game as auxiliary pushdown machines bear to ordinary machines. The worst-case time-space trade-off for pebbling with an auxiliary pushdown is shown to be $T = N \exp \Theta(N/S)$ (where T denotes time, S denotes space and N denotes the size of the graph), which contrasts with $T = N \exp \exp \Theta(N/S)$ for ordinary pebbling. The significance of this result to various questions concerning relations among complexity classes is discussed.

1. INTRODUCTION

In 1970, Paterson and Hewitt [8] introduced a game played with pebbles on the vertices of an acyclic directed graph. A play of this game is a sequence of moves according to the following two rules.

Rule 1. If the immediate predecessors of a vertex all have pebbles on them, a pebble may be put on that vertex.

Rule 2. A pebble may be taken off a vertex.

A complete play of the game is one that starts with no pebbles on the graph and puts a pebble on each vertex at some move of the play. The game abstracts certain properties of computations, especially those dealing with time (reckoned as the number of moves in a play) and space (reckoned as the maximum number of pebbles on the graph at any move of a play).

This paper introduces and explores a variant of this pebble game in which there is available a pushdown stack capable of holding names of vertices in the graph. The pushdown stack is initially empty and may be manipulated according to the following two rules.

Rule 3. If there is a pebble on a vertex, the name of that vertex may be pushed on the stack.

Rule 4. If the name of a vertex is at the top of the stack, it may be popped off the stack and a pebble may be put on that vertex.

A complete play of the game is now one that starts with no pebbles on the graph and no names on the stack and that puts a pebble on each vertex at some move of the

0022-0000/81/050151–15$02.00/0

play. This variant of the pebble game, which we shall call the *auxiliary-pushdown pebble game*, bears to the ordinary pebble game the same relation as auxiliary-pushdown machines, introduced by Cook [1], bear to ordinary machines, in that there is available an auxiliary data storage medium that is not included in space bounds but that can be manipulated only in accordance with a first-in last-out discipline.

By a graph of *size* N we shall mean an acyclic directed graph with N vertices in which every vertex has at most two immediate predecessors (the number of immediate successors is not restricted). Hopcroft *et al.* [5] have shown that every graph of size N has a complete play of the pebble game in space $O(N/\log N)$. Paul *et al.* [10] have shown that there are graphs of size N for which every complete play requires space $\Omega(N/\log N)$. Proposition 2.1 of the present paper shows that the corresponding question for the auxiliary-pushdown pebble game is trivial: for every graph of size N there is a complete play of the game in space 3.

Lengauer and Tarjan [7] have shown that the worst-case trade-off between time T and space S for the pebble game on graphs of size N is $T = S \exp\exp \Theta(N/S)$. This is to be interpreted as two theorems. First, for every N and S in the range $R(N) \leqslant S \leqslant N$ (where $R(N) = \Theta(N/\log N)$) and for every graph of size N, there is a complete play of the game in space S and time $S \exp\exp O(N/S)$. Second, for every N and S in this range, there exists a graph of size N for which any complete play of the game in space S requires time $S \exp\exp \Omega(N/S)$. This time–space trade-off can be written

$$T/N = \exp\exp \Theta(N/S), \tag{1.1}$$

since a factor of N/S can be absorbed into the factor $\exp\exp \Theta(N/S)$.

Propositions 2.1 and 4.1 of the present paper show that the worst-case time-space trade-off for the auxiliary-pushdown pebble game is $T = S \exp \Theta(N/S)$. This can be written

$$T/N = \exp \Theta(N/S), \tag{1.2}$$

since a factor of N/S can also be absorbed into the factor $\exp \Theta(N/S)$.

After introducing an auxiliary pushdown into the pebble game, it is natural to consider the introduction of two or more auxiliary pushdowns. In a forthcoming paper we shall show that the worst-case time-space trade-off for two auxiliary pushdowns is $T = \Theta(N^2/S)$, which can be written

$$T/N = \Theta(N/S). \tag{1.3}$$

We conjecture that

$$T/N = \Theta(\log(N/S)) \tag{1.4}$$

for any fixed number of auxiliary pushdowns greater than or equal to three, and we have proved $T = O(N \log(N/S))$ in this case, but we have only the weaker lower bound $T = \Omega((N \log N)/\log S)$. Even this weaker bound, however, shows that (1.1), (1.2), (1.3) and (1.4) do not continue to $T/N = \Theta(\log \log(N/S))$ for four auxiliary pushdowns, *etc.*

Given an acyclic directed graph, we shall assign *levels* to its vertices as follows. If a vertex has no immediate predecessors, it is assigned the level 0. If a vertex has one or more immediate predecessors, it is assigned the level one greater than the maximum of the levels assigned to these immediate predecessors. This clearly assigns levels to all vertices in an acyclic directed graph. The *width* of such a graph is the maximum over all levels of the number of vertices that are assigned that level or a lower level and that have one or more immediate successors that are assigned a higher level.

Let **P** denote the class of languages recognizable by machines in time $n^{O(1)}$ (where n denotes the length of the input). Let **L** denote the class of languages recognizable by machines in space $O(\log n)$. It is easily seen that **L** is contained in **P**. It seems unlikely that **P** is contained in **L**, but no proof of this is presently known.

Some evidence that **P** is not contained in **L** can nevertheless be provided by the following argument. Consider the following Circuit Value Problem (CVP), due to Ladner [6]: Given an encoding of a Boolean combinational logic network together with Boolean values for its inputs, determine the Boolean value computed at its output. A string of length n can encode a network with $N = \Theta(n/\log n)$ gates. By a result of Pippenger [12], CVP can be recognized by a machine in time $O(N(\log N)^2) = O(n \log n)$; thus CVP certainly belongs to **P**.

A large class of algorithms for CVP comprises what may be called *pebbling algorithms*. Such algorithms determine the Boolean values computed by the gates of the network, maintaining and discarding these values in accordance with a complete play of the pebble game on the graph underlying the network. Cook [2] has shown that there are graphs of size N for which any complete play of the pebble game requires space $\Omega(N^{1/2}) = \Omega((n/\log n)^{1/2})$; Paul *et al.* [10] have improved this lower bound to $\Omega(N/\log N) = \Omega(n/(\log n)^2)$. Since these lower bounds exceed $O(\log n)$, CVP cannot be shown to belong to **L** by virtue of a pebbling algorithm. Since all presently known algorithms for CVP are pebbling algorithms, this may be interpreted as evidence that CVP does not belong to **L**, and thus that **P** is not contained in **L**.

