Complexity Assignment 2

Name: Shubh Sharma, Satya Sreevani Bh
Roll Number: BMC202170, BMC202162

3.

We need to show that by fixing $x, y \in \mathbb{F}_2^n, u, v \in \mathbb{F}_2^k$ we get

$$Pr(Ax + b = u \wedge Ay + b = v) = \frac{1}{2^{2k}}$$

Multiplication by a matrix A is the same as taking dot product with single rows of the matrix, that are picked indpendently from each other.
Probability that $x \cdot u = 1$ where x is a non zero vector is $1/2$, to show that, consider the case where x has exactly l 1s and the rest $n - l$ are 0s. Then let the probablilty that $x.u = 1$ be p_l, let p_{l-m} be the probablility that the vector which is x with first m 1 fillped times u is 1, then we have the recurrence relation
$p_l = \frac{1}{2}(1 - p_{l-m}) + \frac{1}{2}p_{l-m} = p_{l-m}$
This recurrence relation holds because the dot product is defined simply by counting the number of positions in which u has 1(that correspont to locations on x) hence the recurrence says the total number of matching positions is odd, if first one matches and even many of the following match, or the first one does not match and odd many of the following match
also $p_1 = \frac{1}{2}$ hence $p_l = \frac{1}{2}$
This means that the i^{th} index of Ax match the i^{th} index of $u - b$ is $1/2$ and same for y and v hence probability that i^{th} index of Ax matches with $u - b$ and i^{th} index of Ay matches $v - b$ is $1/4$
Doing the same for all k components gives that

$$Pr(Ax + b = u \wedge Ay + b = v) = \frac{1}{2^{2k}}$$

hence the familty $\mathcal{H}_{n,k}$ is a set of pairwise independent hash functions.

4. Prove that if $P = NP$ then $EXP = NEXP$

Given any language in EXP, we can accept in by a Non Deterministic Turing Machine in the same time, by not using non determinism in the turing machien, hence $EXP \subseteq NEXP$

If $P = NP$
Consider a language $L \in NEXP$, then there is a Non-Deterministic Turing Machine that accepts or rejects any word from L in time $O\left(2^{n^c}\right)$. Given any word, $l \in L$ we can construct the word $l\#1^{n^c}$ by appending n^c $1s$ to the word, this new language $L' = \{\, l\#1^{n^c} : l \in L, n = |l| \,\}$ is accepted by a Non-Deterministic Turing Machine in time $O(n^c)$.
Since $NP = P$ There exists a deterministic Turing Macine M that has the language L' in time $O(n^{c'})$.
Now for any given $l \in L$, follow these steps on a Deterministic Turing Machine

- Given a word of length n, appened $O(n^c)$ 1s after the word.
- Run the machine M on the input
  These steps would accept the word in time $O\left(2^{n^{c'}}\right)$ hence L is accepted by a Deterministic Time Turing machien in exponential time, this $NEXP \subseteq EXP$

Thus $EXP = NEXP$

6.

$MA \subseteq PSPACE$ because $MA \subseteq AM \subseteq IP = PSPACE$

To show that if $PSPACE \subseteq P/Poly$ then $PSPACE \subseteq MA$
We use the result that a problem in PSPACE can be simulated using an IP protocol where the computation done by the prover is in PSPACE
let $L \in PSPACE$, Then there exits a polynomial size circuit family C that computes the provers message in the IP protocol for L
We use the following MA protocol for L
Merlin sends Arthur the circuit family C bounded by some polynomial of the input length.
Arthur then simulates the circuit family on the input to get the prover's message at each step. Arthur then accepts iff the verifier it is simluating accpets.

This computation takes polynomial time because Arthur has to check polynomially many circuits atmost polynomially many times.(is IP protocol has polynomie)

If $x \in L$ then Merlin can start with a message that will cause Arthur to accept with probability $\frac{3}{4}$ and if $x \notin L$ then never accpets with probability more than $\frac{1}{4}$ (by completeness and soundness of IP protocol)
hence $PSPACE = MA$