NL vs coNL

$S \qquad coS = \{f : 1-f \in S\}.$

FACT: $P = coP$

$f \in P \iff 1-f \in P$

FACT: $L = coL$

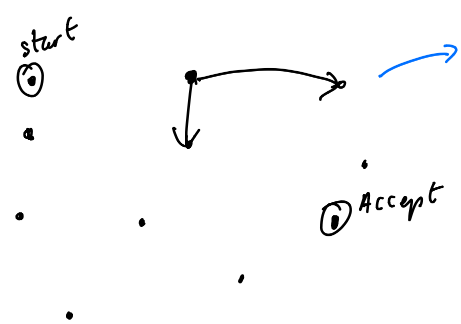
FACT: $Exp = coExp$

$NP \stackrel{?}{=} coNP$

3-SAT

Thm: $NL = coNL$.

$G_{M,x}$



vertices correspond to config.
- contents of work tape
- pointers
- line of code

$2^{O(s)}$ vertices

If

$f \in NL$

Claim: If $\exists$ logspace algorithm for computing connectivity then $L = NL$.

Thm: (main Thm) There is a non-det logspace algorithm $M(G,s,t)$ s.t

1. If there is a path from $s \to t$ then algorithm always rejects.
2. If there is no path from $s \to t$ then some computational path accepts.

Cor: $NL \subseteq coNL$, $coNL \subseteq NL$.

Cor: For space constructible $s(n) \geq \log n$, $NSPACE(s(n)) = coNSPACE(s(n))$.

Proof of Main Thm

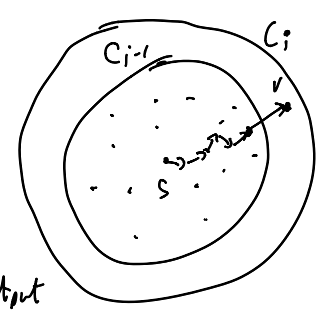
C_i = set of vertices reachable from s in $\leq i$ steps.

Claim: Given a vertex v and i, there is a non-det logspace algorithm s.t

- If $v \in C_i$, then some computational path outputs 1
- If $v \notin C_i$, then every " " " 0.

Claim 2: Given $|C_{i-1}| = c$, and a vertex v, there is a non-det logspace algorithm s.t

- if $v \notin C_i$ there some computational path that outputs 1
- if $v \in C_i$ then every " " outputs 0.



Algorithm

1. count $= 0$, $w = 0$
2. Guess $u \in C_{i-1}$
3. If $u > n$ or $u \leq w$ output 0, else set $w = u$.
4. Use claim 1 to check $u \in C_{i-1}$, if not output 0.
5. count = count + 1.
6. If edge (u,v) or $u = v$, output 0.

7. If count $= c$ halt with output 1.

8. Go to step 2.

Claim 3: Given the size of $|C_{i-1}| = c$ there is a non-det logspace algorithm s.t either the algorithm outputs 0 or it outputs $|C_i|$, and there is a computational path on which it outputs $|C_i|$.

Alg ($|C_{i-1}| = c$)

count $= 0$

For $u = 1, \dots, n$

Guess $u \in C_i$

count $=$ count $+1$.

Use claim 1 to verify $u \in C_i$ (if output is 0 halt with 0)

Guess $u \notin C_i$

Use claim 2 to verify $u \notin C_i$.

(if output is 0 halt with 0)

end for

output count.

Final Algorithm

set $c_0 = |C_0| = 1$

Compute $|C_1|, \dots, |C_{n-1}|$

Use claim 2 to verify $t \notin C_n$.