

ON RELATING TIME AND SPACE TO SIZE AND DEPTH*

ALLAN BORODIN†

Abstract. Turing machine space complexity is related to circuit depth complexity. The relationship complements the known connection between Turing machine time and circuit size, thus enabling us to expose the related nature of some important open problems concerning Turing machine and circuit complexity. We are also able to show some connection between Turing machine complexity and arithmetic complexity.

Key words. space, depth, time, size, computational complexity, parallel arithmetic complexity, parallel time

1. Introduction. Fischer and Pippenger [7] have shown that a $T(n)$ time bounded Turing machine (TM) can be simulated on n bits by a combinational (Boolean) circuit with $O(T(n) \log T(n))$ gates (see also Schnorr [25]). In this paper, we observe that nondeterministic $S(n)$ tape bounded Turing machines can be simulated by circuits of depth $O(S(n)^2)$. In doing so, we relate the power of nondeterminism for space bounded computations to the depth required for the transitive closure problem. As a consequence of this development we show a relationship between the TIME-SPACE problem and the SIZE-DEPTH problem (equivalently the SIZE-FORMULA SIZE problem).

There has always been some ambiguity between the terminology of circuit complexity and TM complexity. In particular, depth is sometimes referred to as "time". But here "time" implicitly means *parallel time*, since several gates in a combinational circuit can operate in parallel. (In arithmetic complexity, depth is almost always referred to as parallel time.) In Pratt and Stockmeyer [22], we are introduced to vector machines, a general parallel machine model; general in the sense that inputs can be of arbitrary length (see also Hartmanis and Simon [9]). It is then shown that polynomial time (that is, parallel time) for these vector machine corresponds to TM polynomial space. Motivated by the simulations of Pratt and Stockmeyer [22] and Hartmanis and Simon [9], it is not hard to see that the crux of our space simulation should rely on the transitive closure problem. In the next section, we define the models more carefully and present the basic simulation. In § 3, we discuss relationships between TM and circuit complexity. In § 4, we conclude with some observations concerning arithmetic complexity.

2. Turing machines, circuits and the basic simulation. We assume that the reader is familiar with Chapters 6 and 10 of Hopcroft and Ullman [11]. Our TM model is an "off-line" machine with a two-way read only input tape. For time bounded computations we allow an arbitrary but finite number of work tapes. For tape bounded computations, it is sufficient to have just one work tape. The benefit of the read only tape is that it allows us to consider tape bounds less than the length of the input. Usually we consider Turing machines as acceptors or recognizers but

* Received by the editors March 30, 1976, and in revised form December 6, 1976.

† Department of Computer Science, University of Toronto, Toronto, Ontario, Canada M5S 1A7. This research was supported by the National Research Council of Canada.

sometimes we will need to consider transducers, in which case we append a write only output tape.

For a nondeterministic machine M , say that $L \subseteq \Sigma^*$ is *accepted* in time $T(n)$ (space $S(n)$) by M if $w \in L$ iff on input w there is a valid computation leading to an accepting state which uses $\leq T(|w|)$ steps (respectively, $S(|w|)$ work tape cells). Here $|w|$ denotes the length of the string. Without loss of generality, we can restrict ourselves to $\Sigma = \{0, 1\}$.

A combinational (Boolean) circuit is a labeled acyclic, directed graph (a network). Nodes with in-degree = 0 are called *input nodes*, nodes with out-degree = 0 are called *output nodes*. Interior nodes (i.e. noninput nodes including output nodes) represent (i.e. are labeled with) logical gates $f: \{\text{true}, \text{false}\}^r \rightarrow \{\text{true}, \text{false}\}$. Since we shall only be concerned with asymptotic complexity bounds, without loss of generality we can use the complete basis AND (denoted \wedge), inclusive OR (\vee) and NOT (\neg). (See Savage [23, pp. 662, 663].) We can have arbitrary fan-out and allow the constants $\{\text{true}, \text{false}\}$ as inputs. By associating true with '1' and false with '0', we think of every Boolean circuit as realizing a function $f: \{0, 1\}^n \rightarrow \{0, 1\}$. That is, let $A^n \subseteq \{0, 1\}^n$; we say A^n is *realized by circuit C* if C has n nonconstant input nodes (labeled x_1, \dots, x_n) and C accepts (i.e. outputs 1 = true) iff $x_1 x_2 \dots x_n$ is in A^n . As a notational convenience, if $A \subseteq \{0, 1\}^*$, let $A^n = A \cap \{0, 1\}^n$.

The size of a circuit C is the number of interior nodes or gates, and the depth of C is the length of the longest path in C . We will also have need to encode a circuit C as a string \bar{C} in $\{0, 1\}^*$. This can be done in a straightforward way; i.e. topologically order the network, give addresses to each of the nodes, and then a circuit can be given as a sequence of instructions.

If the output of C depends on all n inputs x_1, \dots, x_n , the size of C must be $\geq n - 1$; it follows that $|\bar{C}| \geq d \cdot \text{size } C \cdot \log \text{size } C$ for some constant $d > 0$. (Note that the chosen basis implies fan-in ≤ 2 .)

