The Pebbling Problem is Complete in Polynomial Space [1]

John R. Gilbert [2]
Thomas Lengauer [3]
Robert Endre Tarjan [4]

Computer Science Department
Stanford University
Stanford, California 94305

Abstract. We examine a pebbling problem which has been used to study the storage requirements of various models of computation. Sethi has shown this problem to be NP-hard and Lingas has shown a generalization to be P-space complete. We prove the original problem P-space complete by employing a modification of Lingas's proof. The pebbling problem is one of the few examples of a P-space complete problem not exhibiting any obvious quantifier alternation.

Keywords: computational complexity, P-space completeness, pebbling, register allocation.

1. Introduction.

In this paper, we consider the following pebble game. Let G be a directed acyclic graph, all of whose vertices have at most two predecessors.[5] Given a collection of pebbles, we wish to place a pebble on a distinguished vertex of G, called the goal, starting with no pebbles on the graph, by applying the following rules:

(i) A pebble may be removed from a vertex at any time.

(ii) If all predecessors of an unpebbled vertex v are pebbled, a pebble may be placed on v.

(iii) If all predecessors of an unpebbled vertex v are pebbled, a pebble may be moved from a predecessor of v to v.

We shall consider time to be divided into integral steps. At each time step, one of rules (i)-(iii) is applied once. The space required by the pebbling is the maximum number of pebbles ever on the graph at one time; the time required is the number of applications of rules (i)-(iii).

This pebble game has been used to model register allocation [14], to study flowcharts and recursive schemata [9], and to analyze the relative power of time and space as Turing machine resources [1,6]. Our interest lies in determining the computational complexity of the following problem, which we call the pebbling problem: given a graph G, can a given vertex v in G be pebbled using no more than s pebbles? This problem is not

---

1/ Research partially supported by National Science Foundation Grant MCS-75-22870 A02 and Office of Naval Research Contract N00014-76-C-0688. Reproduction in whole or in part is permitted for any purpose of the United States government.

2/ Research partially supported by a Hertz Fellowship.

3/ Research partially supported by the German Academic Exchange Service.

4/ Research partially supported by a Guggenheim Fellowship.

5/ We shall use the following graph-theoretic terminology. A directed graph $G = (V,E)$ is a collection of vertices V and a collection of edges E. Each edge is an ordered pair (v,w) of distinct vertices. If (v,w) is an edge, v is a predecessor of w and w is a successor of v. A source is a vertex with no predecessors; a sink is a vertex with no successors. A path from v to w is a sequence of vertices $v = v_1, v_2, \ldots, v_k = w$ such that v_{i+1} is a successor of v_i for $1 \leq i < k$. A cycle is a path of at least two vertices from v to v. A graph is acyclic if it has no cycles.

necessarily in NP,*/ since the number of moves necessary to pebble G with s pebbles may not be polynomially bounded [10]. However, the problem is in polynomial space, since a sequence of moves can be guessed and checked by a non-deterministic machine; only polynomial space is necessary to remember a single arrangement of pebbles on the graph (or configuration). By Savitch's theorem [12], such a non-deterministic machine can be converted into a deterministic machine for which the space bound is at most squared.

Most of the known P-space complete problems, such as the quantified Boolean formula problem [15] and various game problems [3,4,7,13] possess an obvious quantifier alternation not present in the pebbling problem. Thus we might expect difficulties in showing the pebbling problem P-space complete. Sethi [14] was able to show the problem NP-hard, and NP-complete in the special case that each vertex can be pebbled only once. Lingas [8] generalized the problem by allowing "or" vertices (an "or" vertex can be pebbled if at least one of its predecessors is pebbled) and proved the generalized version P-space complete. We shall prove the original pebbling problem P-space complete by modifying Lingas's construction. The next section of the paper contains the proof. The concluding section mentions some additional consequences of our construction.

2. The Construction.

Quantified Boolean Formulas.

In order to prove the pebbling problem P-space complete, we must reduce a known P-space complete problem to the pebbling problem. For this purpose we choose the quantified Boolean formula problem (QBF) [15]: Determine whether a quantified formula of the form $Q_1x_1\, Q_2x_2 \ldots Q_nx_nF$ is true, where each x_i is a Boolean variable, each Q_i is either an existential or a universal quantifier, and F is an unquantified Boolean formula involving only the variables x_i, in conjunctive normal form with exactly three literals per clause. From the quantified formula we construct a graph G with a goal vertex q_1 and a number of pebbles s such that the quantified formula is true if and only if q_1 can be pebbled with s pebbles. It will be evident that the transformation from formula to graph can be accomplished in logarithmic space; it follows that the pebbling problem is log-space complete in P-space.

We need a notation to denote substitution of truth values in F. For technical reasons we substitute for the literals rather than for the variables: $F(e_1,e_2,\ldots,e_{2k-1},e_{2k})$ denotes the formula obtained from F by replacing each occurrence of x_i by e_{2i-1} and each occurrence of $\bar{x}_i$ by e_{2i}, for $1 \le i \le k$. Here each e_j is either true or false. Thus F(true,false) denotes making x_1 true (and $\bar{x}_1$ false), F(false,true) denotes making x_1 false (and $\bar{x}_1$ true), and F(false,false) denotes the "double false" substitution making x_1 false and $\bar{x}_1$ also false. (We shall have no need to consider the "double true" substitution F(true,true).) Note that if $F(e_1,\ldots,e_{2k-2},\text{false},\text{false},e_{2k+1},\ldots,e_{2n})$ is true, then both $F(e_1,\ldots,e_{2k-2},\text{true},\text{false},e_{2k+1},\ldots,e_{2n})$ and $F(e_1,\ldots,e_{2k-2},\text{false},\text{true},e_{2k+1},\ldots,e_{2n})$ are true. Thus if $Q_{k+1}x_{k+1}\ldots Q_nx_nF(e_1,\ldots,e_{2k-2},\text{false},\text{false})$ is true, so is $\forall x_kQ_{k+1}x_{k+1}\ldots Q_nx_nF(e_1,\ldots,e_{2k-2})$.