Let **SC** denote the class of languages recognizable by machines in time $n^{O(1)}$ and simultaneously (that is, by the same machine) in space $(\log n)^{O(1)}$. Pippenger [13] has shown that **SC** is also the class of languages recognizable by uniform families of networks of size $n^{O(1)}$ and simultaneously of width $(\log n)^{O(1)}$. Let **NC** denote the class of languages recognizable by machines in time $n^{O(1)}$ and simultaneously in reversal $(\log n)^{O(1)}$. Pippenger [13] has shown that **NC** is also the class of languages recognizable by uniform families of networks of size $n^{O(1)}$ and simultaneously of depth $(\log n)^{O(1)}$. It seems unlikely either that **NC** is contained in **SC** or that **SC** is contained in **NC**. Since both **SC** and **NC** are contained in **P**, however, and since **L** is contained in both **SC** and **NC**, to establish either that **NC** is not contained in **SC** or that **SC** is not contained in **NC** would entail that **P** is not contained in **L**, a conjecture that, as we have already noted, has thus far eluded proof.

571/23/2-4

For evidence that **NC** is not contained in **SC**, consider the Shallow Circuit Value Problem (SCVP), which is simply CVP restricted to networks of depth $D \leqslant (\log n)^2$. By Proposition 3.3 of Pippenger [13], SCVP can be recognized in time $O(DN(\log N)^3) = O(n(\log n)^4)$ and simultaneously in reversal $O(D(\log N)^2) = O((\log n)^4)$; thus SCVP certainly belongs to **NC**. Paul and Tarjan [9] have shown that there are graphs of size N with depth $D \leqslant (\log N)^2 \leqslant (\log n)^2$ for which any complete play of the pebble game requires either space $\Omega(N/D) = \Omega(n/(\log n)^3)$ or time $\exp \Omega(D) = \exp \Omega((\log n)^2)$; a similar result follows from Proposition 4.1 of the present paper. Since these lower bounds exceed space $(\log n)^{O(1)}$ and time $n^{O(1)}$, SCVP cannot be shown to belong to **SC** by virtue of a pebbling algorithm. Since all presently known algorithms for SCVP are pebbling algorithms, this may be interpreted as evidence that SCVP does not belong to **SC**, and thus that **NC** is not contained in **SC**.

For evidence that **SC** is not contained in **NC**, consider the Narrow Circuit Value Problem (NCVP), which is simply CVP restricted to networks of width $W \leqslant (\log n)^2$. By Proposition 4.2 of Pippenger [13], NCVP can be recognized in time $O(WN \log N) = O(n(\log n)^2)$ and simultaneously in space $O(W \log N) = O((\log n)^3)$; thus NCVP certainly belongs to **SC**.

Ruzzo [16] has characterized **NC** as the class of languages recognizable by auxiliary-pushdown machines in space $O(\log n)$ and simultaneously in time $\exp(\log n)^{O(1)}$. A large class of algorithms for NCVP comprises what may be called *auxiliary-pushdown pebbling algorithms*. Such algorithms determine the Boolean values computed by the gates of the network, maintaining and discarding these values in accordance with a complete play of the auxiliary-pushdown pebble game. Proposition 4.1 of the present paper shows that there are graphs of size N and width $W \leqslant (\log N)^2 \leqslant (\log n)^2$ for which any complete play of the auxiliary-pushdown pebble game requires either space $\Omega(W) = \Omega((\log n)^2)$ or time $\exp \Omega(N/W) = \exp \Omega(n/(\log n)^3)$. Since these lower bounds exceed space $O(\log n)$ and time $\exp(\log n)^{O(1)}$, NCVP cannot be shown to belong to **NC** by virtue of an auxiliary-pushdown pebbling algorithm. Since all presently known algorithms for NCVP are auxiliary-pushdown pebbling algorithms, this may be interpreted as evidence that NCVP does not belong to **NC**, and thus that **SC** is not contained in **NC**.

For a final observation along these lines, let **LOGCFL** and **LOGDCFL** denote the classes of languages reducible by machines in space $O(\log n)$ to context-free languages and deterministic context-free languages, respectively. Cook [3] has shown that **LOGDCFL** is contained in **SC**, and Ruzzo [15] has shown that **LOGCFL** is contained in **NC**. In light of the observations in the preceeding paragraphs, it seems unlikely either that **SC** is contained in **LOGCFL** or that **NC** is contained in **LOGDCFL**.

The auxiliary-pushdown pebble game also has applications in comparative schematology, one of which was discussed in a preliminary version [14] of this paper.

In Section 2 we shall derive upper bounds to time and space for the auxiliary-pushdown pebble game. In Section 3 we shall show that under certain circumstances the availability of an auxiliary pushdown cannot reduce time or space by more than constant factors. We shall apply this result in Section 4, where we shall derive lower bounds to time and space for the auxiliary-pushdown pebble game.

2. The Upper Bound

In Proposition 2.1 we shall obtain upper bounds to time and space for the auxiliary-pushdown pebble game. It will be convenient to begin by introducing a suitable formalism.

If G is an acyclic directed graph and v is a vertex in G, $P_G(v)$ will denote the set of immediate predecessors of v in G and $P_G^*(v) = \{v\} \cup P_G(v) \cup P_G(P_G(v)) \cup \cdots$ will denote the set of all predecessors of v in G. If Π is a set of vertices in G, $P_G(\Pi)$ and $P_G^*(\Pi)$ will denote the union of $P_G(v)$ and $P_G^*(v)$, respectively, over all v in Π. When no confusion is possible, we may write $P(v)$, $P^*(v)$, $P(\Pi)$ and $P^*(\Pi)$ for $P_G(v)$, $P_G^*(v)$, $P_G(\Pi)$ and $P_G^*(\Pi)$, respectively.

Let V be the set of vertices of an acyclic directed graph. Let $V^\#$ denote the free commutative monoid generated by the elements of V. Elements of $V^\#$ may be regarded as subsets of V with repetitions allowed. The identity element of $V^\#$ will be denoted 0 and the sum of two elements A and B will be denoted $A + B$. When no confusion is possible, we may identify an element of V or a subset of V with the corresponding element of $V^\#$. If Π is an element of $V^\#$, $|\Pi|$ will denote the number of elements in Π with repetitions counted. (Elements of $V^\#$ will be used principally to represent configurations of pebbles. As the pebble game is usually played, there is never more than one pebble on a vertex, so a pebble configuration is represented by a set of vertices. It will be technically convenient, however, to allow more than one pebble on a vertex, and thus to represent a pebble configuration as a set with repetions (an element of $V^\#$). It is easy to see that optimal plays of the pebble game never have more than one pebble on a vertex, so this extension does not affect the significance of our results.)

Let V^* denote the free monoid generated by the elements of V. Elements of V^* may be regarded as strings over V. The identity element of V^* will be denoted ε and the product of two elements α and β will be denoted $\alpha\beta$. If π is an element of V^*, $\|\pi\|$ will denote the length of π. (Elements of V^* will be used principally to represent configurations of vertex names on the stack. The top of the stack is at the left, the bottom at the right.)

Let G be an acyclic directed graph with vertices V. A *configuration* (or position of the auxiliary-pushdown pebble game) for G is a pair (Π, π) comprising a *pebble configuration* Π in $V^\#$ and a *stack configuration* π in V^*. The configuration $(0, \varepsilon)$ will be called the *empty* configuration.

