6. The Immerman–Szelepscényi Theorem

Synopsis. NL $=$ *co*-NL

We know that the s-t connectivity problem on directed graphs DSTCON is complete for NL under logspace reductions. Thus it is an archetypal decision problem in NL—indeed we used the completeness of DSTCON for NL as an expository convenience when we proved NL $\subseteq$ P and NL $\subseteq$ DSPACE$[\log^2 n]$. We now turn our attention to the complement of DSTCON, the decision problem whose "yes" instances are triples $\langle G, s, t\rangle$ where s and t are vertices in G and there is *no* path from s to t in G.

Naturally, $\overline{\text{DSTCON}}$ is complete for *co*-NL, similar to the fact that $\overline{\text{SAT}}$ is complete for *co*-NP. We mentioned that there is no polynomially-bounded proof system with polynomial time verification known for $\overline{\text{SAT}}$, and indeed, it is widely believed that no such proof system exists, that is, NP $\neq$ *co*-NP. It is tempting to guess that similar difficulties will arise in designing a proof system for $\overline{\text{DSTCON}}$ that admits a logspace verifier with one-way access to the proof. Surprisingly, however, such a proof system is possible, and is rather simple to describe.

Before we describe this result, note that a proof system for $\overline{\text{DSTCON}}$ follows from the fact that $\overline{\text{DSTCON}} \in$ P, namely the trace of execution of, say, the breadth-first search algorithm on a graph serves as a proof of connectivity or of non-connectivity. However, a moment's reflection indicates that it is not obvious how, given an alleged trace of BFS on a graph, its correctness can be verified with one-way access to the trace and using only logarithmic space. The proof system that we will present may be thought of a "more detailed" trace of BFS on the graph G starting at vertex s.

An important auxiliary step in the verifier for this proof system will be the verifier from the proof system for DSTCON, slightly modified to verify that a vertex is reachable in a certain number of steps. For completeness, we describe this as a sub-routine.

CHECK-PATH$(G, s, v, i, \pi = \langle \ell, w_0, \ldots, w_\ell \rangle)$
(returns PASS *iff π is a proof that v is reachable from s via $\leq i$ edges)*
if $w_0 \neq s$ return FAIL
if $\ell > i$ return FAIL
$x \leftarrow w_0$
$k \leftarrow 1$
while $k \leq \ell$ do:
    $y \leftarrow w_k$
    if $(x, y) \notin E(G)$ return FAIL
    $x \leftarrow y$
    $k \leftarrow k + 1$
end-while
if $x \neq v$ return FAIL
return PASS

Note the following properties of CHECK-PATH: aside from the parameters s, v, and i, and the inputs G and π, CHECK-PATH uses only three variables, which is $O(\log n)$ space, where $n = |V(G)|$. Furthermore, CHECK-PATH can process π in a one-way fashion (it uses two-way access to G, though).

We are now ready to describe the ideas used in the proof system for $\overline{\text{DSTCON}}$. First we describe some basic notation. Let $n = |V(G)|$, and for integer i, $0 \leq i < n$, define R_i to be the set of vertices reachable from s in G by a path that contains no more than i edges, and let $r_i = |R_i|$. The heart of the proof system for $\overline{\text{DSTCON}}$ is the following component:

Given r_i, proving (to a one-way logspace verifier) that a vertex $u \notin R_{i+1}$

Let's see how this component can be used to build up a proof system for $\overline{\text{DSTCON}}$. Assume that there is no path from s to t in G. As a starting point, note that $r_0 = 1$ since $R_0 = \{s\}$. Secondly—and crucially—by repeated applications of this component, we will show how the proof system will enable the verifier to compute r_{i+1} from r_i, for $0 \leq i < n-1$. Finally, note that once the verifier has r_{n-1} and the proof consists of another application of this component with $i = n-1$ and $u = t$, we're done.

Now for the details of the main component. Suppose we (the verifier) are promised that the correct value of r_i is ρ, and we wish to verify the claim that $u \notin R_{i+1}$. As proof, we are given a list of ρ tuples $\langle v_1, \pi_1 \rangle, \ldots, \langle v_\rho, \pi_\rho \rangle$, where:

(1a) $v_1 < v_i < \ldots < v_\rho$ are distinct vertices;
(1b) for $1 \leq j \leq \rho$, π_j is a legal path of length $\leq i$ from s to v_j, that is, CHECK-PATH(G, s, v_j, i, π_j) accepts;
(1c) $(v_j, u) \notin E(G)$ for $1 \leq j \leq \rho$.