Some Useful Technology.

An important building block in our construction is the "pyramid" graph exhibited in Fig. 1, which we shall abbreviate with a triangle as indicated in the figure. Cook [1] has shown that the sink (or apex) of a pyramid with k sources can be pebbled if and only if at least k pebbles are available. One use of a pyramid is to lock a pebble on a given vertex for a given time interval. This can be done by making the vertex the apex of a pyramid which is so large that in order to repebble the vertex, so many pebbles have to be taken off the graph for use on the pyramid that the results achieved after the vertex was first pebbled are lost.

*/ We use standard concepts from complexity theory without defining them. For a thorough discussion of NP, P-space, and completeness, see [5].

We also note that if any source of a k-pyramid contains a pebble which cannot be moved, then the apex can be pebbled with k-1 additional pebbles.

[Figure 1]

We now make some general observations about pebbling strategies which are similar to those of Pippenger [11]. We partition the pebble placements into <u>necessary</u> and <u>unnecessary</u> placements as follows. The first placement on the goal vertex is necessary; all other placements on the goal vertex are unnecessary. A placement on any other vertex v is necessary if and only if the pebble placed remains on v until a necessary placement occurs on a successor of v. The necessary placements are well-defined since the graph is acyclic. Deletion of all unnecessary placements from a pebbling strategy results in another pebbling strategy. We call a pebbling strategy with no unnecessary placements <u>frugal</u>. The following statements are true of any frugal pebbling strategy.

(i) At all times after the first placement on a vertex v, some path from v to the goal vertex contains a pebble.

(ii) At all times after the last placement on a vertex v, all paths from v to the goal vertex contain a pebble. (This is true also of non-frugal pebbling strategies.)

(iii) The number of placements on a non-goal vertex is bounded by the total number of placements on its successors.

We call a pebbling strategy <u>normal</u> if it is frugal and if it pebbles each pyramid P in G as follows: after the first pebble is placed on P, no placement or removal occurs outside P until the apex of P is pebbled and all other pebbles are removed from P. No new placement occurs on P until after the pebble on the apex of P is removed.

<u>Lemma 1</u>. If the goal vertex is not inside a pyramid, any pebbling strategy can be transformed into a normal pebbling strategy without increasing the number of pebbles used.

<u>Proof</u>. Consider any pebbling strategy. First obtain a frugal strategy by deleting all unnecessary placements; this does not increase the number of pebbles used. Then let t_1 be a time at which a pebble is placed on a k-pyramid P. Let $[t_0,t_2]$ be the largest time interval containing t_1 such that P is never pebble-free during $[t_0,t_2]$. Since the pebbling strategy is frugal and the goal vertex is not in P, the only pebble on P at time t_2 is on the apex of P. Since at time t_0-1 no pebbles are on P, there must be a time t_3 during $[t_0,t_2]$ at which k pebbles are on P. Modify the pebbling strategy as follows. Delete all placements and removals from P during $[t_0,t_2]$. Insert at t_3 a contiguous sequence of moves which pebbles the apex of P using k pebbles and then removes all pebbles on P except the one on the apex. This transformation results in a pebbling strategy since no vertex in P has a predecessor outside P, and the only vertex in P which precedes vertices outside P is the apex. If the inserted sequence contains no unnecessary placements, then the transformed strategy is frugal. Furthermore it uses no more pebbles than the original strategy. Repeating this transformation for each placement on a pyramid results in a normal strategy. □

<u>Details of the Construction</u>.

To describe the construction we need a little more notation. Recall that n is the number of quantifiers. The number of pebbles we allow is $s = 3n+3$. For $1 \leq i \leq n+1$, let $s_i = s-3i+3$; thus $s_1 = s$ and $s_{n+1} = 3$. Roughly speaking, we use three pebbles to keep track of each quantifier and its associated variable, and three more to check the validity of the clauses of F under a given assignment to the variables. Let F contain m clauses $(\ell_{j1} \vee \ell_{j2} \vee \ell_{j3})$ for $1 \leq j \leq m$, where each ℓ_{jk} is a literal. For any variable x, we shall regard $\bar{\bar{x}}$ as synonymous with x.

The graph G to be constructed consists of n+m blocks of vertices, one for each quantifier and its associated variable, and one for each clause in F. The quantifier block for $Q_i x_i$ includes four vertices to represent the variable x_i, as illustrated in Fig. 2. Two pebbles placed on this subgraph encode the truth values of x_i and $\bar{x}_i$ as illustrated in Fig. 2(b)-(d). The remainder of a quantifier block depends on the quantifier.

[Figure 2]

Figure 3 illustrates a universal quantifier block. The way this block works is as follows. There are essentially two ways to pebble q_i with s_i pebbles: (i) pebble q_{i+1} twice with s_{i+1} pebbles, each time with three pebbles fixed on the i-th quantifier block, once representing $\bar{x}_i$ true and once representing x_i true (the third pebble is fixed on d_i or a_i respectively); or (ii) pebble q_{i+1} once with s_{i+1} pebbles, while three pebbles representing x_i false and $\bar{x}_i$ false are fixed on the i-th quantifier block.

[Figure 3]

Figure 4 illustrates an existential quantifier block. The only way to pebble q_i with s_i pebbles is to pebble q_{i+1} with s_{i+1} pebbles, while three pebbles representing one of the three possible truth assignments to x_i and $\bar{x}_i$ are fixed on the i-th quantifier block (the third pebble is fixed on d_i).