A *transition* (or move of the auxiliary-pushdown pebble game) for G is a pair of configurations for G of the form $c_v(\Pi, \pi) = [(\Pi + P(v), \pi), (\Pi + P(v) + v, \pi)]$, $d_v(\Pi, \pi) = [(\Pi + v, \pi), (\Pi, \pi)]$, $i_v(\Pi, \pi) = [(\Pi + v, \pi), (\Pi + v, v\pi)]$ or $j_v(\Pi, \pi) = [(\Pi, v\pi), (\Pi + v, \pi)]$ for some Π in $V^\#$, v in V and π in V^*. When Π and π are clear from context, these transitions will be denoted simply c_v, d_v, i_v and j_v. (These transitions represent moves according to Rules 1, 2, 3 and 4, respectively.)

A *calculation* (or play of the auxiliary-pushdown pebble game) for G is a sequence $(\Pi_0, \pi_0),\ldots, (\Pi_T, \pi_T)$ of configurations for G in which each consecutive pair $[(\Pi_t, \pi_t), (\Pi_{t+1}, \pi_{t+1})]$ of configurations constitutes a transition for G for $0 \leqslant t \leqslant T-1$. The calculations for G form a category $\mathscr{C}(G)$, the free category with the configurations for G as objects and the transitions for G as generating morphisms. When no confusion is possible, we may identify a configuration or transition with the corresponding calculation.

A calculation $f = [(\Pi_0, \pi_0),\ldots, (\Pi_T, \pi_T)]$ will be said to *start at* (Π_0, π_0) and to *finish at* (Π_T, π_T); this may be denoted f: $(\Pi_0 \pi_0) \to (\Pi_T, \pi_T)$. If f: $(A, \alpha) \to (B, \beta)$ and g: $(B, \beta) \to (\Gamma, \gamma)$ are calculations, their *composition* is $f \circ g$: $(A, \alpha) \to (\Gamma, \gamma)$ (note that composition is written in the order opposite to that customary for functions).

A calculation $[(\Pi_0, \pi_0),\ldots, (\Pi_T, \pi_T)]$ will be said to be *run through* a subset Π of V if every vertex in Π appears in at least one of the pebble configurations $\Pi_0,\ldots, \Pi_T$. It will be called *universal* if it runs through all of the vertices V.

A calculation $[(\Pi_0, \pi_0),\ldots, \Pi_T, \pi_T)]$ will be called *original* if it starts at the empty configuration. It will be called *complete* if it is both original and universal. It will be called *terminal* if it finishes at the empty configuration. It will be called *replete* if it is complete and terminal.

If $f = [(\Pi_0, \pi_0),\ldots, (\Pi_T, \pi_T)]$ is a calculation, the number T of transitions will be called the *time* of f and denoted $\|f\|$. The maximum number $\max\{|\Pi_0|,\ldots, |\Pi_T|\}$ of elements in any pebble configuration will be called the *space* of f and denoted $|f|$.

If G is an acyclic directed graph with vertices V, let $v_1,\ldots, v_N$ be a fixed enumeration of the vertices V in "topologically sorted order," that is, such that $P(v_n)$ is contained in $\{1,\ldots, v_{n-1}\}$ for $1 \leqslant n \leqslant N$. This order will henceforth be called the *standard order.*

If W is a subset of V and $|W| = K$, let $w_1,\ldots, w_k$ be the enumeration of the vertices of W in the standard order, and let w be the string $w_1 \cdots w_K$.

If A and B are sets, $A \backslash B$ will denote the set of elements in A but not in B.

For any Π in $V^{\#}$, W a subset of V and π in V^*, we now define the calculations $c_W(\Pi, \pi)$: $(\Pi + (p(W) \backslash W), \pi) \to (\Pi + (P(W) \backslash W) + W, \pi)$, $d_W(\Pi, \pi)$: $(\Pi + W, \pi) \to (\Pi, \pi)$, $i_W(\Pi, \pi)$: $(\Pi + W, \pi) \to (\Pi + W, w\pi)$ and $j_W(\Pi, \pi)$: $(\Pi, w\pi) \to (\Pi + W, \pi)$ to be $c_{w_1} \circ \cdots \circ c_{w_K}$, $d_{w_K} \circ \cdots \circ d_{w_1}$, $i_{w_K} \circ \cdots \circ i_{w_1}$, and $j_{w_1} \circ \cdots \circ j_{w_K}$, respectively. When Π and π are clear from context, these calculations will be denoted simply c_W, d_W, i_W and j_W.

Let $\mathscr{G}_2(N)$ denote the class of graphs of size N.

It is not hard to see how to construct a complete play of the auxiliary-pushdown pebble game for an arbitrary graph G of size N in space at most 3 and time at most $3 \cdot 2^N$: to get a pebble on a vertex, recursively get pebbles on its immediate predecessors, using the stack to facilitate recursion in the usual way. Lemmas 2.1 and 2.2 and Proposition 2.1 below formalize this simple idea, with one additional refinement: a looser space bound is exploited, if available, to obtain a tighter time bound.

LEMMA 2.1. *For every $K \geqslant 1$ and $L \geqslant 3$, every graph G in $\mathscr{G}_2(N)$ with $N \leqslant KL$ and every subset Π of the vertices V of G with $|\Pi| \leqslant K$, there is a calculation $f_\Pi: (0, \varepsilon) \to (\Pi, \varepsilon)$ for G in space at most $3K$ and time at most $3(2^{L-1}-2)K$.*

Proof. We shall construct f_Π by induction on L. If $L=3$, then G is in $\mathscr{G}_2(N)$ with $N \leqslant 3K$. Let $\Sigma = P^*(\Pi)$ comprise the predecessors of Π. Then $f_\Pi: (0, \varepsilon) \to (\Pi, \varepsilon)$ defined by $c_\Sigma \circ d_{\Sigma \backslash \Pi}$ is a calculation for G in space at most $|\Sigma| \leqslant 3K$ and time at most $2|\Sigma| \leqslant 6K$.

If $L \geqslant 4$, let V_1 and V_2 comprise the first $N-K$ and last K vertices of V in the standard order. Let $\Pi_1 = \Pi \cap V_1$ and $\Pi_2 = \Pi \cap V_2$.

Let $\Sigma_2 = P^*(\Pi_2) \cap V_2$ comprise the predecessors of Π_2 in V_2, so that $|\Sigma_2| \leqslant K$ and $\Sigma_2 = \Pi_2 \cup (P(\Sigma_2) \cap V_2)$. Let $\Sigma_1 = \Pi_1 \cup (P(\Sigma_2) \cap V_1)$. Then $|\Sigma_1 + |\Sigma_2| \leqslant |\Pi| + |P(\Sigma_2)| \leqslant |\Pi| + 2|\Sigma_2|$ so that $|\Sigma_1| \leqslant |\Pi| + |\Sigma_2| \leqslant 2K$.

