12/11/20

TOC - TEST-4

Kousshik Raj
17CS30022

2.

We know that graph $G_1$ is bipartite if and only if it doesn't contain odd cycles. So, to show that BIPARTITE is in NL, it is enough to give a non-deterministic log-space algorithm to detect odd cycles in a graph $G_1$. If we find odd cycles in $G_1$, then $G_1 \notin$ BIPARTITE, else $G_1 \in$ BIPARTITE

Algorithm (for $\overline{\text{BIPARTITE}}$)

i) num_nodes = 0

ii) start_node = s (Non-deterministically chosen node)

iii) next_node = t (Non-deterministically chosen neighbour node of s)

iv) while num_nodes $\leq |V(G_1)|$ :
- -> if next_node = start_node :
- -> if num_nodes is odd :
- -> ACCEPT
- -> next_node = r (Non-deterministically chosen neighbour node of 'next_node')
- -> num_nodes += 1

v) REJECT

Space used :- num_nodes (log space)

start_node (constant space)
next_node (constant space)

$\Rightarrow$ BIPARTITE $\in$ NL (as $\overline{\text{BIPARTITE}} \in$ NL and NL = co-NL)

Kousshik Raj
17CS30022

1. ~~We know that~~

No, it is not true.

- We know that NL & coNL are the same, (i.e) NL = coNL. So, coNL completeness is same as coNL-completeness and thus PATH and $\overline{PATH}$ are both NL-complete.

~~PATH ∪ $\overline{PATH}$ = L~~

Consider, PATH ∪ $\overline{PATH}$ = L. Then L is the set of all strings (i.e) $L = \Sigma^*$. Obviously, L cannot be a complete language in any useful complexity class.

> Thus, union of two NL-complete class need not be NL-complete

Reason L cannot be a complete language because L is a trivial language

---

3. We have to show that

$$A \in \Sigma_i^P \implies h(A) \in \Sigma_i^P.$$

Let $S \in \Sigma_i^P$. Then there must be an ATM (alternating TM) that takes in input x and determines whether $x \in S$ in at most i-1 alterations. Now we need to find another ATM with atmost i-1 alterations that determine whether a

Koushik Raj
17CS30022

string $y \in h(S)$. But $y \in h(S)$ iff $\exists x$ s.t $x \in S$ & $y = h(x)$.

The rough idea for the ATM that decides $h(S)$ will be to non deterministically choose a string $x$, use the ATM that decides $S$ to determine whether $x \in S$, and check whether $h(x) = y$. If yes, then accept.

We can see that the new ATM also uses atmost $i-1$ alterations and hence $h(S) \in \Sigma_i^P$ (because $|h(x)| \geq |x|$)

Thus, $S \in \Sigma_i^P \Rightarrow h(S) \in \Sigma_i^P$