[Figure 4]

Figure 5 illustrates the block of vertices representing a clause. After s-3 pebbles are used on the quantifier blocks to fix an assignment to the literals, the remaining three pebbles are available to pebble the clause blocks. For each literal ℓ_{jk}, there is a fixed pebble on vertex ℓ_{jk} if the literal is true, or on vertex ℓ'_{jk} if the literal is false. Thus if F is valid, the clause pyramids can be pebbled in the order $p_0, p_1, \ldots, p_m = q_{n+1}$ with three pebbles; however, if some clause $(\ell_{j1} \vee \ell_{j2} \vee \ell_{j3})$ is false, p_j is the apex of an empty 4-pyramid and cannot be pebbled with three pebbles.

[Figure 5]

Figure 6 illustrates the entire construction. Note that p_0 is a single vertex, and that $p_m = q_{n+1}$.

[Figure 6]

Proof of the Reduction.

Our main result is as follows.

Theorem 1. The quantified Boolean formula $Q_1x_1\,Q_2x_2 \ldots Q_nx_nF$ is true if and only if vertex q_1 in the graph G constructed as above can be pebbled with $s = 3n+3$ pebbles.

We prove this theorem by means of two lemmas which state that if we use $s-s_i$ pebbles to fix truth values for the literals corresponding to the first i-1 variables, then we can pebble q_i with the remaining s_i pebbles if and only if the quantified formula is valid after making the appropriate substitution. The lemmas are proved by induction on i . For $1 \leq i \leq n+1$, we define N_i to be the set of configurations fixing truth values for the literals corresponding to the first i variables. An arrangement of exactly $s-s_i$ pebbles on G is in N_i if and only if, for $1 \leq j < i$, two conditions hold:

(1) If $Q_j = \forall$, there are exactly three pebbles on the j-th quantifier block, on one of the following three sets of vertices:
  - (a) $\{a_j, x_j, \bar{x}'_j\}$, indicating x_j true,
  - (b) $\{d_j, \bar{x}_j, x'_j\}$, indicating x_j false, or
  - (c) $\{d_j, x'_j, \bar{x}'_j\}$, indicating double false.

(2) If $Q_j = \exists$, there are exactly three pebbles on the j-th quantifier block, on one of the following three sets of vertices:
  - (a) $\{d_j, x_j, \bar{x}'_j\}$, indicating x_j true,
  - (b) $\{d_j, \bar{x}_j, x'_j\}$, indicating x_j false, or
  - (c) $\{d_j, x'_j, \bar{x}'_j\}$, indicating double false.

Note that N_1 contains only the configuration with no pebbles on the graph, and N_{n+1} contains all configurations in which a truth assignment has been made to each literal and three pebbles remain to test whether the assignment makes F true.

Lemma 2. Let $1 \leq i \leq n+1$. Suppose the graph is initially in a configuration in N_i . For $1 \leq j < i$, let e_{2j-1} be the truth assignment defined for x_j by that configuration, and let e_{2j} be the truth assignment defined for $\bar{x}_j$. If $Q_ix_i \ldots Q_nx_nF(e_1, e_2, \ldots, e_{2i-3}, e_{2i-2})$ is true, then vertex q_i can be pebbled with s_i additional pebbles without moving any of the $s-s_i$ pebbles initially on the graph.

Proof. Proof is by induction on i from n+1 to 1 .

Basis.

Let i = n+1 and suppose that the assignment defined by the N_i configuration makes F true. We must show that vertex $q_{n+1} = p_m$ can be pebbled with $s_{n+1} = 3$ additional pebbles without moving any of the pebbles of the N_i configuration.

For each clause $(\ell_{j1} \vee \ell_{j2} \vee \ell_{j3})$ of F, there is a pebble of the configuration on ℓ_{j1} or ℓ_{j2} or ℓ_{j3}, and if there is not a pebble on ℓ_{jk} then there is a pebble on ℓ'_{jk}, for $1 \leq k \leq 3$. It follows that with three additional pebbles we can pebble $p_0, p_1, \ldots, p_m$ in turn as described earlier. Note that we need at least three additional pebbles, since each p_j for $j \geq 1$ is the apex of a three-source pyramid initially containing no pebbles.

Inductive step.

Suppose that the lemma holds for $i+1$, and that the assignment defined by the N_i configuration makes the substituted formula $Q_i x_i \ldots Q_n x_n F(e_1, e_2, \ldots, e_{2i-3}, e_{2i-2})$ true.

Case 1 (universal quantifier). Suppose $Q_i = \forall$. Then

$$Q_{i+1}x_{i+1} \ldots Q_n x_n F(e_1, \ldots, e_{2i-2}, \text{true}, \text{false})$$

and

$$Q_{i+1}x_{i+1} \ldots Q_n x_n F(e_1, \ldots, e_{2i-2}, \text{false}, \text{true})$$

are both true.

Vertex q_i can be pebbled with s_i pebbles as follows. First use all s_i pebbles to pebble x'_i, leaving a pebble there. Then use the remaining s_i-1 pebbles to pebble d_i, leaving a pebble there, and the remaining s_i-2 pebbles to pebble $\bar{x}_i$, leaving a pebble there. The current configuration is in N_{i+1}, representing x_i false. Applying the induction hypothesis, pebble q_{i+1} with the remaining $s_{i+1} = s_i-3$ pebbles. Move the pebble on q_{i+1} to c_i, b_i, and a_i. Move the pebble on x'_i to x_i. Leaving pebbles on a_i and x_i, pick up the rest of the pebbles and use the s_i-2 free pebbles to pebble $\bar{x}'_i$, leaving a pebble there. The current configuration is in N_{i+1}, representing x_i true. Applying the induction hypothesis, pebble q_{i+1} again. Finish by moving the pebble on q_{i+1} to g_i, f_i, and q_i.

If $Q_{i+1}x_{i+1} \ldots Q_n x_n F(e_1, \ldots, e_{2j-2}, \text{false}, \text{false})$ is true, there is a way to pebble q_i which only pebbles q_{i+1} once. First pebble x'_i, d_i, and $\bar{x}'_i$, which gives a configuration in N_{i+1} representing x_i and $\bar{x}_i$ both false. Applying the induction hypothesis, pebble q_{i+1}. There are now $s_i-4 \geq 2$ free pebbles. Place one on $\bar{x}_i$ and move it to c_i, b_i, and a_i. Move the pebble on $\bar{x}'_i$ to g_i, and finish by moving the pebble on x'_i to x_i, f_i, and q_i.

