Ruzzo shows the above result still holds when U_E is replaced by U, provided $S \geqslant L^2$.

From their definition, ATM's appear to model a restricted form of parallel computation, because the "processors" in the model are restricted to be Turing machines, and they must be organized in the form of an and-or-tree. This makes Theorem 2.5 all the more interesting. On the other hand, ATM's are more pleasing in one way than circuit families, because there is no question of how to define *uniform*. Each ATM is automatically uniform. In fact, ATM's may be the best candidate proposed so far for defining parallel time, at least in the fixed structure category. But this remains to be seen. The one clear drawback of ATM's is that they do not seem to have any resource that corresponds to hardware size (see section 4).

3. Log Depth vs Log Space

As far as we know, the second inclusions in Theorems 2.2 and 2.4 cannot be improved, even when NSPACE is replaced by DSPACE. (Of course an improvement for NSPACE would improve Savitch's theorem.) Taking $S(n) = \log n$ as the most basic case, it is interesting to look for examples of sets in DSPACE $(\log n)$ which do not appear to be in DEPTH $(\log n)$. Addition of n n-digit binary numbers, and multiplication of two n-digit binary numbers both can be done in $O(\log n)$ circuit depth (see [S3]), as can sorting n n-digit binary numbers (see [MP]). On the other hand, the "cycle free problem" is in DSPACE $(\log n)$ but does not appear to be in DEPTH $(\log n)$.

Definition. The cycle free problem (CFP) is the set of all binary codes for symmetric Boolean $N \times N$ adjacency matrices A of undirected cycle-free graphs.

One can define functions $f: \{0,1\}^* \to \{0,1\}^*$ computable in depth S (or uniform depth S) using circuits with several outputs. We say A_1 is *log depth reducible* to A_2 (respectively *uniformly log depth reducible*) iff there is some function f computable in depth $O(\log n)$ (respectively uniform depth $O(\log n)$) such that $w \in A_1$ iff $f(w) \in A_2$, for all w. We say A is *log depth complete* for the class $\mathcal{S}$ iff $A \in \mathcal{S}$, and every $A' \in \mathcal{S}$ is log depth reducible to A. The uniform case is defined similarly. The main ideas in the proof of the following result appear in Hong [H2].

THEOREM 3.1. (a) CFP *is uniformly log depth complete for* DSPACE ($\log n$).

(b) CFP *is log depth complete for* DSPACE ($\log n$) (NONUNIFORM).

COROLLARY (a) DSPACE ($\log n$) = UDEPTH ($\log n$) iff CFP $\in$ UDEPTH ($\log n$).

(b) DSPACE ($\log n$) (NONUNIFORM) = DEPTH ($\log n$) *iff* CFP $\in$ DEPTH ($\log n$).

We note that because UDEPTH ($\log n$) $\subseteq$ DEPTH ($\log n$), the first equation in the Corollary implies the second. This fact does not seem to be obvious without using the CFP.

To prove CFP $\in$ DSPACE ($\log n$), Hong devised an algorithm for moving several pebbles around the input graph in an attempt to do a depth first search of each of its components. To prove that every

$$A \in \text{DSPACE } (\log n)$$

is uniformly log depth reducible to CFP, one can, given an input w, define a graph whose nodes are $\mathscr{C} \times \{0, 1, ..., T\}$, where $\mathscr{C}$ is the set of possible configurations of the Turing machine M with input w, where M accepts the complement of A in space O ($\log n$), and T is an upper bound on the computation time. Two nodes (c, t) and $c', t')$ are adjacent iff either $c \to c'$ in one step and $t' = t + 1$, or $c' \to c$ and $t = t' + 1$. If we let c_0 be the initial configuration and c_f be the unique accepting configuration, then we also add an edge between $(c_0, 0)$ and (c_f, T). Using the fact that M is deterministic, it is not hard to see that M accepts w iff the graph has a cycle.

A second example for which theorem 3.1 applies is GAP1: the graph reachability problem for directed graphs of outdegree one. The completeness of GAP1 for DSPACE ($\log n$) is proved for reducibilities other than log depth in [J] and in [HIM]. The proof of theorem 3.1 for GAP1 is easier than for CFP.

The following example is interesting, because it is complete for non-uniform $\log n$ space, but no one knows how to solve it in uniform $\log n$ space.

Definition. The undirected graph reachability problem (URP) is the set of codes of symmetric adjacency matrices of graphs with nodes $\{1, 2, ..., N\}$ with a path from node 1 to node N.

THEOREM 3.2. URP *is log depth complete for* DSPACE ($\log n$) (NONUNIFORM).

That URP $\in$ DSPACE ($\log n$) (NONUNIFORM) follows from the existence of a short universal covering string for all n-node undirected connected oriented graphs of fixed degree (see [AKLLR]). The reducibility proof is similar to the above argument.

Many interesting problems have O ($\log^2 n$) as the best known upper bound for both deterministic space and uniform depth. It is interesting to try to reduce these to each other via log depth or uniform log depth reducibility, so as to cut down the number of equivalence classes of problems classified by their depth complexity. For example, the directed graph reachability problem (GRP) is well known to be log space complete for NSPACE ($\log n$) (see [HU]). In fact, it is also *uniform log depth* complete for NSPACE ($\log n$). Two other examples are finding the integer part of the quotient of two n-digit binary numbers, and raising an n-digit number to the power n. The best known upper bound for both problems for both space and depth is O ($\log^2 n$). Hoover [H1] shows that each is log depth reducible to the other, although one of the reductions is not uniform. As a matter of interest, Hoover also points out that the base conversion problem (say converting binary notation to ternary) is in nonuniform depth O ($\log n$) (because the powers of two in ternary can be built in), but the best space upper bound and uniform depth upper bound is O ($\log^2 n$).

4. CONGLOMERATES AND AGGREGATES

Uniform circuits and ATM's are good models for measuring parallel time, but neither is right for measuring the second important resource mentioned in the introduction, namely hardware size. What is needed is to allow circuits to have cycles. Goldschlager's conglomerates [G1] satisfy this requirement. Briefly, a *conglomerate* is an infinite collection $\{M_0, M_1, ...,\}$ of identical deterministic finite state machines connected together in some manner. Each machine has $r \geqslant 1$ inputs and one output, and the connection function f specifies for some inputs of some machines the output of which machine it is connected to. (Inputs left unconnected receive some fixed symbol b.) Cycles are allowed in the connection graph. Initially at time O, the first n machines $M_1, ..., M_n$ store the symbols of the input string $w_1 w_2 ... w_n$, and all other machines start in the initial state q_0. At sub-