Let $\Sigma_{1,1}$ and $\Sigma_{1,2}$ comprise the first $\min\{|\Sigma_1|, K\}$ and last $\max\{0, |\Sigma_1| - K\}$ vertices of Σ_1 in the standard order. Then $|\Sigma_{1,1}| \leqslant K$ and $|\Sigma_{1,2}| \leqslant K$.

The subgraph G_1 of G induced by V_1 is in $\mathscr{G}_2(K(L-1))$, so by inductive hypothesis there are calculations $f_{\Sigma_{1,1}}: (0, \varepsilon) \to (\Sigma_{1,1}, \varepsilon)$ and $f_{\Sigma_{1,2}}: (0, \varepsilon) \to (\Sigma_{1,2}, \varepsilon)$ for G_1 in space at most $3K$ and time at most $3(2^{L-2}-2)K$. Since $P(V_1)$ is contained in V_1, these may be regarded as calculations for G as well.

Let $\sigma_{1,1}$ be the string in V^* such that $i_{\Sigma_{1,1}}(\Sigma_{1,1}, \varepsilon) \to (\Sigma_{1,1}, \sigma_{1,1})$. Let $f'_{\Sigma_{1,2}}: (0, \sigma_{1,1}) \to (\Sigma_{1,2}, \sigma_{1,1})$ be the calculation obtained from $f_{\Sigma_{1,2}}$ by replacing each configuration (Π, π) by the configuration $(\Pi, \pi\sigma_{1,1})$.

Let $\Sigma = \Sigma_1 \cup \Sigma_2$, so that $|\Sigma| \leqslant 3K$. Then $|\Sigma_1 \backslash \Pi_1| + |\Sigma_2 \backslash \Pi_2| \leqslant |P(\Sigma_2)| \leqslant 2|\Sigma_2| \leqslant 2K$, so that $|\Sigma \backslash \Pi| \leqslant 2K$.

Thus $f_\Pi: (0, \varepsilon) \to (\Pi, \varepsilon)$ defined by $f_{\Sigma_{1,1}} \circ i_{\Sigma_{1,1}} \circ d_{\Sigma_{1,1}} \circ f'_{\Sigma_{1,2}} \circ j_{\Sigma_{1,1}} \circ c_{\Sigma_2} \circ d_{\Sigma \backslash \Pi}$ is a calculation for G in space at most $3K$ and time at most $3(2^{L-2}-2)K + 2|\Sigma_{1,1}| + 3(2^{L-2}-2)K + |\Sigma_{1,1}| + |\Sigma_2| + |\Sigma \backslash \Pi| \leqslant 3(2^{L-1}-2)K$. ∎

LEMMA 2.2. *For every $K \geqslant 1$ and $L \geqslant 3$ and every graph G with vertices V in $\mathscr{G}_2(N)$ with $N \leqslant KL$, there is a replete calculation g_V for G in space at most $3K$ and time at most $3 \cdot 2^L K$.*

Proof. We shall construct g_V by induction on L. If $L=3$, then G is in $\mathscr{G}_2(N)$ with $N \leqslant 3K$ and $g_V = c_V \circ d_V$ is a replete calculation for G in space at most $3K$ and time at most $2|V| \leqslant 6K$.

If $L \geqslant 4$, let V_1 and V_2 comprise the first $N-K$ and last K vertices of V in the standard order. The subgraph G_1 of G induced by V_1 is in $\mathscr{G}_2(K(L-1))$, so by inductive hypothesis there is a replete calculation g_{V_1} for G_1 in space at most $3K$ and time at most $3 \cdot 2^{L-1}K$. Since $P(V_1)$ is contained in V_1, this may be regarded as a calculation for G as well.

By Lemma 2.1, there is a calculation $f_{V_2}: (0, \varepsilon) \to (V_2, \varepsilon)$ for G in space at most $3K$ and time at most $3(2^{L-1}-2)K$. Thus $g_V = g_{V_1} \circ f_{V_2} \circ d_{V_2}$ is a replete calculation for G in space at most $3K$ and time at most $3 \cdot 2^{L-1}K + 3(2^{L-1}-2)K + |V_2| \leqslant 3 \cdot 2^L K$. ∎

Proposition 2.1. *For every $3 \leqslant S \leqslant N$ and every graph G in $\mathscr{G}_2(N)$, there is a complete calculation for G in space at most S and time at most $S \exp O(N/S)$.*

Proof. Immediate from Lemma 2.2 with $K = \lfloor S/3 \rfloor$ and $L = \lceil N/K \rceil$. ∎

3. On the Power of an Auxiliary Pushdown

In Proposition 3.1 we shall show that, under certain circumstances, the availability of an auxiliary pushdown cannot reduce time or space by more than constant factors. This proposition is a corollary of Lemma 3.2, which shows that if many vertices receive pebbles during a calculation but few have pebbles on their immediate predecessors, then the total worth of the items on the stack must decreasing during the calculation. Similarly, if the total worth of the items on the stack increases during a calculation, many vertices must have pebbles on their immediate predecessors during the calculation. To make these ideas precise, it is necessary to introduce a means for making the notions of "many" and "few" precise, which is done by the packing numbers defined below. It is also necessary to have a means of measuring the total worth of the items on the stack, which is done by means of the covering numbers. Both the packing number and the covering numbers are, in turn, defined in terms of the notion of diversity.

If G is an acyclic directed graph with vertices V, let λ denote the functor from $\mathscr{C}(G)$ to V^* that sends c_v into v and sends d_v, i_v and j_v into ε. (The functor λ extracts moves that put a pebble on a vertex by virtue of its immediate predecessors having pebbles, ignoring all other moves.) Let ν be the functor that sends c_v and j_v into v and sends d_v into i_v into ε. (The functor ν extracts moves that put a pebble on a vertex, ignoring all other moves.)

For every π in V^*, let $r(\pi)$ (the *diversity* of π) denote the number of distinct elements of V in π. The diversity is positive, $r(\pi) \geqslant 0$, increasing, $r(\alpha\beta) \geqslant r(\alpha)$ and $r(\alpha\beta) \geqslant r(\beta)$, and subadditive, $r(\alpha\beta) \leqslant r(\alpha) + r(\beta)$.

The diversity is also "slowly varying," in the sense that $r(v\pi) \leqslant r(\pi) + 1$ for any v in V. For any natural number K in the range $0 \leqslant K \leqslant r(\pi)$, this implies that if ι is either the greatest right divisor of π such that $r(\iota) \leqslant K$ or the least right divisor of π such that $r(\iota) \geqslant K$, then $r(\iota) = K$.