Case 2 (existential quantifier). Suppose $Q_i = \exists$. Then either

$$Q_{i+1}x_{i+1} \ldots Q_n x_n F(e_1, \ldots, e_{2i-2}, \text{true}, \text{false})$$

or

$$Q_{i+1}x_{i+1} \ldots Q_n x_n F(e_1, \ldots, e_{2i-2}, \text{false}, \text{true})$$

is true.

Suppose first that the former is the case. Vertex q_i can be pebbled with s_i pebbles as follows. First pebble x'_i, leaving a pebble there. Then pebble d_i and f_i, leaving pebbles there. Move the pebble on f_i to $\bar{x}'_i$, and move the pebble on x'_i to x_i. The current configuration is in N_{i+1}, representing x_i true. Applying the induction hypothesis, pebble q_{i+1} with the remaining $s_{i+1} = s_i-3$ pebbles. There are now $s_i-4 \geq 2$ free pebbles. Place one on $\bar{x}_i$ and finish by moving it to c_i, b_i, a_i, and q_i.

Alternatively, suppose that $Q_{i+1}x_{i+1} \ldots Q_n x_n F(e_1, \ldots, e_{2i-2}, \text{false}, \text{true})$ is true. To pebble q_i with s_i pebbles, begin by pebbling x'_i, d_i, and f_i in turn, leaving pebbles there. Move the pebble on f_i to $\bar{x}'_i$ and $\bar{x}_i$, which gives a configuration in N_{i+1} representing x_i false. Applying the induction hypothesis, pebble q_{i+1}. Move the pebble on q_{i+1} to c_i and b_i. Pick up all the pebbles except those on b_i and x'_i, and use the s_i-2 free pebbles to pebble f_i. Move the pebble on f_i to $\bar{x}'_i$ and a_i, and finish by moving the pebble on x'_i to x_i and q_i. □

Lemma 3. Let $1 \leq i \leq n+1$. Suppose the graph is initially in a configuration in N_i. For $1 \leq j < i$, let e_{2j-1} be the truth assignment defined for x_j by that configuration, and let e_{2j} be the truth assignment defined for $\bar{x}_j$. If vertex q_i can be pebbled with s_i additional pebbles without moving any of the $s-s_i$ pebbles initially on the graph, then $Q_i x_i \ldots Q_n x_n F(e_1, e_2, \ldots, e_{2i-3}, e_{2i-2})$ is true.

Proof. Again, proof is by induction on i from $n+1$ to 1.

Basis.

Let $i = n+1$ and suppose $q_i = p_m$ can be pebbled with $s_i = 3$ pebbles without moving any pebbles in the N_i configuration. Then each pyramid of size four representing a clause of F must contain at least one pebble of the N_i configuration, corresponding to a true literal; that is, the assignment defined by the N_i configuration must make F true.

Inductive step.

Suppose that the lemma holds for $i+1$, and that there is a strategy which pebbles q_i with s_i pebbles without moving any pebbles in the N_i configuration. By Lemma 1 we can assume that the strategy is normal.

Case 1 (universal quantifier). Suppose $Q_i = \forall$. By frugality, each of q_i, a_i, b_i, c_i, d_i, f_i, and g_i is only pebbled once.

Let t_0 be the last time s_i pebbles appear on the s_i-pyramid. After t_0, x_i' is only pebbled once. At t_0 no pebbles appear on vertices outside the s_i-pyramid. Since the pebbling is frugal, no placement before t_0 occurs outside the s_i-pyramid. Thus x_i' is only pebbled once, and this occurs before anything else happens. Let t_1 be the time x_i' is pebbled. From t_1 until q_i is pebbled, a pebble is on x_i', x_i, or f_i. From t_1 until a_i is pebbled, a pebble is on x_i'.

To pebble a_i requires pebbling d_i. This requires removing all pebbles from the graph except the one on x_i'. By normality, therefore, d_i is pebbled before anything other than x_i', and a pebble remains on d_i until b_i is pebbled. To pebble b_i requires pebbling c_i and hence $\bar{x}_i'$. This requires removing all pebbles except those on x_i' and d_i. Therefore $\bar{x}_i'$ is pebbled immediately after d_i and a pebble remains on $\bar{x}_i'$ or $\bar{x}_i$ until c_i is pebbled, which happens before b_i is pebbled. By normality, all pebbles except the one on $\bar{x}_i'$ are removed from the s_i-2 -pyramid as soon as $\bar{x}_i'$ is pebbled. Let t_2 be the time these pebbles are removed, and let t_3 be the first time after t_2 that q_{i+1} is pebbled.

At t_2 there are pebbles on x_i', d_i, and $\bar{x}_i'$. Pebbles must remain on x_i' and d_i until t_3, and a pebble must be on either $\bar{x}_i'$ or $\bar{x}_i$ until t_3. Suppose first that a pebble remains on $\bar{x}_i'$ from t_2 until t_3. The configuration at t_2 is in N_{i+1} with a double false assignment to x_i, and none of the pebbles on the graph at t_2 can be removed until t_3. Therefore the induction hypothesis says that

$Q_{i+1}x_{i+1} \cdots Q_n x_n F(e_1,\ldots,e_{2i-2},\text{false},\text{false})$ is true, so $\forall x_i Q_{i+1}x_{i+1} \cdots Q_n x_n F(e_1,\ldots,e_{2i-2})$ is true and the lemma holds in this case.

Alternatively, suppose that the pebble on $\bar{x}_i'$ does not remain until t_3. In this case we will argue that q_{i+1} must be pebbled twice, first with a false assignment to x_i and then with a true assignment to x_i.