Finally, we let $\text{SIZE}_A(n)$ (respectively $\text{DEPTH}_A(n)$) be the minimum size (depth) required for a circuit to realize A^n . Using this notation, we recall the time-size simulation result.

THEOREM 1 (Fischer and Pippenger [7]). *Let A be recognized by a deterministic $T(n) \geq n$ time bounded TM. Then there exists $d > 0$ such that $\text{SIZE}_A(n) \leq d \cdot T(n) \cdot \log T(n)$.*

Analogously, we have the following:

THEOREM 2. *Let A be accepted by a nondeterministic $S(n) \geq \log_2 n$ space bounded TM. Then there exists a $d > 0$ such that $\text{DEPTH}_A(n) \leq d \cdot S(n)^2$.*

Proof. Let M be a $S(n)$ tape bounded nondeterministic TM with $S(n) \geq \log_2 n$. Say M has q states and s symbols on its work tape and that the input $w \in \{0, 1\}^n$. Thinking of w as fixed, the computation sequence is "determined" by the state, the input tape head position, the work tape head position, and the contents of the work tape. Thus there are at most $N = q \cdot n \cdot S(n) \cdot s^{S(n)}$ configurations. We can think of the acceptance problem as the transitive closure problem for a graph with N nodes, whose edges correspond to the allowable moves determined by M and w .

Let $X = (x_{ij})$ where $x_{ij} = 1$ iff there is a move from configuration i to configuration j , and let $X^* = (x_{ij}^*)$ be the transitive closure. That is, $x_{ij}^* = 1$ iff there is a path from configuration i to configuration j . (See Fig. 1.)

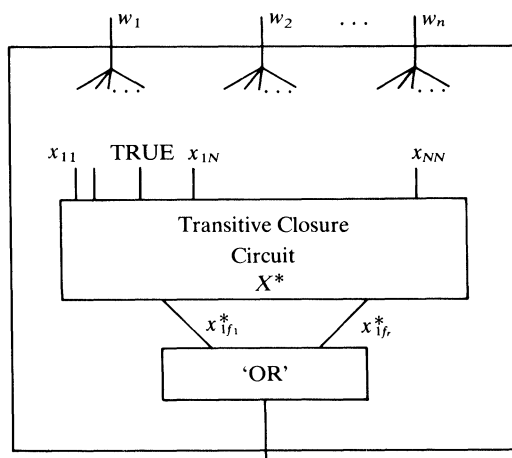


FIG. 1

Let i be the number of a configuration corresponding to the input head being on square k . Then

- w_k is directly connected to x_{ij} iff there is a move from configuration i to j *only when* the k th input bit is 1.
- w_k is negated and then connected to x_{ij} iff there is a move from configuration i to j *only when* the k th input is 0.
- w_k is not connected to x_{ij} iff there is (is not) a move from configuration i to j independent of the value of the k th input bit. In this case x_{ij} is set to the appropriate constant.

Let 1 be the number of the starting configuration and let f_j ($1 \leq j \leq r$) correspond to accepting configurations. By using $\log(N-1)$ levels of Boolean matrix multiplication, it is well known that an $N \times N$ transitive closure circuit requires only $\log^2 N$ depth. Also, an N -way 'OR' can obviously be realized with depth $\log N$. The theorem follows since $N = q \cdot n \cdot S(n) \cdot s^{S(n)}$ and therefore $\log N < d_1 \cdot S(n)$.

Remark. Let A be accepted by M as in Theorem 2. With a little care in the numbering of the configurations, the mapping $1^n \rightarrow \bar{C}_n$ (where C_n realizes A^n) is computable by a *deterministic* $S(n)$ space bounded transducer if $S(n)$ is tape constructible. This follows because of the "uniformly constructive" nature of the $\log^2 N$ depth transitive closure circuits (i.e. they can be generated in $\log N$ space). We shall have more to say about "uniformity" in § 3.

Open Problem 1. If M is *deterministic* $S(n)$ tape bounded, can we improve the simulation so that $\text{DEPTH}_A(n) \leq d \cdot S(n)$?

Since each output in the $N \times N$ transitive closure problem can be computed in nondeterministic $\log N$ space, it follows that improving the simulation for nondeterministic machines is equivalent to improving the depth required for the transitive closure problem. In particular, if the $N \times N$ transitive closure problem can be realized with depth $\leq c \cdot \log^\alpha n$ (where $1 \leq \alpha \leq 2$) then $\text{DEPTH}_A(n) \leq d \cdot S(n)^\alpha$ in Theorem 2.

Open Problem 2. Let A be recognized by a deterministic $S(n)$ tape, $T(n)$ time bounded machine. Can we realize A^n by circuits with $\text{SIZE}_A(n) \leq c_1 T(n)^{k_1}$ and (simultaneously) $\text{DEPTH}_A(n) \leq c_2 T(n)^{k_2}$ for some constants c_1, c_2, k_1, k_2 ?

