CSC2401 Introduction to Complexity, Fall 01-02
Lecture 7, Thursday October 5th 2001
Lecturer: Allan Borodin
Scribe: bowen hui

1 Review from Lecture 6

The present view of complexity classes is:

$$\mathcal{DL} \subseteq \mathcal{NL} \subseteq \mathcal{P} \subseteq \mathcal{NP} \subseteq PSAPCE \subseteq \bigcup 2^{n^k}$$

We also have: $L_1 \leq_{logspace} L_2$ and $L_1 \leq_p L_2$. The corresponding notations in the text are: $L_1 \leq_m^{log} L_2$ and $L_1 \leq_m^p L_2$ respectively, where, $L_1 \leq_m L_2$ if $\exists$ a computable f s.t. $w \in L_1$ iff $f(w) \in L_2$. Note that (i) $L_1 \leq_{logspace} L_2$ implies $L_1 \leq_p L_2$ and (ii) $\leq_{logspace}$ is transitive!

2 Logspace is Transitive

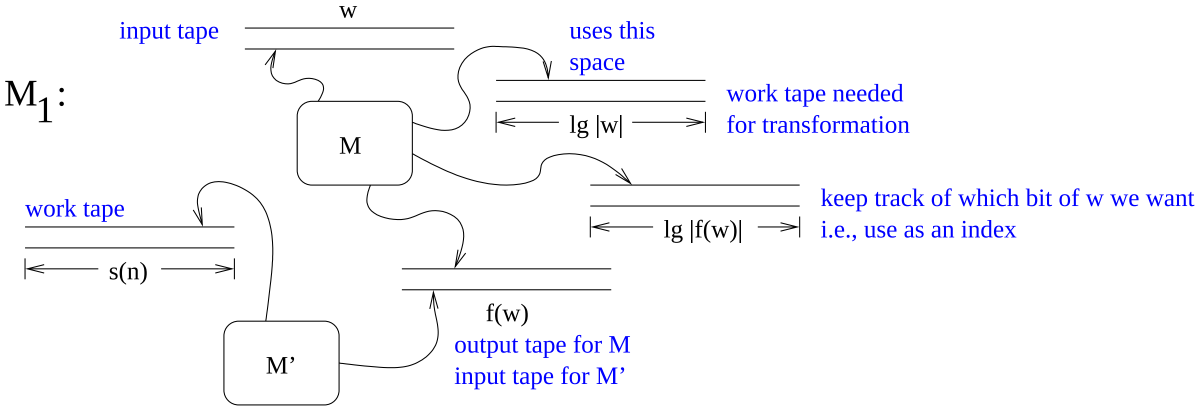
Fact: Let $\mathcal{C}$ be a class of languages and $s(n) \geq lg(n)$. If L is $\leq_{logspace}$ complete for $\mathcal{C}$ then:

$L \in DSPACE(s(n))$ iff $\mathcal{C} \subseteq DSPACE(s(n))$.

Proof:
($\Leftarrow$ part) Just use the fact that $L \in \mathcal{C}$. (Recall that "completeness" has two parts: (i) $L \in \mathcal{C}$ and (ii) $\forall L_1 \in \mathcal{C}$, $L_1 \leq_{logspace} L$.)

($\Rightarrow$ part) Just use "hardness". Let $L_1 \in \mathcal{C}$. Show $L_1 \in DSPACE(s(n))$.

Let M_1 be the TM we are constructing for L_1. L is computed by some $s(n)$ space bounded M' and $L_1 \leq_{logspace} L$ via f computed by logspace M. The machine M_1 uses M and M', as illustrated below.



Problem: What if $f(w)$ is bigger than polynomial? Note that $|f(w)| \leq c^{log(n)}$. So, $log|f(w)| \leq lg(n)$. So we can say that $\leq_{logspace}$ is transitive.

Why do we care? Cf. $STCON$ proof in Section 3. Also, later we will show a problem called CVP (Circuit Value Problem) that is $\leq_{logspace}$ complete for P.

3 Cool Use of Recursion in Savitch's Theorem

Recall that $STCON$ is an encoding of a directed graph (as referred to $GAcc$ in the text):

Definition: $STCON = \{(G, u, v) | \exists$ directed path in G from u to $v\}$

$STCON$ is $\leq_{logspace}$ complete for NL. In terms of languages, we took M and w, and produced $G_{M,w}$ and saw that $w \in \mathcal{L}(M)$ iff $\exists$ path from initial configuration to the final configuration (without loss of generalization, we assumed there is a unique accepting configuration).

Corollary: $\mathcal{DL} = \mathcal{NL}$ iff $STCON \in \mathcal{DL}$

More generally, $\mathcal{NL} \subseteq DSPACE(s(n))$ iff $STCON \in DSPACE(s(n))$. Now, we show Savitch's theorem: $STCON \in DSPACE(lg^2 n)$. This implies that $NL \subseteq DSPACE(lg^2 n)$. Before we prove Savitch's theorem, here is another corollary:

Corollary: Given $s(n) \geq lg(n)$ is constructible, $NSPACE(s(n)) \subseteq DSPACE(s(n)^2)$

(This can be proven by padding.) In general, to get from a nondeterministic space class to a determinist space class, the complexity needs to be squared. Now the proof.

Proof:
Define $reach_G(u, v, i)$ as follows:

$\exists$ directed path from u to v, with length $\leq i$.

Note: $STCON(G, u, v) \equiv reach_G(u, v, n-1)$ where $n = |V|$, for $G(V, E)$.
Since every path has a midpoint, $reach_G$ can be rewritten as follows:

$$reach_G(u, v, i) \equiv \exists i : reach_G(u, z, \lfloor \tfrac{i}{2} \rfloor) \text{ and } reach_G(z, v, \lceil \tfrac{i}{2} \rceil).$$

Also,

$reach_G(u, v, 0)$ (which means $u = v$).

Now, one can write a recursive function to solve the problem and use a stack to implement the recursion. Since each entry of the stack uses $lg(n)$ bits and the height of the stack is $lg(n)$, clearly the algorithm is in $DSPACE(log(n^2))$.

Savitch's theorem gives some insight into the $\mathcal{DL}$ versus $\mathcal{NL}$ problem; and to some, this result may suggest that $\mathcal{DL} = \mathcal{NL}$.

4 Sneak Preview of Lecture 8

Definition: Let $\mathcal{C}$ be a class of languages. Then $co - \mathcal{C} = \{L | \overline{L} \in \mathcal{C}\}$, where $\overline{L}$ is the complement of L (i.e., $\overline{L} = \Sigma^* - L = \{w | w \notin L\}$).
Consider $co - \mathcal{NL}$.

Fact: $co - \mathcal{NL} = \mathcal{NL}$ iff $\overline{STCON} \in \mathcal{NL}$.

Where the definition of $\overline{STCON}$ is $\overline{STCON} = \{(G, u, v) | \nexists$ path from u to $v\}$. To prove this fact, one can use the log space completeness of $STCON$.

Immermann-Szelepcse'nyi Theorem: $\mathcal{NL} = co - \mathcal{NL}$.

Corollary: $NSPACE(s(n)) = co - NSPACE(s(n))$, for $s(n) \geq lg(n)$.

Proof: Cf. next lecture!