Either $\bar{x}_i'$ or $\bar{x}_i$ must have a pebble from t_2 to t_3. The only successors of $\bar{x}_i'$ are $\bar{x}_i$ and g_i, and g_i cannot be pebbled before t_3. Therefore we can rearrange the strategy so that at t_2+1 the pebble on $\bar{x}_i'$ is moved to $\bar{x}_i$, where it must remain until t_3. The configuration at t_2+1 is then in N_{i+1} with a false assignment to x_i, and none of the pebbles on the graph at t_2+1 can be removed until t_3. By the induction hypothesis, $Q_{i+1}x_{i+1} \cdots Q_n x_n F(e_1,\ldots,e_{2i-2},\text{false},\text{true})$ is true.

At t_3, there are pebbles on d_i, $\bar{x}_i$, x_i', and q_{i+1}. Vertices q_i, a_i, b_i, c_i, f_i, and g_i are vacant because they can't be pebbled before q_{i+1} is pebbled. Vertex $\bar{x}_i'$ couldn't have been repebbled between t_2+1 and t_3 because three pebbles were fixed on d_i, $\bar{x}_i$, and x_i' during that interval; thus $\bar{x}_i'$ and (by normality) the entire s_i-2 -pyramid are also vacant at t_3. There may or may not be a pebble on x_i at t_3.

We will now show that immediately after t_3, a configuration in N_{i+1} with a true assignment to x_i is created, and that q_{i+1} must be repebbled while the pebbles in the configuration are fixed.

By frugality, the pebble on q_{i+1} at t_3 remains until either c_i or g_i is pebbled. Vertex q_{i+1} cannot retain a pebble until g_i is pebbled, because to pebble g_i requires placing all but two of the pebbles on the s_i-2 -pyramid, and in addition to the pebble on q_{i+1}, two pebbles are fixed, one on x_i', x_i, or f_i and the other on d_i, b_i, or a_i, until q_i is pebbled. Thus the pebble on q_{i+1} at t_3 remains until c_i is pebbled and is removed before g_i is pebbled. Since $\bar{x}_i$ has a pebble at t_3, we can rearrange the strategy so that the pebble on q_{i+1} is moved to c_i at t_3+1.

Now the only successors of c_i and b_i are b_i and a_i respectively, and since d_i and x_i' both contain pebbles at t_3+1, we can rearrange the strategy so that the pebble on c_i is moved to b_i at t_3+2 and to a_i at t_3+3. A pebble must then remain on a_i until q_i is pebbled. Since a_i is only pebbled once and is the only successor of x_i' except x_i, we can further rearrange the strategy so that the pebble on x_i' is moved to x_i at t_3+4 (or is picked up if there is already a pebble on x_i).

At t_3+4, a_i contains a pebble which will remain until q_i is pebbled, and x_i contains a pebble which will remain until f_i is pebbled. Vertex $\bar{x}_i'$ must be repebbled before f_i is pebbled, which must happen before q_i is pebbled. To pebble $\bar{x}_i'$ requires all the pebbles except the ones on a_i and x_i, so by normality $\bar{x}_i'$ is pebbled immediately after t_3+4, and is only pebbled once before f_i is pebbled. Let t_4 be the time all the pebbles except the one on $\bar{x}_i'$ are removed from the s_i-2 -pyramid after $\bar{x}_i'$ is first pebbled after t_3+4. At t_4 there are pebbles on a_i, x_i, and $\bar{x}_i'$, and nowhere else on the i-th quantifier block. This configuration is in N_{i+1} with a true assignment to x_i, and none of the pebbles on the graph at t_4 can be removed until after q_{i+1} is repebbled. By the induction hypothesis, $Q_{i+1}x_{i+1} \cdots Q_n x_n F(e_1,\ldots,e_{2i-2},\text{true},\text{false})$ is true. Therefore $\forall x_i Q_{i+1}x_{i+1} \cdots Q_n x_n F(e_1,\ldots,e_{2i-2})$ is true. This finishes the inductive step for a universal quantifier.

<u>Case 2</u> (existential quantifier). Suppose $Q_i = \exists$. By frugality, each of q_i, a_i, b_i, c_i, d_i, and q_{i+1} is only pebbled once. Exactly as in Case (1), normality implies that x_i' is only pebbled once, and is pebbled before anything else happens. A pebble remains on x_i' or x_i until q_i is pebbled, and a pebble remains on x_i' or $\bar{x}_i'$ until a_i is pebbled. To pebble a_i requires pebbling d_i, which requires removing all pebbles from the graph except one on x_i'. Thus d_i is pebbled before anything else except x_i', and a pebble remains on d_i until b_i is pebbled.

To pebble b_i requires pebbling c_i and hence f_i. To pebble f_i requires removing all pebbles except those on x_i' and d_i. Thus f_i is pebbled only once before b_i is pebbled, and this happens immediately after d_i is pebbled. A pebble remains on f_i, $\bar{x}_i'$, or $\bar{x}_i$ until c_i is pebbled.

The only successor of f_i is $\bar{x}_i'$, and a pebble remains on x_i' until $\bar{x}_i'$ is pebbled, so we can rearrange the strategy so that the first move after picking up the pebbles on the s_i-2 -pyramid (except the one on f_i) is to move the pebble on f_i to $\bar{x}_i'$. Let t_1 be the time of this move, and let t_2 be the time q_{i+1} is pebbled. Note that since f_i is not repebbled between t_1 and t_2, neither is $\bar{x}_i'$. At t_1 there are pebbles on x_i', $\bar{x}_i'$, and d_i, and until t_2 there must be pebbles on x_i' or x_i, x_i' or $\bar{x}_i'$, $\bar{x}_i'$ or $\bar{x}_i$, and d_i.

