Fifth Week

Instructor: Prof. S. P. Pal
TA: Rahul Gokhale

We saw how it was easy to simulate a boolean circuit over its input bits in time proportional to the circuit description length. The $DSPACE(\log_2 n)$ reduction from an arbitrary decision problem in the class P to this *circuit value problem (CVP)* is instructive in how a circuit model can simulate a TM. However, note that the circuit required for deciding input strings of size n for the the decision problem depends on n and can be created as above in deterministic logspace. We also saw how we can cascade such reductions. It is possible to show that *context-free emptiness* is also complete for the class P in the above sense as much as CVP is. It is easy to show that these reductions run in polynomial time. So, why do we not replace the logspace reductions by polynomial time reductions? A few important reductions are in Examples 8.1, 8.2, 8.3 and 8.4.

The NP-completeness of SAT is easily established using a nondeterministic generalization of the P-completeness proof of CVP. The deterministic logspace reduction in this case starts with valid inputs for the language L accepted by the NDTM M_L, accepting the arbitrary language $L \in NP$. The generated input is a valid input for the CIRSAT problem, which is in turn reducible to SAT. Since we can cascade such reductions (non-trivial cascading discussed in class), we can say that SAT is complete for the class NP as much as $CIRSAT$ is.

We can also study problems complete for other classes such as $NL = NSPACE(log_2 n)$ and $PSPACE$. It is quite easy to simulate language recognition for the class NL by invoking the $REACHABILITY$ question. Work out the details.

With some more effort, we can establish an altogether different kind of result that $2SAT \in P$. In fact, we can show that $2SAT$ is in NL, by first showing that $2UNSAT$ is in NL (why?). Whereas, we know that $NL \subseteq P$.

Finally, leaving aside the more complex proof of the P-completeness of the *context-free emptiness problem*, we conclude here with the $PSPACE$-completeness of the it context-sensitive recognition problem. It is necessary to show that this problem is in $PSPACE$ in the first place; then, we use *padding* to show that any language L $PSPACE$ can be reduced (using a suitable (deterministic logspace?) reduction), to another language L' in $DSPACE(n)$ (a subset of CSLs). If G is the CSG for L' and x any encoding of G, then we can define a polynomial method to convert $y \in L$ to $x|y\$^{p(y)}$ [$p()$ is a polynomial], so that this method is indeed a reduction from L to L_{CS} (*context-sensitive recognition*).