Let \leq_{\log} represent log space reducibility (see Jones and Laaser [12]). We could allow nondeterministic transduction here but we might as well follow the standard meaning of deterministic log space many-one reducibility. By converting every log space transducer to one with a separate track keeping count (in binary) of the number of output bits thus far in the computation, we can generalize Theorem 2 as follows:

THEOREM 3. *Let $A \leq_{\log} B$. Then*

$$\text{DEPTH}_A(n) \leq d \cdot \log^2 n + \max_{m \leq cn^k} \text{DEPTH}_B(m)$$

for some constants c, d and k .

Proof. Let M be the log space transducer which reduces A to B . Then on input w of length n , M can output at most $N = cn^k$ bits since M is log n space bounded. Let y_i^0 (respectively, y_i^1) be the ‘OR’ of those configurations where the i th bit being output is a ‘0’ (respectively, ‘1’). If y_i^0 and y_i^1 are both false, then we know that M outputs less than i bits on input w . Knowing m , the exact number of bits output by M on w , we can “activate” the circuit C_m for B^m with inputs y_i^1 ($1 \leq i \leq m$). (See Fig. 2.)

Following standard notation, let P be the class of languages recognizable in deterministic polynomial time. B is called *log space complete* for P if

- (i) B is in P ,
- (ii) A in P implies $A \leq_{\log} B$.

Cook [3], Jones and Laaser [12] and Ladner [29] exhibit a variety of natural sets which are log space complete for P . We can define an analogous concept for circuits. Namely, let us say that B is *depth complete for polynomial size circuits* if

- (i) $\text{SIZE}_B(n) \leq p(n)$ for some polynomial p .
- (ii) Let A be such that $\text{SIZE}_A(n) \leq p_1(n)$ for some polynomial p_1 . Then there exist constants c and k , and a polynomial q such that for all n there is an n -input circuit T_n with the following properties:

- (a) $\text{depth } T_n \leq c \cdot \log^k n$

Note: of course, we would like $k = 1$.

- (b) T_n outputs $\langle y_1, \dots, y_N \rangle = f_1(\langle x_1, \dots, x_n \rangle)$ and $\langle z_1, \dots, z_N \rangle = f_2(\langle x_1, \dots, x_n \rangle)$ for some fixed $N \leq q(n)$.
- (c) there is a unique output z_m with value 1 and for this m we have $x_1 \cdots x_m$ in A^n iff $y_1 \cdots y_m$ in B^m .

Our definition has been chosen so that the construction in Theorem 3 immediately yields:

COROLLARY 1. *If B is log space complete for P , then B is depth complete for polynomial size circuits.*

We think of depth (log space) complete sets as being “hardest polynomially computable sets” with respect to depth (space) requirements. Ladner defines the following “circuit value problem” and shows it to be log space complete for P :

$$V = \{x_1 \cdots x_n \mid \bar{C} \mid C \text{ outputs true on input } x_1 \cdots x_n\}.$$

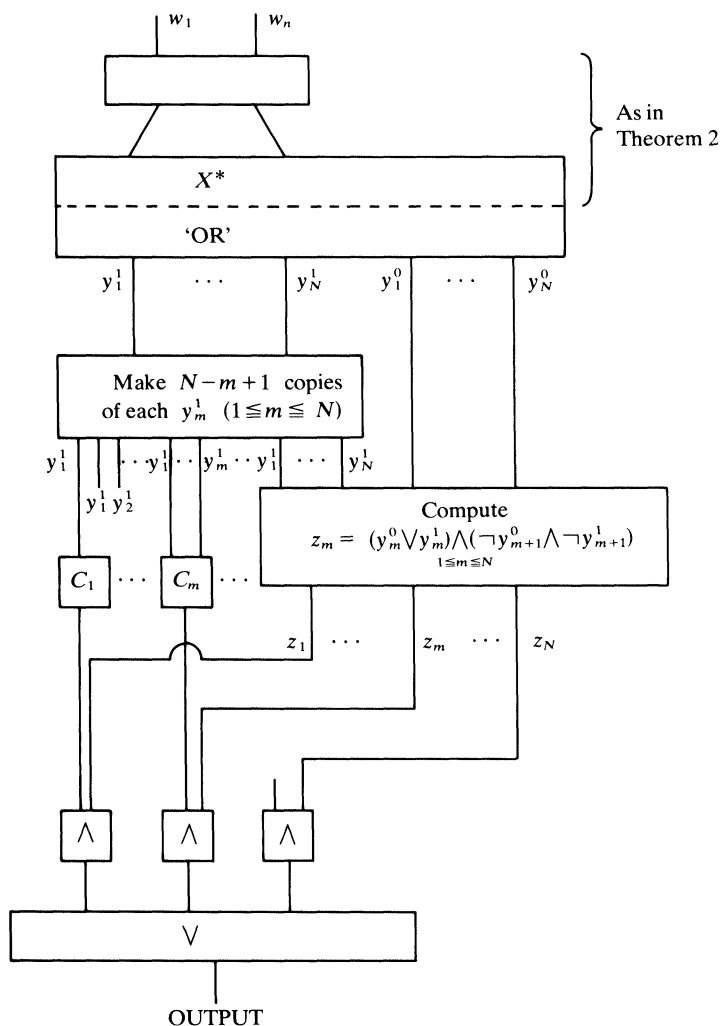


FIG. 2

Corollary 1 verifies the obvious fact that V is depth complete for polynomial size circuits.