<u>Case 2a</u>. The pebble on x_i' is removed before t_2. Since the only successors of x_i' are x_i and $\bar{x}_i'$, and $\bar{x}_i'$ is not repebbled before t_2, we can rearrange the strategy so that the pebble on x_i' is moved to x_i at t_1+1. The configuration at t_1+1 is then in N_{i+1} with a true assignment to x_i, and none of the pebbles can be removed until t_2. By the induction hypothesis, $Q_{i+1}x_{i+1} \cdots Q_n x_n F(e_1,\ldots,e_{2i-2},\text{true},\text{false})$ is true.

<u>Case 2b</u>. A pebble remains on x_i' until t_2, and the pebble on $\bar{x}_i'$ is removed before t_2. We can rearrange the strategy so that the pebble on $\bar{x}_i'$ is moved to $\bar{x}_i$ at t_1+1. The configuration at t_1+1 is in N_{i+1} with a false assignment to x_i, and no pebble can be removed until t_2. By the induction hypothesis, $Q_{i+1}x_{i+1} \cdots Q_n x_n F(e_1,\ldots,e_{2i-2},\text{false},\text{true})$ is true.

<u>Case 2c</u>. Pebbles remain on x_i' and $\bar{x}_i'$ until t_2. The configuration at t_1 is in N_{i+1} with a double false assignment to x_i, and no pebble is removed until t_2. By the induction hypothesis, $Q_{i+1}x_{i+1} \cdots Q_n x_n F(e_1,\ldots,e_{2i-2},\text{false},\text{false})$ is true.

In each of subcases 2(a) - (c), $\exists x_i Q_{i+1}x_{i+1} \cdots Q_n x_n F(e_1,\ldots,e_{2i-2})$ is true. This completes the inductive step for an existential quantifier, and the proof of the lemma. ☐

Proof of Theorem 1. Theorem 1 is simply the case $i = 1$ of Lemmas 2 and 3. □

3. Remarks.

Variants of our construction give a couple of additional interesting results. Lingas [8] exhibited an infinite family of graphs with the following property: pebbling an n-vertex graph in the family with the minimum number of pebbles requires $\Omega\left(2^{\sqrt[3]{n}}\right)$ time, but allowing two additional pebbles reduces the time to $O(n)$. Van Emde Boas and van Leeuwen [2] independently obtained a similar result; in their construction only one additional pebble is necessary to reduce the pebbling time to $O(n)$.

We can obtain the same result as follows: Select any value of k. Let $s = 3k+2$. Construct a graph G_k corresponding to the formula

$$\forall x_1 \forall x_2 \ldots \forall x_k (x_1 \vee \bar{x}_1) \wedge (x_2 \vee \bar{x}_2) \wedge \ldots \wedge (x_k \vee \bar{x}_k)$$

as described in Section 2, representing each clause by a three-source pyramid as in Fig. 7 instead of by a four-source pyramid as in Fig. 5. G_k has $O(k^3)$ vertices and requires at least s pebbles, since it contains a pyramid of size s. Since the formula is true, G_k can be pebbled with s pebbles, but only in $\Omega(2^k)$ time, since any double false substitution makes the formula false. With $s+1$ pebbles, G_k can be pebbled in $O(k^3)$ time by selecting the double false assignment for all variables and using the remaining three pebbles to pebble the clause pyramids.

Another variant of our construction shows the following problem to be P-space complete: given a graph G and a number of pebbles s sufficient to pebble a given vertex v, can v be pebbled within a specified time bound t? We assume t is expressed in binary notation; if t is expressed in unary, it is immediate from Sethi's result [14] that the problem is NP-complete. We shall reduce the quantified Boolean formula problem to this problem of pebbling with a time bound.

Let $E = Q_1 x_1 \ldots Q_n x_n F$ be a quantified Boolean formula. Construct a new formula $E' = \exists y_1 \ldots \exists y_n Q_1 x_1 \ldots Q_n x_n F'$, where F' is formed from F by adding clauses $x_i \vee \bar{x}_i \vee y_i$ and $x_i \vee \bar{x}_i \vee \bar{y}_i$ to F for $1 \leq i \leq n$. The new formula E' is true if and only if E is true, but a double false substitution for any universally quantified variable in E' makes F' false. Let m be the number of clauses in F' (note that $m \geq 2n$), and let $n_\forall$ be the number of universally quantified variables in E'. Construct a formula

$E'' = \forall z_1 \forall z_2 \ldots \forall z_k \exists y_1 \ldots \exists y_n Q_1 x_1 \ldots Q_n x_n F''$ from E', where F'' is formed from F' by replacing every clause $\ell_{j1} \vee \ell_{j2} \vee \ell_{j3}$ by the set of clauses $\{\ell_{j1} \vee \ell_{j2} \vee \ell_{j3} \vee z_i \vee \bar{z}_i \mid 1 \leq i \leq k\}$. Here k is a suitably large integer whose value we shall select later.

Let $s = 3k+6n+5$. Construct a graph G corresponding to the new formula as in Section 2, using a pyramid of size six to represent each clause. Since E'' is true, G can be pebbled with s pebbles. If a double false substitution is made for some variable z_i in E'', the resulting formula is true if and only if the original formula E is true. Thus if E is false, pebbling G requires $\Omega\left(mk2^{k+n_\forall}\right)$ time. If E is true, G can be pebbled in $O\left((mk+(2n+k)^3)2^{n_\forall}\right)$ time by selecting the double false assignment for every variable z_i. Thus if k is sufficiently large ($k = m$ suffices for large m), there is a time $t(m)$ such that G can be pebbled with s pebbles in time $t(m)$ if and only if E is true.

Acknowledgement. We would like to thank Michael Loui for suggesting the existential quantifier block in Fig. 4, which considerably simplified our original construction.

References.

[1] S. Cook, "An observation on time-storage trade off," Proc. Fifth Annual ACM Symp. on Theory of Computing (1973), 29-33.

[2] P. van Emde Boas and J. van Leeuwen, "Move rules and trade-offs in the pebble game," Technical Report RUU-CS-78-4, Dept. of Computer Science, University of Utrecht, Utrecht, The Netherlands (1978).