For every $K \geqslant 1$ and every string π in V^*, let $p_K(\pi)$ (the *K-packing number* of π) denote the maximum number of contiguous substrings, each of diversity at least K, into which π can be parsed: $p_K(\pi) = \max\{J: \pi = \pi_1 \cdots \pi_J, r(\pi_1) \geqslant K, ..., r(\pi_J) \geqslant K\}$. The packing numbers are positive, $p_K(\pi) \geqslant 0$, and superadditive $p_K(\alpha\beta) \geqslant p_K(\alpha) + p_K(\beta)$; it follows from this that they are increasing, $p_K(\alpha\beta) \geqslant p_K(\alpha)$ and $p_K(\alpha\beta) \geqslant p_K(\beta)$. They are also "almost subadditive," in the sense that $p_K(\alpha\beta) \leqslant p_K(\alpha) + p_K(\beta) + 1$. They satisfy the inequalities $\lfloor r(\pi)/K \rfloor \leqslant p_K(\pi) \leqslant \lfloor \|\pi\|/K \rfloor$. Finally, if $p_K(\pi) \geqslant 1$ and if ι is the least right divisor of π such that $r(\iota) \geqslant K$, then $p_K(\pi) = p_K(\pi/\iota) + 1$, which allows $p_K(\pi)$ to be computed by a "greedy algorithm."

For every $K \geqslant 1$ and every string π in V^*, let $q_K(\pi)$ (the *K-covering number* of π)

denote the minimum number of contiguous substrings, each of diversity at most K, into which π can be parsed: $q_K(\pi) = \min\{J: \pi = \pi_1 \cdots \pi_1, r(\pi_1) \leqslant K,..., r(\pi_J) \leqslant K\}$. The covering numbers are strictly positive, $q_K(\pi) \geqslant 0$ and $q_K(\pi) = 0$ only if $\pi = \varepsilon$, and subadditive $q_K(\alpha\beta) \leqslant q_K(\alpha) + q_K(\beta)$. They are also "almost superadditive," in the sense that $q_K(\alpha\beta) \geqslant q_K(\alpha) + q_K(\beta) - 1$; it follows from this that they are increasing, $q_K(\alpha\beta) \geqslant q_K(\alpha)$ and $q_K(\alpha\beta) \geqslant q_K(\beta)$. They satisfy the inequalities $\lceil r(\pi)/K \rceil \leqslant q_K(\pi) \leqslant \lceil \|\pi\|/K \rceil$. Finally, if $q_K(\pi) \geqslant 1$ and if ι is the greatest right divisor of π such that $r(\iota) \leqslant K$, then $q_K(\pi) = q_K(\pi/\iota) + 1$, which allows $q_K(\pi)$ to be computed by a "greedy algorithm."

A calculation $f = [(\Pi_0, \pi_0),..., (\Pi_T, \pi_T)]$ will be called a *diminishing* calculation if π_T is a common right divisor of $\pi_0,..., \pi_T$. (Informally, a diminishing calculation is one that never pushes onto the stack any of the items in the final stack configuration.) A calculation $f = [(\Pi_0, \pi_0),..., (\Pi_T, \pi_T)]$ will be called an *augmenting* calculation if π_0 is a common right divisor of $\pi_0,..., \pi_T$. (Informally, an augmenting calculation is one that never pops off of the stack any of the items in its initial stack configuration.)

If $f: (A, \alpha) \to (B, \beta)$ is a calculation $[(\Pi_0, \pi_0),..., (\Pi_T, \pi_T)]$, let ι be the greatest common right divisor of $\pi_0,..., \pi_T$ (ι is the largest string that remains on the stack during f). Choose some t in the range $0 \leqslant t \leqslant T$ such that $\iota = \pi_t$ and let $I = \Pi_t$. Then f can be written as $g \circ h$, where $g: (A, \alpha) \to (I, \iota)$ is a diminishing calculation and $h: (I, \iota) \to (B, \beta)$ is an augmenting calculation. This may be denoted $f: (A, \alpha) \to (I, \iota) \to (B, \beta)$.

If $f: (A, \alpha) \to (I, \iota) \to (B, \beta)$, then

$$r(\nu(f)) \leqslant |A| + r(\lambda(f)) + r(\alpha/\iota).$$

(Any vertex that receives a pebble during f must either have one in the initial pebble configuration A, receive one by virtue of its immediate predecessors having pebbles, and thus appear in $\lambda(f)$, or be popped off of the initial stack configuration, and thus appear in α/ι.) For a diminishing calculation we may take $(I, \iota) = (B, \beta)$ and write

$$r(\nu(f)) \leqslant |A| + r(\lambda(f)) + r(\alpha/\beta).$$

If $f: (A, \alpha) \to (I, \iota) \to (B, \beta)$, then

$$r(\beta/\iota) \leqslant |I| + r(\lambda(f)).$$

(Any vertex that appears in β/ι must be pushed into the stack after (I, ι) and must either appear in the intermediate pebble configuration I or receive a pebble by virtue of its immediate predecessors having pebbles, and thus appear in $\lambda(f)$. For an augmenting calculation we may take $(I, \iota) = (A, \alpha)$ and write

$$r(\beta/\alpha) \leqslant |A| + r(\lambda(f)).$$

LEMMA 3.1. *For every $K \geqslant 1$ and every augmenting calculation f: $(A, \alpha) \to (B, \beta)$ in space at most K,*

$$p_K(\lambda(f)) + 1 \geqslant q_{2K}(\beta) - q_{2K}(\alpha).$$

Proof. Since f is augmenting, α is a right divisor of β and $q_{2K}(\beta) \leqslant q_{2K}(\beta/\alpha) + q_{2K}(\alpha)$. Thus it will suffice to show that

$$p_K(\lambda(f)) + 1 \geqslant q_{2K}(\beta/\alpha). \tag{3.1}$$

We shall prove (3.1) by induction on $q_{2K}(\beta/\alpha)$. If $q_{2K}(\beta/\alpha) \leqslant 1$, then (3.1) is trivial. If on the other hand $q_{2K}(\beta/\alpha) \geqslant 2$, then $r(\beta/\alpha) \geqslant 2K + 1$. Let ι be the greatest right divisor of β such that $r(\iota/\alpha) \leqslant 2K$. Then $r(\iota/\alpha) = 2K$. Let $f = [(\Pi_0, \pi_0),..., (\Pi_T, \pi_T)]$ and choose the minimum t in the range $0 \leqslant t \leqslant T$ such that ι is a common right divisor of $\pi_t,..., \pi_T$. Then $i = \pi_t$. Let $I = \Pi_t$. Then f can be written as $g \circ h$, where g: $(A, \alpha) \to (I, \alpha)$ and h: $(I, \iota) \to (B, \beta)$ are augmenting calculations.

Since $r(\iota/\alpha) = 2K$, we have $|A| + r(\lambda(g)) \geqslant 2K$. Since $|A| \leqslant K$, we have $r(\lambda(g)) \geqslant K$, and so

$$p_K(\lambda(g)) \geqslant q_{2K}(\iota/\alpha). \tag{3.2}$$

Since ι/α is the greatest right divisor of β/α such that $r(\iota/\alpha) \leqslant 2K$, $q_{2K}(\beta/i) = q_{2K}(\beta/\alpha) - 1$, and thus

$$p_K(\lambda(h)) + 1 \geqslant q_{2K}(\beta/\iota) \tag{3.3}$$

by inductive hypothesis.