3. Relating open problems in Turing machine and circuit complexity. As usual, we let $\text{DTIME}(T(n))$ (respectively, $\text{DSpace}(S(n))$, $\text{NSpace}(S(n))$) denote the class of languages accepted in deterministic Time $T(n)$ (respectively, deterministic and nondeterministic space $S(n)$). Analogously, define $\text{SIZE}(T(n)) = \{B | \text{SIZE}_B(n) \leq c \cdot T(n) \text{ for some constant } c\}$ and $\text{DEPTH}(S(n))$

$= \{B \mid \text{DEPTH}_B(n) \leq c \cdot S(n)\}$. We first want to complete the relationship between TM space and circuit depth and thus we need a “converse” to Theorem 2.

LEMMA 1. *Let V be the circuit value problem. There is a deterministic TM M which recognizes V such that on input $x_1 \cdots x_n \neq \bar{C}$, M only uses space bounded by $\text{Depth } C + \log \text{size } C$.*

Proof. The idea is just to recursively evaluate (say, first the left and then the right) inputs to a given gate. A straightforward implementation using a pushdown store would require depth $C \cdot \log \text{size } C$ storage (i.e. depth C levels in the store, $\log \text{size } C$ space for each gate address entry). We use a suggestion by S. Cook to improve this to the desired bound. In evaluating the output of C , we only need to store the full address of the gate currently being evaluated while the status (e.g. left input has value “false” and right input now being evaluated) of each gate on the pushdown store can be accommodated within constant space. We can recompute the address of any gate on the stack by working up from the bottom of the stack (i.e. the circuit output gate) via the status entries.

DEFINITION. We say that A is *uniformly* in $\text{DEPTH}(S(n))$ if there is a constant c such that for all n , there is a circuit \bar{C}_n of depth $\leq c \cdot S(n)$ realizing A^n and, moreover, \bar{C}_n can be generated in deterministic space $S(n)$; i.e. the transformation $1^n \rightarrow \bar{C}_n$ is deterministic $S(n)$ space computable.

THEOREM 4. *Suppose A is uniformly in $\text{DEPTH}(S(n))$, $S(n) \geq \log n$. Then A is in $\text{DSPACE}(S(n))$.*

Note. We view this as a “converse” to Theorem 2 for that result can be stated as: “If A is in $\text{NSPACE}(S(n))$, then A is uniformly in $\text{DEPTH}(S(n)^2)$ ”.

Proof. Given an input $w = x_1 \cdots x_n$, we apply Lemma 1. Now whenever we need to know the i th bit of \bar{C}_n (as in the lemma), we compute it (in the required space) by using the uniformity hypothesis.

It is well-known that one can define arbitrarily complex or nonrecursive sets A such that for all n , $A^n = \phi$ or $A^n = \Sigma^n$. Since a trivial circuit realizes A^n for each n , it is clear that A in $\text{DEPTH}(S(n))$ does not imply that A in $\text{DSPACE}(S'(n))$ for any $S'(n) \geq S(n)$. In order to relate space and depth, we chose to assert a uniformity condition on the circuits. There is another choice. Following Schnorr [25], Meyer and Stockmeyer [17] suggest “making the Turing machines nonuniform” by giving them oracles. Then they observe that our Theorems 2 and 4 can be modified as follows:

- (a) If A is recognized by a nondeterministic $S(n)$ space bounded TM M with a $\{0, 1\}^*$ oracle, then A is in $\text{DEPTH}(S(n)^2)$.
- (b) If A is in $\text{DEPTH}(S(n))$, then A is recognized by a deterministic $S(n)$ space bounded TM M with a $\{0, 1\}^*$ oracle.

In (b), the oracle is used to encode the appropriate efficient circuit. In both (a) and (b) we count the space needed for the oracle tape questions. This formulation does have a very nice mathematical appeal. We have, however, chosen to assert the uniformity of circuits because from a “practical” point of view, experience tells us that if we can show A is in $\text{DEPTH}(S(n))$, then we usually can show A is uniformly in $\text{DEPTH}(S(n))$. The same choice also exists for the TIME-SIZE relationship. Here “uniformly in $\text{SIZE}(T(n))$ ” means that we can generate the appropriate circuit description in deterministic time $T(n)$. Then we can relate

uniform size and time or (see Schnorr [25]) we can instead relate size and time of Turing machines with oracles. Using Schnorr's [25] model, nonuniformity takes form in a $\{0, 1\}^*$ oracle explicitly listed on a separate tape; Meyer [16] shows that with the more conventional model which uses a separate tape for inputs to an oracle, nonuniformity for time takes form in a $\{0\}^*$ oracle.

We can use Theorems 2 and 4 to make explicit the role of transitive closure in Savitch's [24] construction. Throughout the remainder of this paper we let α ($1 \leq \alpha \leq 2$) be such that the $N \times N$ transitive closure problem can be realized (uniformly) with depth $\leq c \cdot \log^\alpha N$.

