Computational Complexity

Example Solutions

Henrique Campos Navas

2023-2024

Homework 2

4.2

Clearly, $SPACETM$ is in $PSPACE$, as we can simply simulate M on input w while using space n in linear space.

It remains to show that every problem in $PSPACE$ can be reduced in polynomial time to $SPACETM$. Let L be a language in $PSPACE$. Then, there is are $k, c \in \mathbb{N}$ and a Turing machine M_0 such that M_0 decides L in at most kn^c space. Now, we define $f(x)$ as $\langle M_0, x, 1^{kn^c} \rangle$. f can be computed in polynomial time, since we can write M_0 in constant time, x in linear time and 1^{kn^c} in $O(n^c)$ time. Clearly, $x \in L \Leftrightarrow f(x) \in SPACETM$, since $x \in L$ if and only if M_0 accepts 0, which it does in kn^c space. Thus, f provides us we a polynomial time reduction from L to $SPACETM$. As L can be any language in $PSPACE$, we have that $SPACETM$ is $PSPACE$-complete.

4.3

We prove that every such language L is NL-hard (for it to be NL-complete we need it to be in NL).

Since L is neither the empty set nor $\{0,1\}^*$, we have that there are strings w_0 and w_1 such that $w_0 \notin L$ and $w_1 \in L$. Let L' be a language in NL. We define $f(x)$ as w_0 if $x \notin L'$ and as w_1 if $x \in L'$. Since L' is in NL, it is also in P, and thus we can compute f in polynomial time. Thus, we have a polynomial time reduction from L' to L for every $L' \in NL$, and thus L is NL-hard.

4.10

We do induction on n. If $n = 1$, then either the first player was a winning move, and thus a winning strategy, or the second player always wins. Now we prove that if every game that ends in at most n moves has a winning strategy for one of the players, then every game with at most $n+1$ moves has a winning strategy for one of the players.

Let G be a game that ends in at most $n+1$ moves. The first player has a certain set of possible plays. Each one of these plays will result in a game G_x that ends in at most n moves and in which they are the now second player. By the induction hypothesis, all of these games have a winning strategy for one of the players. If all of these games have a winning strategy for the first player, then the second player has a winning strategy for G, they just have to simulate the winning strategy for whatever game results from the first play as if they were the first player. Otherwise, there if there is at least one game G_a in which the second player is the one with a winning strategy, then the first player has a the following winning strategy. In the first play of G, they make a move such that the game is reduced to G_a and afterwards they simulate the winning strategy for the second player on G_a. Thus, G has a winning strategy for one of the players, and or induction step in done.

With this, we have the statement proved.