Since $p_K(\lambda(f)) \geqslant p_K(\lambda(g)) + p_K(\lambda(h))$ and $q_{2K}(\beta/\alpha) \leqslant q_{2K}(\beta/\iota) + q_{2K}(\iota/\alpha)$, (3.1) follows by summing (3.2) and (3.3). ∎

LEMMA 3.2. *For every $K \geqslant 1$ and every calculation f: $(A, \alpha) \to (B, \beta)$ in space at most K,*

$$2p_K(\lambda(f)) \geqslant p_{6K}(\nu(f)) + q_{2K}(\beta) - q_{2K}(\alpha), \tag{3.4}$$

provided that $p_{6K}(\nu(f)) \geqslant 1$.

Proof. The string $\nu(f)$ can be parsed into $J = p_{6K}(\nu(f)) \geqslant 1$ substrings $\nu_1,..., \nu_J$ such that $\nu(f) = \nu_1 \cdots \nu_J$ and $r(\nu_j) \geqslant 6K$ for $1 \leqslant j \leqslant J$. The calculation f can thus be parsed into J subcalculations f_1: $(A_1, \alpha_1) \to (B_1, \beta_1),..., f_J$: $(A_J, \alpha_J) \to (B_J, \beta_J)$ such that $f = f_1 \circ \cdots \circ f_J$ (so that $(A, \alpha) = (A_1, \alpha_1)$, $(B_j, \beta_j) = (A_{j+1}, \alpha_{j+1})$ for $1 \leqslant j \leqslant J - 1$ and $(B_J, \beta_J) = (B, \beta)$) and $r(\nu(f_j)) \geqslant 6K$ for $1 \leqslant j \leqslant J$. It will suffice to show that

$$2p_K(\lambda(f_j)) \geqslant 1 + q_{2K}(\beta_j) - q_{2K}(\alpha_j) \tag{3.5}$$

for $1 \leqslant j \leqslant J$; (3.4) will then follow by summing (3.5) over $1 \leqslant j \leqslant J$.

Write f_j as $g_j \circ h_j$, where g_j: $(A_j, \alpha_j) \to (I_j, \iota_j)$ is a diminishing calculation and h_j: $(I_j, \iota_j) \to (B_j, \beta_j)$ is an augmenting calculation. By Lemma 3.1, $p_K(\lambda(h_i)) + 1 \geqslant q_{2K}(\beta_j) - q_{2K}(\iota_j)$, and since $p_K(\lambda(f_j)) \geqslant p_K(\lambda(h_j))$, we have

$$p_K(\lambda(f_j)) + 1 \geqslant q_{2K}(\beta_j) - q_{2K}(\iota_j). \tag{3.6}$$

Since $r(\nu(f_j)) \geqslant 6K$, $|A_j| + r(\lambda(f_j)) + r(\alpha_j/\iota_j) \geqslant 6K$, and since $|A_j| \leqslant K$, we have $r(\lambda(f_j)) + r(\alpha_j/\iota_j) \geqslant 5K$. This implies $p_K(\lambda(f_j)) + q_{2K}(\alpha_j/\iota_j) \geqslant 3$. (If $r(\alpha_j/\iota_j) \geqslant 4K + 1$, then $q_{2K}(\alpha_j/\iota_j) \geqslant 3$; if $2K + 1 \leqslant r(\alpha_j/\iota_j) \leqslant 4K$, then $q_{2K}(\alpha_j/\iota_j) \geqslant 2$, $r(\lambda(f_j)) \geqslant K$ and $p_K(\lambda(f_j)) \geqslant 1$; finally if $r(\alpha_j/\iota_j) \leqslant 2K$, then $r(\lambda(f_j)) \geqslant 3K$ and $p_K(\lambda(f_j)) \geqslant 3$.) Thus

$$p_K(\lambda(f_j)) \geqslant 3 - q_{2K}(\alpha_j/\iota_j). \tag{3.7}$$

Since $q_{2K}(\alpha_j) \geqslant q_{2K}(\alpha_j/\iota_j) + q_{2K}(\iota_j) - 1$, (3.5) follows by summing (3.6) and (3.7). ∎

PROPOSITION 3.1. *For every $K \geqslant 1$ and every original calculation f in space at most K,*

$$2p_K(\lambda(f)) \geqslant p_{6K}(\nu(f)).$$

provided that $p_{6K}(\nu(f)) \geqslant 1$.

Proof. The proposition follows immediately from Lemma 3.2, since for an original calculation f: $(0, \varepsilon) \to (B, \beta)$, we have $q_{2K}(\varepsilon) = 0$ and $q_{2K}(\beta) \geqslant 0$. ∎

6. THE LOWER BOUND

In Proposition 4.1 we shall show that the upper bounds obtained in Section 2 are, to within constant factors, the best possible. The proof of this proposition is very similar to the argument used by Paul and Tarjan [9] for the ordinary pebble game, with two main differences. First, a recent result of Gabber and Galil is used instead of a probabilistic or counting argument so as to obtain "uniform" graphs. This means that an encoding of a graph with N vertices can be generated by a machine in space $O(\log N)$. (If we were willing to sacrifice this uniformity, a probabilistic argument similar to the one in Pippenger [11] could be used to reduce the constant $D = 7^{50}$ that occurs below to $D = 276$.) A second and more important difference is that Proposition 3.1 is used to take account of the auxiliary pushdown.

For every $M \geqslant 1$, let $W_M = \{0,..., 10M - 1)\} \times \{0,..., 10M - 1\}$ and for every $w = (x, y)$ in W_M, let $Q(w) = \{(x, y),\ (x, 2x + y),\ (x, 2x + y + 1),\ (x, 2x + y + 2),\ (x + 2y, y),\ (x + 2y + 1, y),\ (x + 2y + 2, y)\}$, (all additions are modulo $10M$). For $0 \leqslant \xi \leqslant 1$, let $\varphi(\xi) = \xi(1 + (2 - \sqrt{3})(1 - \xi)/2)$.

LEMMA 4.1. *For every $M \geqslant 1$ and every subset Π of W_M,*

$$|Q(\Pi)|/100M^2 \geqslant \varphi(|\Pi|/100M^2).$$

Proof. See Gabber and Galil [4], Theorem 2′. ∎

For every $I \geqslant 0$, $M \geqslant 1$ and w in W_M, define $Q^{(I)}(w)$ to be $\{w\}$ if $I = 0$ and $Q(Q^{(I-1)}(w))$ if $I \geqslant 1$. Note that since $|Q(w)| \leqslant 7$, $|Q^{(I)}(w)| \leqslant 7^I$. For $0 \leqslant \xi \leqslant 1$, define $\varphi^{(I)}(\xi)$ to be ξ if $I = 0$ and $\varphi(\varphi^{(I-1)}(\xi))$ if $I \geqslant 1$. Note that since φ is increasing, so is $\varphi^{(I)}$.