[3] S. Even and R. E. Tarjan, "A combinatorial problem which is complete in polynomial space," Journal ACM 23 (1976), 710-719.

[4] A. S. Fraenkel, M. R. Garey, D. S. Johnson, T. Schaefer, and Y. Yesha, "The complexity of checkers on an N x N board -- preliminary report," Proc. Nineteenth Annual Symp. on Foundations of Computer Science (1978), 55-64.

[5] M. R. Garey and D. S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, Freeman, San Francisco, 1979.

[6] J. E. Hopcroft, W. Paul, and L. Valiant, "On time versus space," Journal ACM 24 (1977), 332-337.

[7] D. Lichtenstein and M. Sipser, "GO is P-space hard," Proc. Nineteenth Annual Symp. on Foundations of Computer Science (1978), 48-54.

[8] A. Lingas, "A PSPACE complete problem related to a pebble game," Automata, Languages, and Programming, Fifth Colloquium, Udine (1978), 300-321.

[9] M. S. Paterson and C. E. Hewitt, "Comparative schematology," Record of Project MAC Conference on Concurrent Systems and Parallel Computation (1970), 119-128.

[10] W. Paul and R. E. Tarjan, "Time-space trade-offs in a pebble game," Acta Informatica 10 (1978), 111-115.

[11] N. Pippenger, "A time-space trade-off," Journal ACM 25 (1978), 509-515.

[12] W. J. Savitch, "Relationships between nondeterministic and deterministic tape complexities," J. Computer and System Sciences 4 (1970), 177-192.

[13] T. J. Schaefer, "Complexity of decision problems based on finite two-person perfect-information games," Proc. Eighth Annual ACM Symp. on Theory of Computing (1976), 41-49.

[14] R. Sethi, "Complete register allocation problems," SIAM J. Comput. 4 (1975), 226-248.

[15] L. J. Stockmeyer and A. R. Meyer, "Word problems requiring exponential time: preliminary report," Proc. Fifth Annual ACM Symp. on Theory of Computing (1973), 1-9.

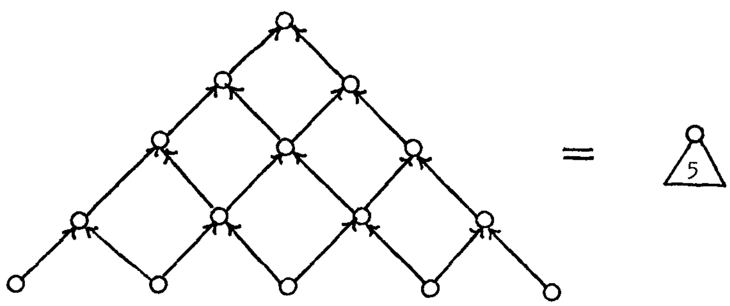

Figure 1. A 5 -pyramid.

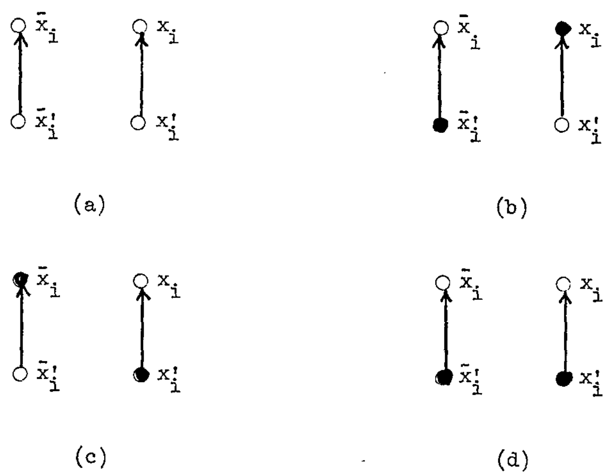

Figure 2. (a) Vertices representing a variable.

(b) True configuration.

(c) False configuration.

(d) Double false configuration.

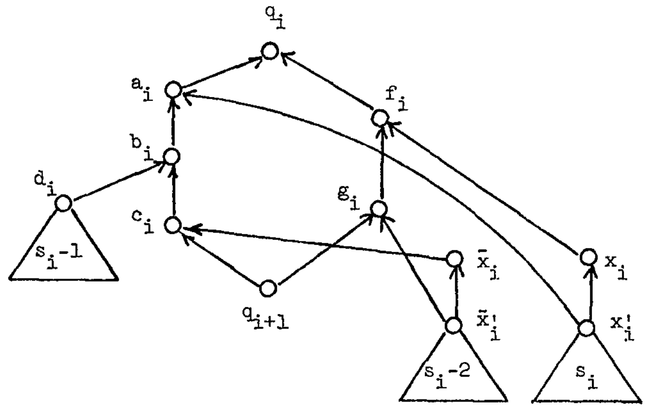

Figure 3. Universal quantifier block. Vertex q_{i+1} is part of the $i+1$ -st quantifier block.

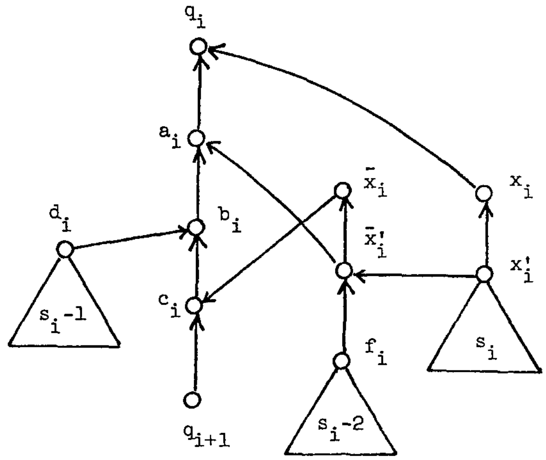

Figure 4. Existential quantifier block. Vertex q_{i+1} is part of the $i+1$ -st quantifier block.