COROLLARY 2. *Let α be as above. Then $\text{NSPACE}(S(n)) \subseteq \text{DSPACE}(S(n)^\alpha)$ for all tape constructible $S(n) \geq \log n$.*

Proof. Let M be a nondeterministic $S(n)$ space bounded TM accepting A .

For every input $w = x_1 \cdots x_n$ we can generate a circuit C_n corresponding to M, w (as in Theorem 2). By hypothesis, this can be done in deterministic space $S(n)$, and depth $C_n \leq c \cdot S(n)^\alpha$ for some constant c (see the remarks following Theorem 2).

That is, A is uniformly in $\text{DEPTH}(S(n)^\alpha)$ and hence A is in $\text{DSPACE}(S(n)^\alpha)$. More constructively, by using Theorem 2 and Lemma 1, we can produce a deterministic $S(n)^\alpha$ tape bounded machine M' recognizing A . \square

One of the most important problems in computational complexity concerns efficient space simulations of time bounded computations. In particular, there is a conjecture that $\text{DTIME}(T(n)) \subseteq \text{DSPACE}(\log^k T(n))$ for some constant k . (Indeed, $k = 1$ is still possible.) Cook [3], and Cook and Sethi [4] present important evidence that the conjecture is false, and this represents the consensus of opinion at this time. On the positive side, Hopcroft, Paul and Valiant [10] have shown that $\text{DTIME}(T(n)) \subseteq \text{DSPACE}(T(n)/\log T(n))$. Independent of this result (and independent of our observations), Paterson and Valiant [19] proved that $\text{SIZE}(T(n)) \subseteq \text{DEPTH}(T(n)/\log T(n))$, noting that their result only had significance when $T(n) \approx n$. The known relationships between TIME-SIZE, and SPACE-DEPTH are not refined enough to show that either of these results follows from the other, but we can show that the problems are related.

COROLLARY 3. *Suppose $\text{DTIME}(n) \subseteq \text{NSPACE}(S(n))$. Then $\text{SIZE}(T(n)) \subseteq \text{DEPTH}(S'(n))$ uniformly for any time constructible $T(n)$ where $S'(n) = [S(T(n) \log^3 T(n))]^\alpha$. Here uniformly means that the required $cS'(n)$ depth circuit can be constructed in deterministic space $S'(n)$. Recall $\alpha \leq 2$.*

Proof. Given a $T(n)$ size circuit C , we encode it as a word \bar{C} of length $\leq c \cdot T(n) \cdot \log T(n)$. A straightforward circuit simulation by a TM can be performed in deterministic time m^2 , where m is the length of the circuit description. Recently, Pippenger [21] showed how to recognize the circuit value problem V in time $m \cdot \log^2 m$. By hypothesis, and using a standard translation argument, we have $\text{DTIME}(T(n)) \subseteq \text{DSPACE}(S[T(n)])$. Hence there is a deterministic TM M recognizing V in space $S(m \cdot \log^2 m)$; in particular, $x_1 \cdots x_n \# \bar{C}$ will be accepted or rejected in space $S(m \cdot \log^2 m) = S(T(n) \cdot \log^3 T(n))$. Hence there is a circuit C' of depth $c \cdot S'(n)$ which realizes $x_1 \cdots x_n \# \bar{C}$. Finally we can fix the input gates for \bar{C} and the resulting circuit has depth $\leq c \cdot S'(n)$.

For example, we have " $\text{DTIME}(n) \subseteq \text{NSPACE}(\log^k n)$ implies $\text{SIZE}(T(n)) \subseteq \text{DEPTH}(\log^{k\alpha} T(n))$ uniformly".

COROLLARY 4. *Suppose $\text{SIZE}(n) \subseteq \text{DEPTH}(S(n))$ uniformly. Then $\text{DTIME}(T(n)) \subseteq \text{DSPACE}(S[T(n) \cdot \log T(n)])$ for all constructible $T(n)$.*

Proof. Let M be $T(n)$ time bounded. We construct an equivalent M' . M' on input $w = x_1 \cdots x_n$ constructs a circuit C_n of size $c \cdot T(n) \log T(n)$ according to the Fischer–Pippenger simulation. (We claim with their oblivious $T \cdot \log T$ machine that this can be done in space $\log T(n)$.) Then, by hypothesis, we construct an equivalent circuit C'_n of depth $d \cdot S[T(n) \cdot \log T(n)]$ and finally apply Lemma 1 or Theorem 4 to produce the desired M' .

Again, for example, “ $\text{SIZE}(n) \subseteq \text{DEPTH}(\log^k n)$ uniformly implies $\text{DTIME}(T(n)) \subseteq \text{DSPACE}(\log^k T(n))$ ”. Note that Corollary 4 is “almost good enough” to derive the Hopcroft, Paul and Valiant [10] result from the Paterson and Valiant [19] construction. (The latter construction can be realized in space $n/\log n$.) Summarizing, we have shown that the TIME-SPACE problem for Turing machines is “roughly” equivalent to an “efficiently constructive” version of the SIZE-DEPTH problem for circuits.