LEMMA 4.2. *For every $I \geqslant 0$, $M \geqslant 1$ and Π a subset of W_M,*

$$|Q^{(I)}(\Pi)|/100M^2 \geqslant \varphi^{(I)}(|\Pi|/100M^2).$$

Proof. The lemma follows immediately from Lemma 4.1 by induction on I. ∎

LEMMA 4.3. *For every $M \geqslant 1$ and Π a subset of W_M, if $|\Pi| \geqslant 2M^2$, then $|Q^{(50)}(\Pi)| \geqslant 50M^2$.*

Proof. Since $\varphi(\xi)/\xi$ is decreasing in ξ, $\xi \leqslant 1/2$ implies $\varphi(\xi)/\xi \geqslant \varphi(1/2)/(1/2) = (6 - \sqrt{3})/4$. Since $((6 - \sqrt{3})/4)^{50} \geqslant 25$, $\varphi^{(50)}(1/50) \geqslant 1/2$. The lemma now follows from Lemma 4.2. ∎

Let $\mathscr{G}_D(N)$ denote the class of acyclic directed graphs with N vertices in which every vertex has at most D immediate predecessors (the number of immediate successors is not restricted). In the derivation of the lower bound it will be convenient to work with graphs in $\mathscr{G}_D(N)$ for $D = 7^{50}$. These will later be transformed into graphs in $\mathscr{G}_2(N)$.

Let $\mathscr{F}_D(L, M)$ denote the class of graphs in $\mathscr{G}_D(LM)$ that have depth $L - 1$, a set V_l of precisely M vertices assigned level $l - 1$ for each $1 \leqslant l \leqslant L$ and $P(V_{l+1})$ contained in V_l for each $1 \leqslant l \leqslant L - 1$.

For every $L \geqslant 2$ and $M \geqslant 1$, consider the graph $F_{L,M}$ obtained by taking, for every $1 \leqslant l \leqslant L$, $V_l = \{l\} \times W_M$ and by taking, for every $1 \leqslant l \leqslant L - 1$ and every $v = (l + 1, w)$ in V_{l+1}, $P(v) = \{l\} \times Q^{(50)}(w)$. Then $F_{L,M}$ belongs to $\mathscr{F}_D(L, 100M^2)$ for $D = 7^{50}$.

Let G be a graph with vertices V, let V_0 be a subset of V and let G_0 be the subgraph of G induced by V_0. There is a functor from $\mathscr{C}(G)$ to $\mathscr{C}(G_0)$ that acts on configurations by omitting all appearances of vertices outside V_0 from pebble configurations and pushdown configurations and acts on calculations by omitting all transitions of the form c_v, d_v, i_v or j_v for which v lies outside V_0. The image of a calculation f under this functor will be denoted $f | V_0$.

LEMMA 4.4. *For every $L \geqslant 2$ and $M \geqslant 1$ and every calculation for $F_{L,M}$ in space at most $K = 2M^2$,*

$$p_{6K}(\nu(f | V_l)) \geqslant 4 p_K(\lambda(f | V_{l+1})) \tag{4.1}$$

for $1 \leqslant l \leqslant L - 1$, provided $p_K(\lambda(f | V_{l+1})) \geqslant 1$.

Proof. The string $\lambda(f | V_{l+1})$ can be parsed into $J = p_K(\lambda(f | V_{l+1})) \geqslant 1$ substrings

$\lambda_1,\ldots,\lambda_J$ such that $\lambda(f|V_{l+1})=\lambda_1\cdots\lambda_J$ and $r(\lambda_j)\geqslant K$ for $1\leqslant j\leqslant J$. The calculation f can thus be parsed into J subcalculations $f_1,\ldots,f_J$ such that $f=f_1\circ\cdots\circ f_J$ and $r(\lambda(f_j|V_{l+1}))\geqslant K$ for $1\leqslant j\leqslant J$. We shall show that

$$p_{6K}(\lambda(f_j|V_l))\geqslant 4 \tag{4.2}$$

for $1\leqslant j\leqslant J$; since $p_{6K}(\lambda(f|V_l))\geqslant p_{6K}(\lambda(f_1|V_l))+\cdots+p_{6K}(\lambda(f_J|V_l))$, (4.1) will then follow by summing (4.2) over $1\leqslant j\leqslant 1$.

Let the subset Π of V_{l+1} comprise the vertices appearing in $\lambda(f_j|V_{l+1})$, so that $|\Pi|=r(f_j|V_{l+1})\geqslant K=2M^2$. Then $f_j|V_l$ runs through $P(\Pi)$, and by Lemma 4.3, $|P(\Pi)|\geqslant 50M^2=25K$. Thus $(f_j|V_l)$: $(A_j,\alpha_j)\to(B_j,\beta_j)$, so that $|A_j|+r(\nu(f_j|V_l))\geqslant 25K$. Since $|A_j|\leqslant K$, we have $r(\nu(f_j|V_l)\geqslant 24K$, which implies (4.2). ∎

LEMMA 4.5. *For every $L\geqslant 2$ and $M\geqslant 1$, every complete calculation f for $F_{L,M}$ in space at most $K=2M^2$ requires time at least $12K\cdot 2^L$.*

Proof. For $1\leqslant l\leqslant L$, we shall prove

$$I_l\colon\ p_{6K}(\nu(f|V_l))\geqslant 2^{L+2-l}$$

and

$$II_l\colon\ p_K(\lambda(f|V_l))\geqslant 2^{L+1-l}$$

by induction on l. Since a complete calculation f is original and runs through V_L, we have $r(\nu(f|V_L))=|V_L|=50K$ and thus $p_{6K}(\nu(f|V_L))\geqslant 4$, which implies I_L. By Proposition 3.1, I_{l+1} implies II_{l+1}, and by Lemma 4.4, II_{l+1} implies I_l. Thus we have I_1, which asserts $p_{6K}(\nu(f|V_1))\geqslant 2\cdot 2^L$ and thus $\|\nu(f|V_1)\|\geqslant 12K\cdot 2^L$, which implies the lemma. ∎

It remains to show how this lower bound for the graph $F_{L,M}$, in which a vertex may have at most 7^{50} immediate predecessors, can be extended to graphs in which a vertex may have at most two immediate predecessors.

LEMMA 4.6. *For every $D\geqslant 2$, there is a transformation $\mathscr{A}_D$ from $\mathscr{G}_D(N)$ to $\mathscr{G}_2((D-1)N)$ that maps graphs in $\mathscr{F}_D(L,M)$ into graphs with depth at most $(D-1)L$ and width at most $2M$, and for every graph G in $\mathscr{G}_D(N)$ there is a functor $\mathscr{B}_{D,G}$ from $\mathscr{C}(\mathscr{A}_D(G))$ to $\mathscr{C}(G)$ that maps complete calculations in time T and space S into complete calculations in time at most $(D-1)T$ and space at most $(D-1)S$.*

Proof. The transformation $\mathscr{A}_D$ acts as follows. If G has vertices V, then $\mathscr{A}_D(G)$ has vertices $\mathscr{A}_D(V)=\{1,\ldots,D-1\}\times V$ and (k,v) is an immediate predecessor of $(k+1,v)$ in $\mathscr{A}_D(G)$ for every $1\leqslant k\leqslant D-2$ and every v in V. If $w_1,\ldots,w_K$ are the immediate predecessors of v in the standard order in G for some $0\leqslant K\leqslant D$, then $(D-1,v)$ is an immediate predecessor of $(\max\{1,k-1\},v)$ in $\mathscr{A}_D(G)$ for every $1\leqslant k\leqslant K$. It is easy to verify that this maps graphs in $\mathscr{G}_D(N)$ into graphs in

$\mathscr{G}_2((D-1)N)$ and graphs in $\mathscr{F}_D(L, M)$ into graphs of depth at most $(D-1)L$ and width at most $2M$.

