The Immerman-Szelepcsényi Theorem: $NL = coNL$

This proof was presented in class on Thursday, Feb 5, 2015. Throughout, points that you are encouraged to think through and justify in detail are marked by "(WHY?)."

Recall first that PATH is NL-complete and, equivalently, that PATH is coNL-complete. Thus, it suffices to show that PATH, the set of triples $(G, s, t)$ in which $G$ is a directed graph that does not contain a path from $s$ to $t$, is in NL.

We will do so by exhibiting a deterministic, logspace verifier that takes as input both an instance $(G, s, t)$ and a certificate of this instance’s membership in PATH. As usual, the tape on which the instance is written is read-only. New to our discussion of nondeterministic logspace is the requirement that the tape on which the certificate is written is not just read-only but read-once, left-to-right. The work/output tapes of this machine are, as usual, read/write, and they are the only tapes that are restricted to logspace.

If $V(G) = \{1, 2, \ldots, n\}$, and $G$ is encoded on the input tape as an $n \times n$ adjacency matrix, then the input is of length $O(n^2)$. Thus, we need certificates of length $poly(n^2) = poly(n)$ and space complexity $O(\log(O(n^2))) = O(\log n)$.

Let $C_i = \{v \in V(G) \mid v \text{ is reachable from } s \text{ by a path of length at most } i\}$. Note that $C_0 = \{s\}$ and that $C_n$ contains all nodes in $G$ that are reachable from $s$ by any path whatsoever. (WHY?) The desired certificate that $(G, s, t)$ is in PATH must therefore certify the fact that $t \notin C_n$. It comprises three types of “subcertificates,” as follows.

- $\text{CERT}_1(v, i, q_i)$ proves that $v \not\in C_i$, given that $|C_i| = q_i$.
- $\text{CERT}_2(v, i, q_i-1)$ proves that $v \not\in C_i$, given that $|C_{i-1}| = q_{i-1}$.
- $\text{CERT}_3(i, q_i, q_{i-1})$ proves that $|C_i| = q_i$, given that $|C_{i-1}| = q_{i-1}$.

Overall, to prove that $(G, s, t) \notin \text{PATH}$, we use the certificate

$$\text{CERT}_3(1, q_1, 1)\text{CERT}_3(2, q_2, q_1)\cdots\text{CERT}_3(n, q_n, q_{n-1})\text{CERT}_1(t, n, q_n).$$

That is, starting with the obvious fact that $|C_0| = 1$, the logspace verifier first checks, for each successive $i$, $2 \leq i \leq n$, that $|C_i| = q_i$; once it has checked that $|C_n| = q_n$, it checks that $t$ is not one of the $q_n$ nodes in $C_n$. If each of the constituent subcertificates is polynomial-length and logspace verifiable in a read-once, left-to-right manner, then so is the entire certificate. (WHY?)

$\text{CERT}_1(v, i, q_i)$ is a list of $q_i$ paths to all of the nodes reachable from $s$ along paths of length at most $i$. If we denote by $\ell(1), \ldots, \ell(q_i)$ the lengths of these paths, then this subcertificate has the form:

$$\langle u_1^1 u_2^1 \ldots u_{\ell(1)}^1 \rangle \langle u_1^2 u_2^2 \ldots u_{\ell(2)}^2 \rangle \cdots \langle u_1^{q_i} u_2^{q_i} \ldots u_{\ell(q_i)}^{q_i} \rangle,$$

where $u_j^i = s$, for $1 \leq j \leq q_i$, and $u_{\ell(1)}^1 < u_{\ell(2)}^2 < \cdots < u_{\ell(q_i)}^{q_i}$. That is, all of the paths start at $s$, and we list them in increasing order of the labels of their terminal vertices.

It suffices for the deterministic, logspace verifier to check the following. (WHY?)

- The total number of paths in the subcertificate is $q_i$.  

1
• $v$ is not in any of the paths.
• For $1 \leq j \leq q_i - 1$, $u_{\ell(j)}^i < u_{\ell(j+1)}^i$.
• The arcs $u_k^j \rightarrow u_{k+1}^j$ are all in $E(G)$.
• $s = u_1^i$, for $1 \leq j \leq q_i$.
• $\ell(j) \leq i$, for $1 \leq j \leq q_i$.

Indeed, all of these conditions can be verified in deterministic logspace in a read-once, left-to-right manner. (WHY?)

$\text{CERT}_2(v, i, q_i - 1)$ is the same as $\text{CERT}_1(v, i - 1, q_i - 1)$, but its verification procedure contains one more condition: The arc $u_{\ell(j)}^i \rightarrow v$ is not in $E(G)$ for any $j$, $1 \leq j \leq q_i - 1$. That is, if one knows all of the nodes that can be reached by paths of length at most $i - 1$, checking that $v$ is not reachable in one step from any of them suffices to show that $v$ cannot be reached by a path of length at most $i$.

Finally, $\text{CERT}_3(i, q_i, q_i - 1)$ consists of $n$ subcertificates $D_1, \ldots, D_n$. If $j \in C_i$, then $D_j$ is a path from $s$ to $j$ of length at most $i$. If $j \notin C_i$, then $D_j = \text{CERT}_2(j, i, q_i - 1)$. The verifier reads them all left-to-right, checks their validity, and checks that exactly $q_i$ of them certify that the vertex $j$ in question is a member of $C_i$.

Note that, in addition to being verifiable in a read-once, logspace manner,

$$\text{CERT}_3(1, q_1, 1) \text{CERT}_3(2, q_2, q_1) \cdots \text{CERT}_3(n, q_n, q_{n-1}) \text{CERT}_1(t, n, q_n)$$

is indeed polynomial-length. (WHY?)