Circuits with a fan-out = 1 restriction correspond to formulas. Spira [26] has shown that a formula of size $T(n) \geq n$ can be transformed to an equivalent formula of depth $\leq c \cdot \log T(n)$. (Consider also Brent’s [30] analogous result for arithmetic expressions). Moreover, it should be clear that formula size $\leq 2^{\text{depth}}$. Hence, if we are looking for an example where formula size is exponentially larger than (arbitrary circuit) size, we might as well look at any of the languages which are log space complete for P . It should be noted that Spira’s construction is reasonably uniform; that is, a formula of size n can be transformed within $\log^2 n$ space to an equivalent formula of depth $c \cdot \log n$. (This transformation should be compared with the hypothesis of Corollary 4.) We do not see how to construct a depth $c \cdot \log n$ formula in space $\log n$. However, using Lynch’s [15] $\log n$ formula evaluation in conjunction with Theorem 2, we can construct a depth $c \cdot \log^2 n$ formula in space $\log n$.

4. Some comments on arithmetic circuits. An arithmetic circuit is like a Boolean circuit except that now the inputs are indeterminates x_1, \dots, x_n (and possibly constants $c_i \in F$, F a field), the internal gates are $+$, $-$, \times , \div , and the outputs are considered to be elements of $F(x_1, \dots, x_n)$ (see Borodin and Munro [1]). For definiteness, let’s take $F = \mathbb{Q}$, the rationals. Strictly speaking, the size-depth question for arithmetic circuits is not a problem. Kung [14] has shown that x^{2^k} requires k depth (depth is called parallel time in the literature of arithmetic complexity) and x^{2^k} can obviously be realized with size (or sequential time) k . However, if one restricts attention to functions of small degree, the size-depth question is meaningful.

Throughout the remainder of this discussion, let us restrict our attention to the computation of multivariate polynomials or rational functions $p(x_1, \dots, x_n)$ of degree $\leq n$. (Also, if we do not allow arbitrary constants in \mathbb{Q} as inputs, then we should also restrict the coefficients occurring in p .)

To argue the case that arithmetic complexity and the more traditional studies of computational complexity are related, let us consider a current problem concerning parallel arithmetic computations. Csanky [5] has shown that if $\text{PWR}(A) = \{A^2, A^3, \dots, A^n \mid A \text{ an } n \times n \text{ matrix}\}$ is computable in $L(n)$ depth (parallel steps) then A^{-1} , $\det A$, coefficients of $\text{char}(A)$ would all be computable

in $O(L(n))$ depth. Schönhage has demonstrated that a converse also holds; specifically, $\text{PWR}(A)$ can be obtained from B^{-1} where

$$B = \begin{pmatrix} I & A & & & \\ & I & A & & \\ & & \ddots & \ddots & \\ & & & I & A \\ & & & & I \end{pmatrix}.$$

Now we know $\log n \leq L(n) \leq \log^2 n$ and the question arises as to whether or not $L(n) = O(\log n)$. To dramatize the consequences of Corollary 2, we can make the following observation.

COROLLARY 5. *We now consider only circuits with $+$, $-$, \times (no \div) and constants in Q .*

Suppose there is a deterministic $L(n)$ transformation $1^n \rightarrow \bar{C}_n$ which generates a depth $L(n)$ arithmetic circuit C_n realizing $A_{n \times n}^n$ (say $L(n) = \log^\beta n$). Then $\text{NSPACE}(S(n)) \subseteq \text{DSpace}(S(n)^\beta \log S(n))$ for all constructible $S(n) \geq \log n$.

Proof. Let $A^* = (a_{ij}^*)$ be the transitive closure (considered as a set of Boolean functions). Let $A = (a_{ij})$ be a matrix with $\{0, 1\}$ integer entries, and assume $a_{ii} = 1$, $1 \leq i \leq n$. Letting $\tilde{A} = (\tilde{a}_{ij}) = A^n$ (with respect to arithmetic matrix multiplication), we then have $a_{ij}^* = \min(\tilde{a}_{ij}, 1)$. Starting with a 0-1 matrix A , we know $\tilde{a}_{ij} \leq n^n$. We would like to simulate (integer) arithmetic as in Munro [18] and Fischer and Meyer [6] but “mod n^n ” arithmetic is $n \cdot \log n$ bit arithmetic and costs depth $\log n$. Instead following another suggestion by S. Cook, we can simulate the arithmetic mod p_i ($1 \leq i \leq m$) where $\{p_1, \dots, p_m\}$ are the first m primes and $\prod_{i=1}^m p_i \geq n^n$. Since by the prime number theorem the number of primes less than x is asymptotically equal to $x/\log x$, this can certainly be done with $p_m \leq cn \cdot \log^2 n$. (In the case of rational constants $q = r/s$, we must make sure that $s^{-1} \bmod p_i$ exists; that is, we choose our $\{p_i\}$ so that no such s is equal to 0 mod p_i . Since we are only considering circuits with $L(n) \leq \log^2 n$, the size of the circuit is $\leq n^{\log n}$ and so we need only avoid at most $n^{\log n}$ “bad primes”.) The depth cost of the mod p_i arithmetic results in a $\log \log n$ factor. We do not need to reconstruct any \tilde{a}_{ij} since $\tilde{a}_{ij} = 0$ iff $\tilde{a}_{ij} \bmod p_i = 0$ for all p_i . Thus a transitive closure circuit of depth $L(n) \cdot \log \log n$ can be constructed (by a deterministic $L(n)$ tape TM). Corollary 2 completes the proof.