Let v be a vertex of G and let $w_1, ..., w_K$ be its immediate predecessors in the standard order in G for some $0 \leqslant K \leqslant D$. For every (k, v) in $\mathscr{A}_D(V)$, let $\Psi(k, v)$ in $V^{\#}$ comprise v alone if $k = D-1$ and the first $\min\{k+1, K\}$ elements of $w_1, ..., w_K$ if $1 \leqslant k \leqslant D-2$. Let $\psi(k, v)$ in V^* comprise the elements of $\Psi(k, v)$ in the standard order.

The functor $\mathscr{B}_{D,G}$ acts as follows. It acts on pebble configurations by sending each generator (k, v) of $\mathscr{A}_D(V)^{\#}$ into $\Psi(k, v)$ in $V^{\#}$ and it acts on stack configurations by sending each generator (k, v) of $\mathscr{A}_D(V)^*$ into $\psi(k, v)$ in V^*. This determines the action on configurations. It acts on calculations by sending each generator $c_{(k,v)}$, $d_{(k,v)}$, $i_{(k,v)}$ or $j_{(k,v)}$ of $\mathscr{C}(\mathscr{A}_D(G))$ into $c_{\Psi(k,v)}$, $d_{\Psi(k,v)}$, $i_{\Psi(k,v)}$ or $j_{\Psi(k,v)}$, respectively, in $\mathscr{C}(G)$. Since V is contained in $\Psi(\mathscr{A}_D(V))$, $\mathscr{B}_{D,G}$ maps complete calculations into complete calculations, and since $|\Psi(k, v)| \leqslant D-1$ and $\|\psi(k, v)\| \leqslant D-1$, $|\mathscr{B}_{D,G}(f)| \leqslant (D-1)|f|$ and $\|\mathscr{B}_{D,G}(f)\| \leqslant (D-1)\|f\|$. ∎

PROPOSITION 4.1. *For every $3 \leqslant S \leqslant N$, there is a graph G in $\mathscr{G}_2(N)$ of depth $O(N/S)$ and width $O(S)$ such that every complete calculation for G in space at most S requires time at least $S \exp \Omega(N/S)$.*

Proof. Immediate from Lemmas 4.5 and 4.6 with $D = 7^{50}$, $M = \lceil \sqrt{(D-1)}\, S/2 \rceil$ and $L = \lfloor N/100(D-1)M^2 \rfloor$. ∎

ACKNOWLEDGMENTS

The author is indebted to Stephen Cook for suggesting the application of this work to comparisons between **SC** and **NC**, and to Terry Wu for suggesting an improvement in Section 3.

REFERENCES

1. S. A. COOK, Characterizations of pushdown machines in terms of time-bounded computers, *Assoc. Comput. Mach.* **18** (1971), 4–18; preliminary version: *in* "Proceedings, 1st Annual ACM Symposium on Theory of Computing, 1969," pp. 229–231.
2. S. A. COOK, An observation on time-space trade-off, *J. Comput. System Sci.*, **9** (1974), 308–316; preliminary version: *in* "Proceedings, 5th Annual ACM Symposium on Theory of Computing, 1973," pp. 29–33.
3. S. A. COOK, Deterministic CFL's are accepted simultaneously in polynomial time and log squared space, *in* "Proceedings, 11th Annual ACM Symposium on Theory of Computing, 1979," pp. 338–345.
4. O. GABBER AND Z. GALIL, Explicit construction of linear size superconcentrators, *J. Comput. System Sci.*, in press; preliminary version: *in* "Proceedings, IEEE 20th Annual Symposium on Foundations of Computer Science, 1979," pp. 364–370.
5. J. E. HOPCROFT, W. J. PAUL, AND L. G. VALIANT, On time versus space, *J. Assoc. Comput. Mach.* (1977), 332–377; preliminary version: *in* "Proceedings, IEEE 16th Annual Symposium on Foundations of Computer Science, 1975," pp. 57–64.

6. R. E. LADNER, The circuit value problem is log space complete for *P*, *ACM SIGACT News* **7** (1975), 18–20.
7. T. LENGAUER AND R. E. TARJAN, Asymptotically tight bounds on time-space trade-offs in a pebble game *Assoc. Comput. Mach.* in press; preliminary version: *in* "Proceedings, 11th Annual ACM Symposium on Theory of Computing, 1979," pp. 262–277.
8. M. S. PATERSON AND C. E. HEWITT, Comparative schematology, *in* "Proj. MAC Conf. on Concurrent Systems and Parallel Computation," 2–5 June 1970, Woods Hole, Mass., pp. 119–127.
9. W. J. PAUL AND R. E. TARJAN, Time-space trade-offs in a pebble game, *Acta Inform.* **10** (1978), 111–115; preliminary version: *in* "Proceedings, 4th Annual Colloq. on Automata, Languages and Programming, 1977," pp. 365–369.
10. W. J. PAUL, R. E. TARJAN, AND J. R. CELONI, Space bounds for a game on graphs, *Math. Systems Theory* **10** (1977), 239–251; preliminary version: *in* "Proceedings, 8th Annual ACM Symposium on Theory of Computing, 1975," pp. 149–160.
11. N. PIPPENGER, Superconcentrators, *SIAM J. Comput.* **6** (1977), 298–304.
12. N. PIPPENGER, Fast simulation of combinational logic networks by machines without random-access storage, *in* "Allerton Conf. on Communication, Control and Computing," 28–30 Sept. 1977, Monticello, Ill., pp. 25–33.
13. N. PIPPENGER, On simultaneous resources bounds, *in* "Proceedings, IEEE 20th Symposium on Foundations of Computer Science, 1979," pp. 307–311.
14. N. PIPPENGER, Comparative schematology and pebbling with auxiliary pushdowns, *in* "Proceedings, 12th Annual ACM Symposium on Theory of Computing, 1980," pp. 351–356.
15. W. L. RUZZO, Tree-size bounded alternation, *J. Comput. System Sci.* **21** (1980), 218–235; preliminary version: *in* "Proceedings, 11th Annual ACM Symposium on Theory of Computing, 1979," pp. 352–359.
16. W. L. RUZZO, On uniform circuit complexity, *in* "Proceedings, 20th IEEE Symposium on Foundations of Computer Science, 1979," pp. 312–318.