CMU 15-855* Fall 2017
Graduate Computational Complexity Theory
Homework 1

Zhikun Wang*

February 9, 2021

For problem statements and course notes, please see this 120MB pdf file published by Ryan O'Donnell.

This homework is not scored or reviewed by a professor or a TA. If you believe you've found a mistake then please never hesitate to email me or comment on my blog! Thanks!

This homework is completed without any form of discussion.

1 Valiant's Depth-Reduction Lemma.

1.1

As the "depth" of a graph is defined to be the length of the longest path in the graph and every vertex in a path will contain different labels, all paths must have a length less than d. Conversely, we can use $l(v)$, the length of the longest path ending at v to be the label of vertex v. It is obvious that as $l(u) = \max_u(l(u)+1)[(u,v) \in E]$, $l(u) \geq l(v) + 1$ for all directed edges $(u, v) \in E$.

1.2

For all edges $(u, v) \in E$ that remains in the graph after deletion, there are two cases: Most significant different bit of $l(u)$ and $l(v)$ is more significant than kth bit, then after deletion $l(u) < l(v)$ as the kth bit is never used for comparison. Otherwise, the kth bit will be the same for $l(u)$ and $l(v)$ and therefore deleting it will not affect the comparison.

1.3

To reduce the depth to less than $d/2^r$ using the method discussed above, it is necessary r sets in $E_0, E_1, \ldots, E_{j-1}$ where E_j contains all edges (u, v) such that the most significant bit where $l(u)$ and $l(v)$ differs is the jth. As every edge is in one of k sets, the r sets with least amount of total edges will contain no more than $(r/k)m$ edges.

2 Block-respecting TMs.

2.1 The tapes used

In this solution M' uses $3k + 1$ tapes, where we use 3 tapes for each tape in M. We name these tapes the working tape, the positioning tape and the register tape respectively. The 1 other tape is called the clock tape.

*nocrizwang@gmail.com

2.2 The clock

First, in the clock tape M' configures a clock. M' calculates $B(n)$ on the tape and write 1 and 2, separated by $B(n) - 2$ 0s on the first $B(n)$ cells of the clock tape. The clock runs non-stop in parallel with the other movements. This is achieved by duplicating all states by a constant c, the states the clock would need.

The clock runs as follows:

- The clock starts from the leftmost cell with 2 in it.
- The clock head runs from the left to the right until it sees the 2 on the right border.
- Then, the clock stays there for 1 step.
- The clock head runs from the right to the left until it sees the 1 on the left border.
- Then, the clock stays there for 1 step.
- The clock moves right again, as in step 2.

The head of the clock tape never crosses block borders.

2.3 The tracks

The working tape of M' contains 3 tracks. For every block on the working tape one track is used to store the same contents of M it is trying to simulate, and the other two tracks contain the reverse of the block left and right to it.

2.4 The positioning tape

The positioning tape is initialized to have the same content as the clock tape. The purpose of the positioning tape is for M' to see if the tape head has reached the border of a block. The head movement of the positioning tape is exactly the same as the corresponding working tape when the tape head does not cross borders. When the tape head of the working tape crosses borders, the tape head of the positioning tape will move in the reversed direction.

The head of these positioning tapes never cross block borders.

2.5 Movement of tape head in M'

Duplicate states in the TM so that when the head would cross the border of the working tape(e.g. when the positioning tape is 2 and the tape head is moving right), the head would not move, and would change the track it is working on instead. When working on the reversed track, the tape head movements would be reversed.

2.6 The simulation

The timer would let M' simulate M like this for $B(n)$ steps. Then M' enters a mode when the tape heads on the working tape are moved to the actual blocks they are supposed to be in and reversed track contents of the block the heads were on before the break happens is copied to the corresponding not-reversed tracks of nearby blocks.

This happens as follows: First, the TM marks where the head was on the positioning tape using a character 4. Then, all heads are moved to the left block border, and the right reversed track is copied to the register tape. After copying the whole block, the TM waits until the clock shows that it is OK to cross the border. Then it cross the border and copies from the register tape to the not-reversed track of the nearby right block. After copying the other reversed tape to the left block in a similar way, the TM moves the tape head to the supposed position based on the mark left on the first extra tape. and removes the mark. Then, the M' starts to run $B(n)$ steps simulating M again.

This solution uses $3k+1$ tapes and $O(t(n))$ time.

3 Improving the Time Hierarchy Theorem via padding.

3.1

Suppose language $L \in \mathsf{TIME}(t_2(f(n)))$ and TM M decides it in $t_2(f(n))$ time. Then the language $L_{pad} = \{\langle x, 1^{f(|x|)}\rangle \mid x \in L\}$ is in $\mathsf{TIME}(t_2(n))$. The TM runs as follows: Check if the input string is in the given format of $\langle x, 1^{f(|x|)}\rangle$, otherwise reject. Then, run M on x. As it is given that $\mathsf{TIME}(t_1(n)) = \mathsf{TIME}(t_2(n))$, $L_{pad} \in \mathsf{TIME}(t_1(n))$. At last we show an algorithm to solve L in $\mathsf{TIME}(t_1(f(n)))$: First pad the string to $\langle x, 1^{f(|x|)}\rangle$, then run the $\mathsf{TIME}(t_1(n))$ algorithm on it.

3.2

The contrapositive of 3.1 is as follows: $\mathsf{TIME}(t_1(f(n))) \neq \mathsf{TIME}(t_2(f(n)))$ implies $\mathsf{TIME}(t_1(n)) \neq \mathsf{TIME}(t_2(n))$.

From the time hierarchy theorem we see that $\mathsf{TIME}(n^5 \log^{1.1} n) \neq \mathsf{TIME}(n^5)$. Let $f(n) = n^{\frac{5}{3}}$, then we get $\mathsf{TIME}(n^3 \log^{0.66} n) \neq \mathsf{TIME}(n^3)$. As we now know that $\mathsf{TIME}(n^3) \subsetneq \mathsf{TIME}(n^3 \log^{0.66} n) \subseteq \mathsf{TIME}(n^3 \log^{0.75} n)$, $\mathsf{TIME}(n^3) \subsetneq \mathsf{TIME}(n^3 \log^{0.75} n)$.

Thus we have proved $\mathsf{TIME}(n^3 \log^{0.75} n) \neq \mathsf{TIME}(n^3)$.