Some discussion on Corollary 5 is appropriate. The restriction that C_n be *uniformly* generated is certainly necessary (see also Corollary 2). Our viewpoint, however, is that the discovery of a depth efficient method for A^n would be sufficiently constructive to yield the uniformity hypothesis. The restriction that \div is not allowed in Corollary 5 is both annoying and possibly unnecessary. It is annoying because division is obviously necessary for A^{-1} and thus any method for A^n which is derived from A^{-1} would use division. The problem with \div is that during the computation we might be dividing by a very large y for which $y \equiv 0 \bmod (p_i)$ for all small primes p_i . For the computation of a polynomial of degree n , one can use the method of Strassen [27] to eliminate \div , but this results in an $O(\log n)$ factor in the depth bound. It is not known whether this factor can be improved. Yet in spite of all our restrictions, one has “the feeling” that an

$O(\log n)$ depth method for any of the problems $\det A$, A^{-1} , A^n would lead to a positive solution to the LBA problem (i.e. $\text{NSPACE}(n) = \text{DSPACE}(n)$? See Hartmanis and Hunt [8]). The present consensus is that this is very unlikely; that is, $\text{NSPACE}(S(n)) \neq \text{DSPACE}(S(n))$ for constructible $S(n)$, and indeed any improvement to Savitch's $\text{NSPACE}(S(n)) \subseteq \text{DSPACE}(S(n)^2)$ would be a significant result for "traditional computational complexity".

Looking at Schönhage's observation on how to use A^{-1} to compute A^n , one sees that a depth efficient circuit for A^n can be composed of a $c \cdot \log n$ depth transformation $\langle y_1, \dots, y_m \rangle = f(\langle x_1, \dots, x_n \rangle)$ followed by a circuit for A^{-1} . In other words, we can define a reducibility for arithmetic circuits (as we could for Boolean circuits but here m should only depend on n) in analogy to the log space TM reducibility. Motivated by Corollary 1, and Paterson and Valiant [19], we are led to ask the following questions.

Open Problem 3. Is there a "natural" class of polynomial or rational functions which are depth complete for polynomial size arithmetic circuits? Can every rational function computable in size $T(n)$ be computed in depth $T(n)/\log T(n)$? (Note: we are still only considering rational functions $f(x_1, \dots, x_n)$ of degree $\leq n$.) Is depth $\log^\alpha T(n)$ possible?

In general, one cannot expect that positive results for Boolean computations always have arithmetic analogues. For example, Pippenger [20] shows that every Boolean symmetric function on n variables has formula size $\leq n^{3.6}$ (and hence depth $\leq c \cdot \log n$) whereas the elementary symmetric function $\sum_{1 \leq i_1 < i_2 < \dots < i_n} x_{i_1} x_{i_2} \dots x_{i_n}$ appears to need $O(\log^2 n)$ depth. Moreover, even positive results for arithmetic computations may not always have Boolean analogues. For example, we can simulate a Boolean circuit by an arithmetic circuit; i.e. $x \vee y$ is simulated by $x + y - x \times y$, $\neg x$ by $1 - x$. But even if every size $T(n)$ arithmetic circuit (computing functions of degree $\leq n$) was transformable into an equivalent depth $\log^\alpha T(n)$ circuit, a corresponding result would not necessarily hold for Boolean circuits. Suppose we try the following: Given Boolean circuit C_1 , first convert to the arithmetic circuit C_2 which "simulates" C_1 , then transform to an equivalent C_3 and finally obtain a depth efficient Boolean circuit C_4 by simulating C_3 with mod 2 arithmetic. The problem is that circuit C_2 (which is only "equivalent" to circuit C_1 for $\{0, 1\}$ valued inputs) may be computing arithmetic functions whose degrees can be exponential in the size of C_2 , and hence exponential in n . What is missing is the concept of the degree of a Boolean function. In any case, we consider it a major open problem to exhibit a "polynomial size" function, Boolean or arithmetic, which is provably not computable in $O(\log n)$ depth.

5. Conclusion. The main results of this paper establish a relation between TM space and circuit depth. This can be interpreted as another piece of evidence (see Pratt and Stockmeyer [22], Hartmanis and Simon [9] and more recently Chandra and Stockmeyer [2], Kozen [13], and Tzourakis [28]), that parallel time and space are roughly equivalent within a polynomial factor. The simplicity of the circuit model focuses our attention on the importance of the transitive closure problem. As a result, we have been able to unify a number of open problems in computational complexity. We also claim that questions in "traditional" computational complexity have relevance to arithmetic complexity and conversely.

Acknowledgments. The author is greatly indebted to S. Cook for a number of important suggestions, and to A. Meyer, L. Stockmeyer and the referees for their helpful comments.

