CS 6743 **Lecture 16** **Fall 2007**

1 Relating circuit classes with time and space

As we said before, it is possible to solve undecidable problems with even small circuits. So in order to compare circuit classes with time and space we need an additional restriction, which would eliminate the strange situations such as solving the halting problem. This restriction is *uniformity.*

Definition 1. *A circuit family is (e.g., logspace) uniform if there exists a Turing machine in the respective class (such as logspace) that, on input consisting of n 1s, generates the n^{th} circuit in the family.*

Here (and in Sipser's book) when we say "uniform" we will mean logspace-uniform.

In the rest of this lecture we will prove the following:

$$NC_1 \subseteq L \subseteq NL \subseteq NC_2 \subseteq NC \subseteq P$$

Here, we already showed that $L \subseteq NL$ and $NC_2 \subseteq NC$. Now we will outline the proofs of the remaining three inclusions.

Theorem 2. $NC_1 \subseteq L$

Proof. The main idea is that in logspace it is possible to do a kind of depth-first search on a binary tree (or acyclic graph) of logarithmic depth. This is due to the fact that it is not necessary to remember a path in a tree as a list of vertices; instead, the logspace machine remembers the path as a log-length sequence of "left/right" directions. Also, it is not necessary for the machine to remember the value on the left branch of the tree: if it is "true" for an AND gate and "false" for an OR gate, the machine does not bother to follow the right branch.

Therefore, it is possible in logspace to search through an NC_1 circuit and evaluate it. □

Theorem 3. $NL \subseteq NC_2$

Proof. The idea of the proof of this statement resembles the proof of Savich's theorem. Recall that the classical NL-complete problem is graph reachability. So all we need is to show how to solve graph reachability in NC_2.

Recall that boolean matrix multiplication is in NC_1. Also, remember that to compute the transitive closure of a graph it is sufficient to take its adjacency matrix to the n^{th} power (where n is the number of vertices). Since we can power the matrix by repeated squaring (that is $A * A, A^2 * A^2, A^4 * A^4 ...$), powering requires $\log n$ steps. Therefore, an NC_2 circuit graph reachability can work as follows: make $\log n$ levels where each level has NC_1 circuits to compute product of matrices and give the resulting matrix as an input to the following level. The total depth is thus $\log^2 n$. Finally, to check if there is a path from s to t in the graph, look at the value of the cell $[s, t]$ in the final matrix. □

In the last theorem here, we show that in polynomial time it is possible to simulate parallel computation of polynomial size and depth $\log^k$ for any k.

Theorem 4. $NC \subseteq P$

Proof. Since we consider uniform circuits, a polynomial-time algorithm can generate the circuit on its own and the circuit will have polynomial size. Now, all that is left is to evaluate the circuit. This can be done in polynomial time, by a depth-first search on the circuit. □