1

TOC – TEST – 3

Kousshik Raj M
17CS30022

2. We will prove this for time complexity, a similar arg. can be made for space complexity.

Let M_{L1} be a TM that decides a language $L \in CL_1(f(n^c))$ in $f(n^c)$ time for any $c \geq 2$.

Now, consider the language created from padding the language L,

$$L_{pad} = \{ x01^{|x|^c - |x| - 1} \mid x \in L \}$$

Now, let M_{Lpad} be a TM that decides the language L_{pad}. It is easy to M_{Lpad} determines whether a string y is of the form $x01^{|x|^c-|x|-1}$ in $f(|x|^c) = f(|y|)$ time.

$\Rightarrow L_{pad} \in CL_1(f(n))$

$\Rightarrow L_{pad} \in CL_2(g(n))$ {from initial statement}

$\Rightarrow \exists$ TM N_{Lpad} that decides L_{pad} in time $g(n)$.

From this, we will construct a new machine N_L that decides the language L by using N_L as a subroutine (i.e)

$$N_L(x) \equiv N_{Lpad}(x01^{|x|^c-|x|-1})$$

~~$\Rightarrow L \in CL_2$~~

N_L works in $|x|^c$ time.

$\Rightarrow L \in CL_2(g(|x|^c))$

$$\boxed{\therefore \; CL_2(g(n^c)) \supseteq CL_1(f(n^c))}$$

2) If we can give a polynomial space algorithm for the GTICTACTOE language, we can say GTICTACTOE ∈ PSPACE.

Kousshik Raj
17CS30022

Algorithm :-

→ i) First check validity of the configuration c, by checking if there are equal number of X's & O's in the board, since X starts first and now it is O's turn — Done in $O(nm)$ space.

ii) Now, check if there are k consecutive X's or O's or both are present. If only X's are present, we can accept (as X as won). But if there is only O's (O as won) or both X's & O's (invalid config), reject. — Done in $O(nm)$ space, as we need space only to store the entire board.

→ REC (c) :-
(Here c is a valid board configuration in which the current turn is X)

~~i) If ...~~

i) Check if there are k consecutive X's. If they are, then accept

ii) If c is full, reject

iii) Now create a board config. c_{next} by placing X in an empty cell. Now, generate c_{poss_i} by generating all board configurations from placing a O in c_{next}.

Kousshik Roy
17CS30022

iv) call REC(C_{poss_i}) for all $poss_i$. If all of them accept, accept. Else reject.
Now, clear the space used and reuse space for each C_{poss_i}.

v) Repeat steps iii) & iv) for all such position where X can be placed. If any of them accept, accept. Else reject.

The only place where memory is used, is in step iii) & iv) where we remember C, C_{next}, C_{poss_i} which is $O(nm)$ and the number of times we need to remember them together is same as the recursion depth, which is $O(nm)$. Therefore total space used is $O((nm)^2)$.

$\Rightarrow$ GTICTACTOE is in PSPACE

Kousshik Raj
17CS30022

1. The relationship is $P \,!= DSPACE(n^c)$.

Suppose by contradiction let $P = DSPACE(n^c)$. Then there exists an algorithm that simulates a TM that uses n^c space. But this means there exist an algorithm that is decided in n^{2c} space (by the padding argument; there exists such an algo) TM is also in P because both uses polynomial time.

$$\Rightarrow P = DSPACE(n^c) = DSPACE(n^{2c})$$

But this contradicts the SPACE hierarchy theorem, hence $P \,!= DSPACE(n^c)$ for any c.