REFERENCES

- [1] A. BORODIN AND I. MUNRO, *The Computational Complexity of Algebraic and Numeric Problems*, American Elsevier, New York, 1975.
- [2] A. K. CHANDRA AND L. STOCKMEYER, *Alternation*, Proc. of Seventeenth Annual Symp. on Foundations of Computer Science, Houston, Oct. 1976, pp. 98–108.
- [3] S. COOK, *An observation on time-storage tradeoff*, Proc. of the Fifth Annual ACM Symp. on Theory of Computing, May 1973, pp. 29–33.
- [4] S. COOK AND R. SETHI, *Storage requirements for deterministic polynomial time recognizable languages*, Proc. of Sixth Annual ACM Symp. on Theory of Computing, May 1974, pp. 33–39.
- [5] L. CSANKY, *Fast Parallel Matrix Inversion Algorithms*, Proc. of Sixteenth Annual Symp. on Foundations of Computer Science, Oct. 1975, pp. 11–12.
- [6] M. J. FISCHER AND A. MEYER, *Boolean matrix multiplication and transitive closure*, Proc. Twelfth Annual IEEE Symp. on Switching and Automata Theory, Oct. 1971, pp. 129–131.
- [7] M. FISCHER AND N. PIPPENGER, *M. J. Fischer Lectures on Network Complexity*, Universität Frankfurt, preprint, 1974.
- [8] J. HARTMANIS AND H. B. HUNT, *The LBA Problem and its Importance in the Theory of Computing*, SIAM-AMS Proc., vol. 7. American Mathematical Society, Providence, RI, 1974.
- [9] J. HARTMANIS AND J. SIMON, *On the power of multiplication in random access machines*, Proc. of Fifteenth Annual Symp. on Switching and Automata Theory, Oct. 1974, pp. 13–23.
- [10] J. HOPCROFT, W. PAUL AND L. VALIANT, *On time versus space and related problems*, Proc. of Sixteenth Annual Symp. on Foundation of Computer Science, Oct. 1975, pp. 57–64.
- [11] J. HOPCROFT AND J. ULLMAN, *Formal Languages and their Relation to Automata*, Addison-Wesley, Reading, MA, 1969.
- [12] N. D. JONES AND W. T. LAASER, *Complete problems for deterministic polynomial time*, Proc. Sixth Annual ACM Symp. on Theory of Computing, May 1974, pp. 40–46.
- [13] D. KOZEN, *On parallelism in Turing machines*, Proc. of Seventeenth Annual Symp. on Foundation of Computer Science, Houston, Oct. 1976, pp. 89–97.
- [14] H. T. KUNG, *Some complexity bounds for parallel computation*, Proc. of Sixth Annual ACM Symp. on Theory of Computing, May 1974, pp. 323–333.
- [15] N. LYNCH, *Log Space Recognition and Translation of Parenthesis Languages*, preprint.
- [16] A. MEYER, Personal communication, 1975.
- [17] A. MEYER, AND L. STOCKMEYER, Personal communication, 1975.
- [18] I. MUNRO, *Efficient determination of the transitive closure of a graph*, Information Processing Lett., 1 (1971), no. 2.
- [19] M. S. PATERSON AND L. G. VALIANT, *Circuit size is nonlinear in depth*, Univ. of Warwick Theory of Computation Rep., vol. 8, Sept. 1975.
- [20] N. PIPPENGER, *Short formulae for symmetric functions*, IBM Res. Rep. RC-5143, Yorktown Heights, NY, 1974.
- [21] ———, *The Complexity of Monotone Boolean Functions*, Math Systems Theory, submitted.
- [22] V. PRATT AND L. STOCKMEYER, *A characterization of the power of vector machines*, J. Comput. System Sci., 12 (1976), pp. 198–221.
- [23] J. SAVAGE, *Computational work and time on finite machines*, J. Assoc. Comput. Mach., 19 (1972), pp. 660–674.
- [24] W. SAVITCH, *Relationships between nondeterministic and deterministic tape complexities*, J. Comput. System Sci., 4 (1970), pp. 177–192.
- [25] C. P. SCHNORR, *The network complexity and the Turing machine complexity of finite functions*, Acta Informat., 7 (1976), pp. 95–107.
- [26] P. M. SPIRA, *On time hardware complexity tradeoffs for Boolean functions*, Fourth Hawaii International Symp. on Systems Science, 1971, pp. 525–527.

- [27] V. STRASSEN, *Vermeidung von Divisionen*, J. Reine Angew. Math., 264 (1973), pp. 184–202.
- [28] G. TOURLAKIS, *A universal parallel machine and the efficient simulation of storage bounded sequential computations*, Dept. of Computer Science and Mathematics Tech. Rep. 1, Atkinson College, York University, Toronto, Canada, Dec. 1976.
- [29] R. LADNER, *The circuit value problem in Logspace complete for P*, SIGACT News, 7 (1975), pp. 18–20.
- [30] R. P. BRENT, *The parallel evaluation of several arithmetic expressions*, J. Assoc. Comput. Mach., 21 (1974), pp. 201–206.