# [Homework 3]: More on Markov chains

## Problem 1 (FTMC for countably infinite chains)

Recall that the fundamental theorem of Markov chains states that [F] + [A] + [I] implies [S]+[U]+[C]. In this problem, we will develop its generalization, namely, [PR] + [A] + [I] implies [S]+[U]+[C] (Please refer to our lecture notes for the meaning of these abbreviations). We assume $\Omega$ is the state space of the Markov chain, which can be finite or countably infinite. Let $P \in [0,1]^{\Omega\times\Omega}$ be the transition function of the chain. That is, for every $i, j \in \Omega$, $P(i,j) = \mathbf{Pr}\left[X_{t+1} = j \mid X_t = i\right]$. Assume $P$ has the properties of [PR], [A] and [I].

1. Prove that this is indeed a generalization. That is, "[PR] + [A] + [I] implies [S]+[U]+[C]" implies "[F] + [A] + [I] implies [S]+[U]+[C]".
2. We now define another Markov chain on $\Omega^2$. For each pair $(i,j) \in \Omega^2$, the chain moves to $(i',j') \in \Omega^2$ following $P$ coordinate-wise and independently. (That is, $\mathbf{Pr}\left[(X_{t+1}, Y_{t+1}) = (i',j') \mid (X_t, Y_t) = (i,j)\right] = P(i,i') \cdot P(j,j')$.) Prove that in this chain, for any $i, j, k \in \Omega$, it holds that $\mathbf{Pr}_{(i,j)}\left[T_{(k,k)} < \infty\right] = 1$.
3. Use above to prove the FTMC for countably infinite chains.

## Problem 2 (A Randomized Algorithm for 3-SAT)

Recall the randomized algorithm we developed for 2-SAT. In the problem, we apply it to those 3-SAT instances. Since 3-SAT is NP-complete in general, we cannot expect it to terminate in polynomial-time and output the correct answer with high probability. However, it is still better than the brute-force algorithm sometimes. Let $n$ be the number of variables of the input formula.

1. In the 2-SAT algorithm shown in the class, if we repeat the random flipping operation for $100n^2$ times, then the algorithm outputs the correct answer with probaiblity at least $(1 - \frac{1}{100})$. Consider the following way to boost the correct probability. The algorithm only repeats the random flipping operation for $2n^2$ times. If it outputs a satisfying assignment, then we just output as it is. Otherwise, we run the algorithm again. Repeat this for 50 times (So the total number of iterations is still $100n^2$). If all these algorithms claim the formula is not satisfiable, then we output "not satisfiable". What is the probability of correctness of our new algorithm?

2. Now we apply our algorithm on a 3-SAT instance (Now in each step, if $\sigma_t$ is not satisfiable, we choose an unsatisfied clause, pick one of its three literals uniformly at random, and flip its value). Assume the same notations in the class. Prove that $\mathbf{Pr}\left[X_{t+1} = X_t + 1\right] \geq \frac{1}{3}$ and $\mathbf{Pr}\left[X_{t+1} = X_t - 1\right] \leq \frac{2}{3}$.

3. Prove that in order for our algorithm to be correct with probability $0.99$, we need to repeat the random flipping operations for $\Theta(2^n)$ times.

4. Suppose we start with $X_0 = n - i$ for some $i > 0$. Can you find a good lower bound for the probability $\mathbf{Pr}\left[\exists t \in [1, 3n] : X_t = n\right]$?

5. Now we consider how to improve the performance of the algorithm. Suppose the input formula is satisfiable. The following observation is crucial. The $X_t$ is more likely to decrease, and therefore, when $t$ is large, it is more likely to be close to $0$. This observation suggests us that we should not repeat the flipping process for too long. As a result, we only repeat the flipping process for $3n$ times. Suppose we start with some $\sigma_0$ which is uniform at random from all $2^n$ assignments of the variables. What is the probability that the algorithm outputs a satisfying assignment?

6. Design an algorithm to find a solution for 3-SAT with probability 0.99. Determine the smallest constant $c \in (1, 2]$ so that your algorithm terminates in $n^{O(1)} \cdot c^n$ time.