IITM-CS6840: Advanced Complexity Theory March 9, 2012

Lecture 36 : Non-uniformity of circuits

Lecturer: Jayalal Sarma M.N. *Scribe: Prashant Vasudevan*

We start with a mention of how to associate decision problems with circuits and other definitions.

Definition 1. A language L is said to be computed by a family of circuits $\{C_n\}_{n>0}$ if:

$$x \in L \Leftrightarrow C_{|x|}(x) = 1$$

Definition 2. $SIZE(f(n)) \triangleq \{L \mid L \text{ is computed by a family of size } f(n)\}$

Definition 3. $\mathsf{PSIZE} = \bigcup_{k\geq 0} SIZE(n^k)$

PSIZE has certain sociable properties like closure under composition, much like P.

Now we bring in a class introduced while discussing BPP.

Definition 4. P/poly is the class of languages that can be computed in deterministic polynomial time given access to a polynomial size advice function, i.e., some $h : \mathbb{N} \to \{0,1\}^*$ such that there is a polynomial p such that $\forall n \in \mathbb{N}\ |h(n)| \leq p(n)$.

(Note that the definition says nothing more about h - it may even be uncomputable.)

For any $L \in \mathsf{P/poly}$ we can say there exist a family of poly-time TMs $\{M_n\}_{n\geq 0}$ such that M_n has $h(n)$ built into it and can thus decide membership of strings of length n in L. We show below that this is more than a passing resemblance to the way circuit families are used to define languages.

Theorem 5. $\mathsf{PSIZE} = \mathsf{P/poly}$

Corollary 6. *Any $L \in \mathsf{P}$ has polynomial-size circuits.*

Hence to show that some language is not in P, it suffices to show that it does not have polynomial-size circuits. This is indeed a bit of overkill as doing so would show that the language is not in P/poly itself which is a much bigger class, but still worth exploring as the circuit model permits various algebraic and graph theoretic approaches that were absent in the Turing Machine model.

Questions:

1. Is NP in P/poly?

- No $\Rightarrow$ $\mathsf{P} = \mathsf{NP}$.
- Yes $\Rightarrow$ $\mathsf{PH} \subseteq \Sigma_2^p$, by the Karp-Lipton Theorem.

2. Is the Halting Problem in $\mathsf{P/poly}$?

The Circuit Value Problem (CVP) is defined as follows:

Definition 7. $\mathsf{CVP} = \{(C, x) \mid C(x) = 1\}$, where C is a circuit built on a basis of 2-input gates, represented in the input as a directed graph along with a mapping of vertices to inputs, gates and outputs.

Clearly, $\mathsf{CVP} \in \mathsf{P}$, as the evaluation of the circuit may be done bottom-up starting from the input. This makes proving one side of above theorem quite easy. To see that $\mathsf{PSIZE} \subseteq \mathsf{P/poly}$, simply use the the advice $h(n) = C_n$ for a language defined by a circuit family $\{C_n\}$. Given this circuit, a machine can evaluate it on the input in polynomial time and decide whether it is indeed in the language.

Claim 8. CVP *is* P*-Complete under many-one* $\mathsf{LOGSPACE}$ *reductions.*

($\mathsf{LOGSPACE}$ reductions because poly-time reductions don't really make sense in P.)

Proof. For the reduction, we shall give a circuit family that performs the same computations as the TM for the given language and show that computing it can be done in $\mathsf{LOGSPACE}$.

Given a language $L \in \mathsf{P}$ over $\{0,1\}^*$, let M be the corresponding TM that runs in $p(n)$ time. Without loss of generality, assume that M starts with its head on the leftmost cell of the tape and on halting erases the tape before accepting or rejecting. We have a

The circuit C_n has n inputs (for the bits of the input string) and $p(n)$ levels (one for each step taken by M). At each level, the circuit has two wires for each of the $p(n)$ cells on the tape that M could possibly use in its $p(n)$ steps, one for the symbol, and one to indicate whether the head is at that position, and a set of wires that store the machine's state. As the state of the wires for any of the cells depends only on the wires in the previous level and that too only on those corresponding to the same cell and the two adjacent cells and the state according to the rules of M. The wires for the state depend on all the wires of the previous level.

It can be seen now that all these connections may be written down on a separate input tape without the need for much space for computation on the work tape - hardly anything other than a counter indicating which level the reduction machine is currently working on and which gate in that level, as these are needed to specify the vertex in the graph. The state wires need $O(\log p(n))$ gates and the others only a constant number of gates. There are $O(p(n)$ wires on each level and $O(p(n)$ levels, hence there are only $O(p(n)^2)$ gates and wires in the entire circuit, and the indices are only $O(\log n)$ bits long. Hence the reduction can be performed in $\mathsf{LOGSPACE}$. $\square$