Then it follows that u is not reachable from s in at most $i+1$ steps—after all, we know how many vertices are reachable from s in i steps, which vertices they are (and why they are reachable), and that there is no edge from any of these vertices to u. The crucial point is that with the proof format described above, the verification of this step can be done in logspace with one-way access to the proof $\langle v_1, \pi_1 \rangle, \ldots, \langle v_\rho, \pi_\rho \rangle$. This follows from the fact that CHECK-PATH is a one-way logspace procedure and the condition $v_1 < \ldots < v_\rho$ can also be verified "on the fly."

CHECK-UNREACHABILITY$(G, s, u, i, \sigma = \langle \rho, \langle v_1, \pi_1 \rangle, \ldots, \langle v_\rho, \pi_\rho \rangle \rangle)$
(returns PASS *iff σ is a proof that $u \notin R_{i+1}$)*
$j \leftarrow 1$
$v \leftarrow 0$ *(assume lowest vertex number is 1)*
while $j \leq \rho$ do:
  if $v_j \leq v$ return FAIL
  $v \leftarrow v_j$
  if CHECK-PATH$(G, s, v, i, \pi_j) =$ FAIL return FAIL
  if $(v, u) \in E(G)$ return FAIL
end-while
return PASS

Next we turn to the proof that guides the verifier from r_i to r_{i+1}. This proof consists, for each vertex $u \in V(G)$, one of the two claims "$u \in R_{i+1}$" or "$u \notin R_{i+1}$," along with a proof σ_u that certifies the claim. If the claim is that $u \in R_{i+1}$, the proof σ_u is simply a path of length at most $i+1$ from s to u, in the right format so as to pass the test CHECK-PATH$(G, s, u, i+1, \sigma_u)$; if the claim is that $u \notin R_{i+1}$, the proof σ_u is one that would pass the test CHECK-UNREACHABILITY$(G, s, u, i+1, \sigma_u)$. The verification step for this proof is to verify all the claimed assertions, and *count* the number of assertions of the form "$u \in R_{i+1}$," so that at the end of the verification step, the verifier has the value r_{i+1}.

INDUCTIVE-COUNTING$(G, s, i, \rho, \Pi = \langle\langle 1, c_1, \sigma_1\rangle, \langle 2, c_2, \sigma_2\rangle, \ldots, \langle n, c_n, \sigma_n\rangle\rangle)$
    (assumption: $\rho = r_i$;
      returns r_{i+1} if for each u, $1 \le u \le n$, $c_u \in \{0, 1\}$,
      and σ_u is a valid proof of the assertion $u \in R_{i+1}$ (resp. $u \notin R_{i+1}$)
      if $c_u = 1$ (resp. $c_u = 0$); otherwise returns -1)

$u \leftarrow 1$
$r \leftarrow 0$
while $u \le n$ do:
    if $c_u = 1$ then
        if CHECK-PATH$(G, s, u, i, \sigma_u) =$ FAIL
            then return -1
            else $r \leftarrow r + 1$
        endif
    else
        if CHECK-UNREACHABILITY$(G, s, u, i, \sigma_u) =$ FAIL
            then return -1
        endif
    endif
$u \leftarrow u + 1$
end-while
return r

Finally, to prove that $\langle G, s, t\rangle \notin$ DSTCON, the proof will consist of "blocks" $\Pi_1, \Pi_2, \ldots, \Pi_{n-1}$, where block Π_i will empower a logspace verifier (with one-way access to the proof) to compute $r_i = |R_i|$ from r_{i-1}. In addition, the proof will consist of a claimed value R for r_{n-1}, and R distinct vertices that are claimed to be reachable from s, along with proofs of their reachability.

UN-CONNECTED$(G, s, t, P = \langle R, \Pi_1, \ldots, \Pi_{n-1}, v_1, \pi_1, \ldots, v_R, \pi_R\rangle)$
accepts iff P is a proof that $\langle G, s, t\rangle \notin$ DSTCON
$i \leftarrow 0$
$\rho \leftarrow = 0$
while $i < n - 1$ do:
    $r \leftarrow$ INDUCTIVE-COUNTING(G, s, i, ρ, Π_i)
    if $r = -1$ reject
    else $\rho \leftarrow r$
    $i \leftarrow i + 1$
end-while
if $\rho \neq R$ reject
$j \leftarrow 1$
$v \leftarrow 0$
while $j \leq R$ do:
    if $v_j \leq v$ reject
    if $v_j = t$ reject
    if CHECK-PATH$(G, s, v_j, n - 1, \pi_j) =$ FAIL reject
    $j \leftarrow j + 1$
end-while
accept