Chapter 6

Deterministic and Nondeterministic Logarithmic Space

Let

$$\begin{aligned}
\mathcal{L} &= \text{DSPACE}(\log n), \\
\mathcal{NL} &= \text{NSPACE}(\log n), \\
\mathcal{P} &= \bigcup_{c>0} \text{DTIME}(n^c) \;=\; \text{ASPACE}(\log n), \text{ and} \\
\mathcal{NP} &= \bigcup_{c>0} \text{NTIME}(n^c).
\end{aligned}$$

From Theorem 4.1, $\mathcal{L} \subseteq \mathcal{NL} \subseteq \mathcal{P} \subseteq \mathcal{NP}$. It is an open question (part of Open Problem 4.3 on page 20) whether any of these containments is proper; it is even possible that $\mathcal{L} = \mathcal{NP}$. We concentrate now on the first of these containments:

Open Problem 6.1: Does $\mathcal{L} = \mathcal{NL}$?

As motivation, we know from Savitch's Theorem (Corollary 4.15), that $\mathcal{L} \subseteq \mathcal{NL} \subseteq$ DSPACE$(\log^2 n)$, so the classes $\mathcal{L}$ and $\mathcal{NL}$ are "very close".

Proposition 6.2: $\mathcal{L}$ is closed under $\leq_m^{\mathcal{L}}$.

Proof: Suppose $A \leq_m^{\mathcal{L}} B$ and $B \in \mathcal{L}$. It is an easy exercise to prove that $B \in \mathcal{L}$ if and only if $B \leq_m^{\mathcal{L}} \{1\}$. By Lemma 5.3, $A \leq_m^{\mathcal{L}} \{1\}$, so $A \in \mathcal{L}$. □

The next proposition shows that concentrating on $O(\log n)$ space may have ramifications to higher space bounds.

Proposition 6.3: If $\mathcal{L} = \mathcal{NL}$, then DSPACE$(S(n))$ = NSPACE$(S(n))$ for all space constructible $S(n) \geq \log_2 n$.

Proof: Let L be accepted by a nondeterministic Turing machine N in space $S(n)$. Consider $L' = \{x10^{2^{S(n)}-n-1} \mid x \in L \text{ and } |x| = n\}$.

$L' \in \mathcal{NL}$, as follows: Check that the input is of the form $y = x10^*$, where $|y| = 2^{S(|x|)}$, and then simulate N on x. The space needed to compute $S(|x|)$, measure $|y|$, and simulate N is all $S(|x|) = \log_2 |y|$.

Since $\mathcal{L} = \mathcal{NL}$, L' is accepted by a deterministic Turing machine D in space $\log_2 n$.

Then $L \in \text{DSPACE}(S(n))$, as follows: On input x, simulate D on $y = x10^{2^{S(n)}-n-1}$. If D's input head tries to leave the right end of x, use a worktape to keep track of its position within y. The space needed is $S(n)$ to compute $S(|x|)$ and keep track of D's input head position, along with another $\log_2 \left(2^{S(n)}\right) = S(n)$ space to simulate D. □

The method of proof in Proposition 6.3 is called "padding", since the strings in L are padded with useless symbols to obtain L'.

6.1. Directed Graph Connectivity

Proposition 5.7 shows how to get more hard problems once you have one. To get the first hard language B for a class, you have to do a generic reduction, that is, reduce a generic language in the class to B. We now present a problem that is $\leq_m^{\mathcal{L}}$–complete for $\mathcal{NL}$.

Let $STCON = \{(G, s, t) \mid G = (V, E)$ is (the encoding of) a directed graph that has a path from vertex s to vertex $t\}$.

Convention 6.4: We will assume that all graphs are encoded as follows: the elements of V are encoded as the binary representations of (not necessarily consecutive) integers, each of length $O(\log |V|)$, and E is encoded as a list of ordered pairs of these binary representations.

Theorem 6.5 (Savitch [41]): *$STCON$ is $\leq_m^{\mathcal{L}}$–complete for $\mathcal{NL}$.*

Proof: The proof has two parts: we must show that $STCON \in \mathcal{NL}$, and that $STCON$ is $\leq_m^{\mathcal{L}}$–hard for $\mathcal{NL}$.

1. $STCON \in \mathcal{NL}$: Start at s. In general, if the current vertex is u, nondeterministically choose some v such that $(u, v) \in E$, and replace u by v. Accept if and only if the vertex t is reached. The space to record u and v is $O(\log n)$.

2. $STCON$ is $\leq_m^{\mathcal{L}}$–hard for $\mathcal{NL}$: Let $A \in \mathcal{NL}$ be an arbitrary language accepted by some nondeterministic Turing machine N in space $O(\log n)$. Construct a deterministic Turing machine D that reduces A to $STCON$, as follows.