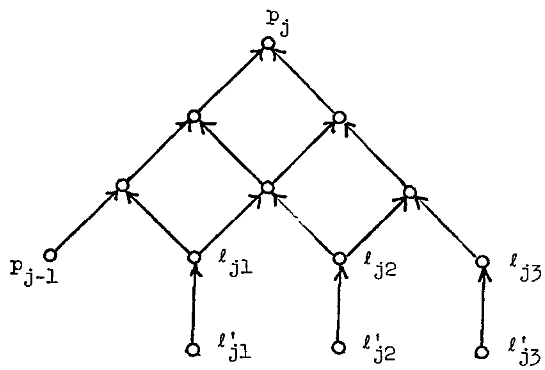

Figure 5. Block of vertices for clause $\ell_{j1} \vee \ell_{j2} \vee \ell_{j3}$. Note that the vertices ℓ_{jk} and ℓ'_{jk} occur among the quantifier blocks. Vertex p_{j-1} is part of the $j-1$ -st clause block; p_0 is a single vertex and $p_m = q_{n+1}$.

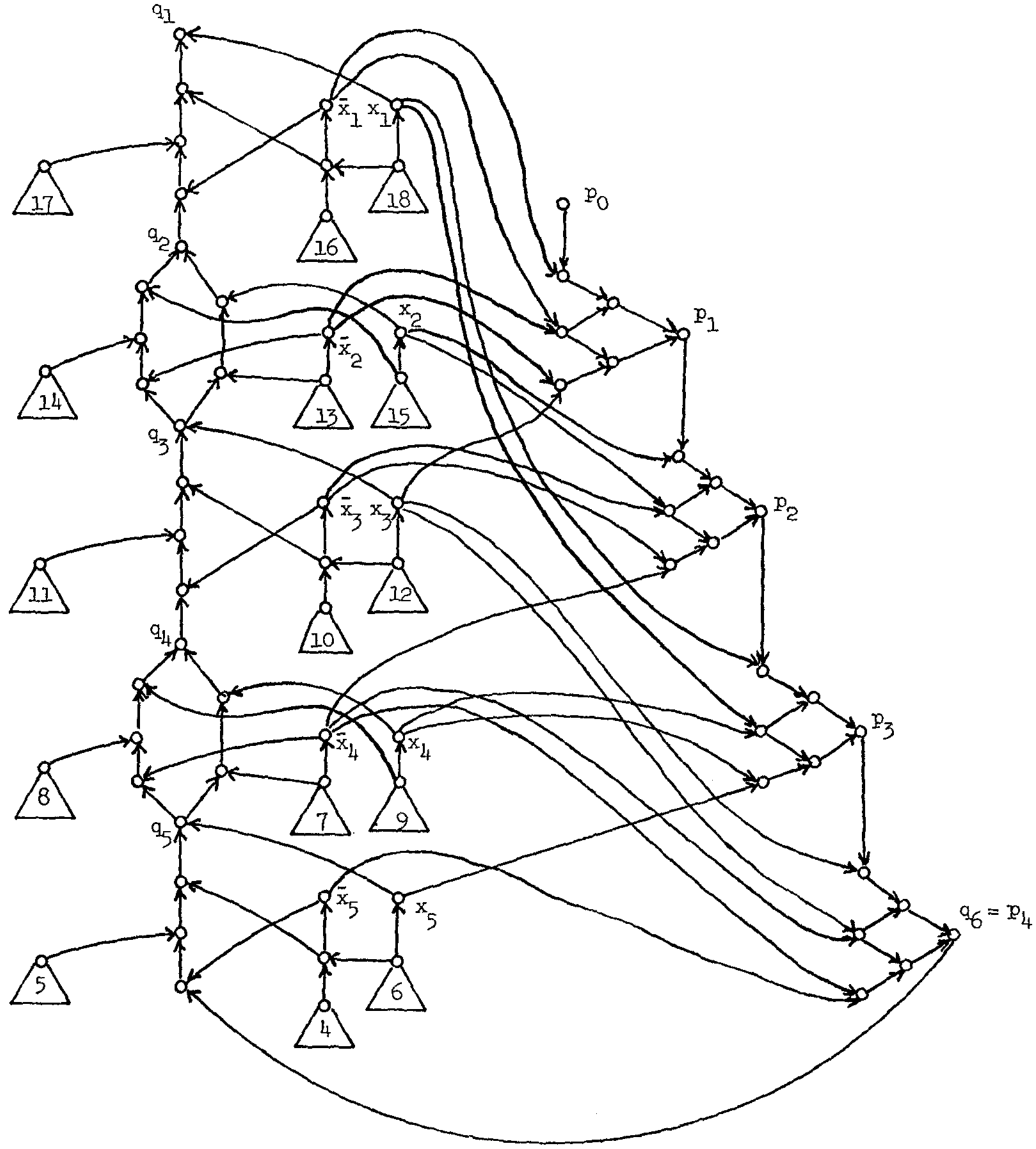

Figure 6. Graph for $E = \exists x_1 \forall x_2 \exists x_3 \forall x_4 \exists x_5 (\bar{x}_1 \vee \bar{x}_2 \vee x_3) \wedge (x_2 \vee \bar{x}_3 \vee \bar{x}_4)$
$\wedge (x_1 \vee x_4 \vee x_5) \wedge (x_3 \vee \bar{x}_4 \vee \bar{x}_5)$.

Number of pebbles $= 5 \cdot 3 + 3 = 18$.

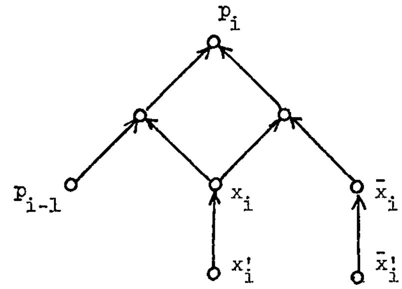

Figure 7. Block of vertices for clause $x_i \vee \bar{x}_i$.