

S-T Path Problem aka Graph Accessibility Problem (GAP)

INST: A directed graph  $G=(V,E)$ , and nodes  $s,t \in V$ . source sink

QUES: Is there a path from  $s$  to  $t$  in  $G$ ? wlog  $G$  can be acyclic.

Hence  $NE_{NFA}$  and thus  $E_{NFA}, E_{DFA}, NE_{DFA}$  are all in  $P$

GAP  $\in P$ : we can do Breadth-First Search (BFS) from  $s$  to  $t$ . When BFS closes, the set  $B$  of nodes reachable from  $s$  in time given "random access" to generate neighbors of nodes. answer is yes  $\Leftrightarrow t$  is in the final  $B$ .

In fact  $GAP \in NL = NSPACE[O(\log n)]$ .

$R_1 = s$   
 While  $(R_1 \neq t)$ :  
 $N$ : Traverse the read-only input tape to guess an out-neighbor  $j$  of  $R_1$ .  
 Overwrite  $R_1$  by  $j$ .  
 If  $R_1 = t$  accept.

$R_1$  1, 2 1, 5 ... i, j 5, 8 29, 29  
 $R_2$  29 label of target state  $t$

$S=1$  Read only  
 $R_1$  1, 2 label of current state initially  $S=1$   $R_1$   
 $R_2$  29 label of target state  $t$   $R_2$

$O(\log n)$  wide  
 $N$  uses only  $\log$  space.

Non-deterministic: If there is a path from  $s$  to  $t$ , then some computation by  $N$  will find it, so  $\langle G, s, t \rangle \in L(N)$ .  
 Else,  $N$  cannot accept.  $\therefore L(N) = GAP$ .

Friday we will see  $NL \subseteq P$  by BFS in a graph built from possible IDs of  $N$ .

How about "GST<sub>2</sub>" : Given nodes  $s_1, s_2$  and  $t_1, t_2$ , are there paths from  $s_1$  to  $t_1$  and from  $s_2$  to  $t_2$ ? — that have no vertices in common?

DST<sub>2</sub> "Disjoint S-T Paths".

GST<sub>2</sub> is also in  $NL \subseteq P$

In  $NL$  again, so in  $P$ .

This works even if we have a variable number  $s_1, \dots, s_r$  and  $t_1, \dots, t_r$  of source and target nodes for the disjoint increasing connecting paths. But if  $r \neq O(1)$ , this is more than  $O(\log n)$  space. But in general:

DISJOINT CONNECTING PATHS  
 DCP, could be called "GST<sub>r</sub>" with  $r$  variable — i.e. can depend on  $G$ .

INST: Directed  $G, S=(s_1, \dots, s_r), T=(t_1, \dots, t_r)$ .

QUES: Are there paths  $s_1$  to  $t_1, \dots, s_r$  to  $t_r$  such that no two paths share a vertex?

29  
s<sub>2</sub>  $R_1$   
t<sub>2</sub>  $R_{t_2}$

It's from  $(s_1)$  for  $N$

Theorem:  $(3)\text{-SAT} \leq_m^p \text{DCP}$ , so  $\text{DCP}$  is NP-complete (it is clearly in NP because for NP we need not care how much space is used) Finally we

Proof: Given  $\phi = C_1 \wedge C_2 \wedge \dots \wedge C_m$  in variables  $x_1, \dots, x_n$ , make two kinds of source-target pairs. How about "GAPs"  $S_i \rightarrow T_j$

- $S_i \rightarrow T_i$  for  $i = 1$  to  $n$ . Each has a "high road" for  $x_i = 1$  and a "low road" for  $x_i = 0$ . (GAPs is also)
- $S_j \rightarrow T_j$  for  $j = 1$  to  $m$ . Each has <sup>up to</sup> 3 possible paths, one for each literal  $x_{i(j)}$  or  $\bar{x}_{i(j)}$  in  $C_j$ . In NL again,

Goal: Show the "vertical"  $S_i \rightarrow T_i$  paths allow non-conflicting "horizontal"  $S_j \rightarrow T_j$  paths for all  $j$  if and only if the assignment  $(a_1, \dots, a_n)$  induced by the former satisfies each  $C_j$  via the path of the satisfied literal. This works even if a variable number and of source and to But if  $r \neq 0$

eg.  $\phi = (\bar{x}_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \bar{x}_3 \vee x_4) \wedge (x_2 \vee x_3 \vee \bar{x}_4)$

Then disjoint paths exist iff  $\phi$  is satisfiable so the reduction is correct. DISJOINT CONN

Q: Can we label the added nodes so that labels increase along each path? DCP, could be with  $r$  variable - is

The answer to the final question, as noted on Piazza by one of our undergraduate visitors, is \*yes\*: we simply use a "row-major" (or "column-major" or one that fans out from northwest to southeast) indexing which will be increasing down and to the right everywhere. What this means is that the problem remains hard even under the restriction that the paths must be increasing with regard to the vertex labels. In that case, however, one needs to have a variable number of paths for the problem to be hard---if the  $r$  parameter is constant then the problem with that restriction belongs to NL as we saw. What we may explore further is the opposite possibility: if we have a fixed number  $r$  of paths but they can go up and down with regard to the node labels, how hard is the problem then?