CONSTRUCTION: Given x, D outputs (G, s, t), where

$$\begin{aligned} t &\text{ is not a configuration of } N \text{ on } x, \\ G &= (V, E), \\ V &= \{P \mid P \text{ is a configuration of } N \text{ on } x\} \cup \{t\}, \end{aligned}$$

$$E = \{(P,Q) \mid P \vdash_{N,x} Q\} \cup \{(P,t) \mid P \text{ is a final configuration}\}, \text{ and}$$
$$s \text{ is the initial configuration of } N \text{ on } x.$$

CORRECTNESS: Let $\vdash^*_{N,x}$ be the reflexive, transitive closure of $\vdash_{N,x}$. Then

$$\begin{aligned} x \in A \quad & \text{if and only if} \quad N \text{ accepts } x \\ & \text{if and only if} \quad s \vdash^*_{N,x} P, \text{ for some final configuration } P \\ & \text{if and only if} \quad \text{there is a path from } s \text{ to } t \text{ in } G \\ & \text{if and only if} \quad (G,s,t) \in STCON. \end{aligned}$$

ANALYSIS: D can cycle through the configurations P of N in $O(\log n)$ space, since N runs in space $O(\log n)$. For each such P, D can output the edges $\{(P,Q) \mid P \vdash_{N,x} Q\}$ in $O(\log n)$ space. Note that D produces output with size polynomial in n, but this output is not counted toward the space bound. □

Corollary 6.6: *$STCON \in \mathcal{L}$ if and only if $\mathcal{L} = \mathcal{NL}$.*

Proof: This follows from Propositions 5.6 and 6.2, and Theorem 6.5. □

Thus, Open Problem 6.1 is equivalent to resolving whether the particular language $STCON$ is in $\mathcal{L}$.

Corollary 6.7: If $STCON \in \mathcal{L}$, then DSPACE$(S(n))$ = NSPACE$(S(n))$ for all space constructible $S(n) \geq \log_2 n$.

Proof: This follows from Proposition 6.3 and Corollary 6.6. □

6.2. Undirected Graph Problems

6.2.1. Directed vs. Undirected Connectivity

Exercise 6.8: Let $\overline{STCON} = \{(G,s,t) \mid G \text{ has no path from } s \text{ to } t\}$. Show that $\overline{STCON}$ is $\leq^{\mathcal{L}}_m$–complete for $\mathcal{NL}$.

Now consider the undirected version of $STCON$, namely

$$USTCON = \{(G,s,t) \mid G \text{ is an undirected graph with a path from } s \text{ to } t\}.$$

Then $USTCON \in \mathcal{NL}$.

Open Problem 6.9: Is $USTCON \in \mathcal{L}$? Alternatively, is $USTCON$ $\leq_m^{\mathcal{L}}$–complete for $\mathcal{NL}$?

The remainder of this section suggests evidence that $USTCON$ is not as hard as $STCON$, and hence is not $\leq_m^{\mathcal{L}}$–complete for $\mathcal{NL}$.

Definition 6.10: For any $k \geq 1$, $\mathcal{SC}^k = \{L \mid L$ is accepted by some deterministic Turing machine in $O(\log^k n)$ space and polynomial time, simultaneously$\}$.

Note that $\mathcal{SC}^1 = \mathcal{L}$, and $\mathcal{SC}^k \subseteq \mathcal{P} \cap \mathrm{DSPACE}(\log^k n)$, with equality unlikely in the latter.

Definition 6.11: $\mathcal{SC} = \bigcup_{k \geq 1} \mathcal{SC}^k$.

Theorem 6.12 (Nisan [32]): $USTCON \in \mathcal{SC}^2$.

Open Problem 6.13: Is $STCON \in \mathcal{SC}$? The conjecture is "no", and some (relatively weak) evidence is provided by Tompa [47]. If the answer is indeed "no", then $STCON \not\leq_m^{\mathcal{L}} USTCON$, since $\mathcal{SC}^k$ is closed under $\leq_m^{\mathcal{L}}$ for $k \geq 1$.

6.2.2. Shortest Paths in Undirected Graphs

Although $USTCON$ seems unlikely to be $\leq_m^{\mathcal{L}}$–complete for $\mathcal{NL}$, the problem of finding shortest paths in undirected graphs is. More specifically, let

$$USP = \{(G, s, t, k) \mid G \text{ is an undirected graph with a path of length at most } k \text{ from } s \text{ to } t\ \}.$$

Convention 6.14: We assume all integers to be encoded in binary, unless otherwise specified.

Theorem 6.15: USP is $\leq_m^{\mathcal{L}}$–complete for $\mathcal{NL}$.

Proof:

1. $USP \in \mathcal{NL}$: This is as in the proof of Theorem 6.5, but also count the length of the path traversed from s, and reject if this exceeds $\min(n, k)$. The reason for this "min" is that, if the encoding of k has $\omega(\log n)$ bits, there will not be enough space for the counter. However, a shortest path never passes through any vertex more than once. Hence, $|V| < n$ is an upper limit on the length of the path.

2. USP is $\leq_m^{\mathcal{L}}$–hard for $\mathcal{NL}$: We need not do another generic reduction, since we already have a problem $STCON$ that is $\leq_m^{\mathcal{L}}$–hard for $\mathcal{NL}$. Hence, by Proposition 5.7, it suffices to show that $STCON \leq_m^{\mathcal{L}} USP$.

CONSTRUCTION: On input (G, s, t), where $G = (V, E)$, we will make $|V|$ copies of V, with edges corresponding to E between consecutive copies. More precisely, output (G', s', t', k), where

$$\begin{aligned} G' &= (V', E'), \\ V' &= \{v_{i,j} \mid i \in V, 1 \leq j \leq |V|\}, \end{aligned}$$

$$
\begin{aligned}
E' &= \{\{v_{i,j}, v_{k,j+1}\} \mid (i,k) \in E, 1 \le j \le |V|-1\} \cup \{\{v_{t,j}, v_{t,j+1}\} \mid 1 \le j \le |V|-1\},\\
s' &= v_{s,1},\\
t' &= v_{t,|V|}, \text{ and}\\
k &= |V|-1.
\end{aligned}
$$

CORRECTNESS: We need to show that $(G,s,t) \in STCON$ if and only if $(G',s',t',k) \in USP$.

"Only if" clause: Assume that $(G,s,t) \in STCON$. Then there is an $l \le |V|$ such that G has a path of length exactly $l-1$ from s to t. Then G' has a path of length $l-1$ from $v_{s,1}$ to $v_{t,l}$, and hence a path of length $|V|-1$ from $v_{s,1}$ to $v_{t,|V|}$. Thus, $(G',s',t',k) \in USP$.

"If" clause: Assume that $(G',s',t',k) \in USP$. Then there is an $l \le |V|$ such that G' has a path of length exactly $l-1$ from $v_{s,1}$ to $v_{t,l}$ that does not pass through $v_{t,j}$ for any $j < l$. Because the length $l-1$ of this path equals the difference of the second subscripts, and because it does not use any of the edges $\{v_{t,j}, v_{t,j+1}\}$, it corresponds to a path in G from s to t. Thus, $(G,s,t) \in STCON$.

ANALYSIS: To construct G', we need enough space to store the index j so that, for each input $i \in V$, we can output $|V|$ copies $v_{i,j}$, and for each input $(i,k) \in E$, we can output $|V|-1$ copies $\{v_{i,j}, v_{k,j+1}\}$. Since n is an upper limit on $|V|$, at most $\log n$ bits are needed to store j. □

For additional problems $\le_m^{\mathcal{L}}$–complete for $\mathcal{NL}$, see Jones, Lien, and Laaser [21].

6.3. Exercises

1. For any language B, prove that $B \in \mathcal{L}$ if and only if $B \leq_m^{\mathcal{L}} \{1\}$.

2. Let the "formula value problem" be defined as $FV = \{(F, A) \mid F$ is a propositional formula with some number k of variables, and A is a satisfying truth assignment for F, that is, A makes F true$\}$. Show that $FV \in \mathcal{L}$.

3. Let $CYCLE$ be the set of directed graphs that contain a cycle. Prove that $CYCLE$ is $\leq_m^{\mathcal{L}}$–complete for $\mathcal{NL}$.

4. A directed graph G is said to be *k-connected* if and only if, for every set U of $k-1$ vertices and for every pair (v, w) of vertices not in U, there is a path from v to w that does not contain any vertex in U. Let $CONNECTIVITY = \{(G, k) \mid G$ is k-connected$\}$. Prove that $CONNECTIVITY$ is $\leq_m^{\mathcal{L}}$–complete for $\mathcal{NL}$.

   (Hint: Use Exercise 4.27.)

5. Let *2UNSAT* be the set of unsatisfiable propositional formulas in conjunctive normal form with at most two literals per clause. Prove that *2UNSAT* is $\leq_m^{\mathcal{L}}$–complete for $\mathcal{NL}$. What is the complexity of *2SAT*, the set of satisfiable formulas in conjunctive normal form with at most two literals per clause? (I.e., is it in $\mathcal{NL}$? Is it $\leq_m^{\mathcal{L}}$–hard for $\mathcal{NL}$?)

6. Do Exercise 6.8.

7. Prove that $\mathcal{SC}$ is closed under $\leq_m^{\mathcal{L}}$.