From Monkeys to Markov Chains: Analyzing Random Character Sequences

Gabriel Cardella
Advisor: Dr. Kevin Milans

Project Introduction

The Infinite Monkey Theorem, as Banerji, Mansour, and Severini note, is a popular concept involving probability that states that a monkey hitting random typewriter keys for a sufficiently long period of time will almost definitely type any piece of text, even works of Shakespeare. Of course, this "monkey" can refer to any device that produces a random sequence of characters. Although humorous and extreme, it does highlight that, when given enough independent trials, even events of exceedingly low probabilities will take place. The motivation of this project came from this concept, specifically the infinite generation of random characters and the appearance of given *target strings* in these sequences. The goal of this project was to develop algorithms that calculate the average *Hitting Time* (HT) of an input string, the *Probability Mass Function* (PMF) of this hitting time, and the winning probabilities for the players in a related multiplayer game.

Markov Chain Introduction

A Markov chain, named after Russian mathematician Andrey Markov, is a mathematical system that transitions between states. Markov chains are referred to as Discrete-Time Markov Chains (DTMCs) when these transitions occur at discrete time steps. A DTMC is a set of states, Q, and a transition probability function f : $Q \times Q \to \mathbb{R}$ which gives the *transition probability* P_{ab} that the next state is b given that the current state is a. These transition probabilities are usually organized into a *transition matrix*, P. See below, in Figure 1, for an example weather transition matrix and its accompanying Markov chain.

$$P = \begin{bmatrix} P_{SS} & P_{SR} \\ P_{RS} & P_{RR} \end{bmatrix} = \begin{bmatrix} 0.6 & 0.4 \\ 0.75 & 0.25 \end{bmatrix}$$

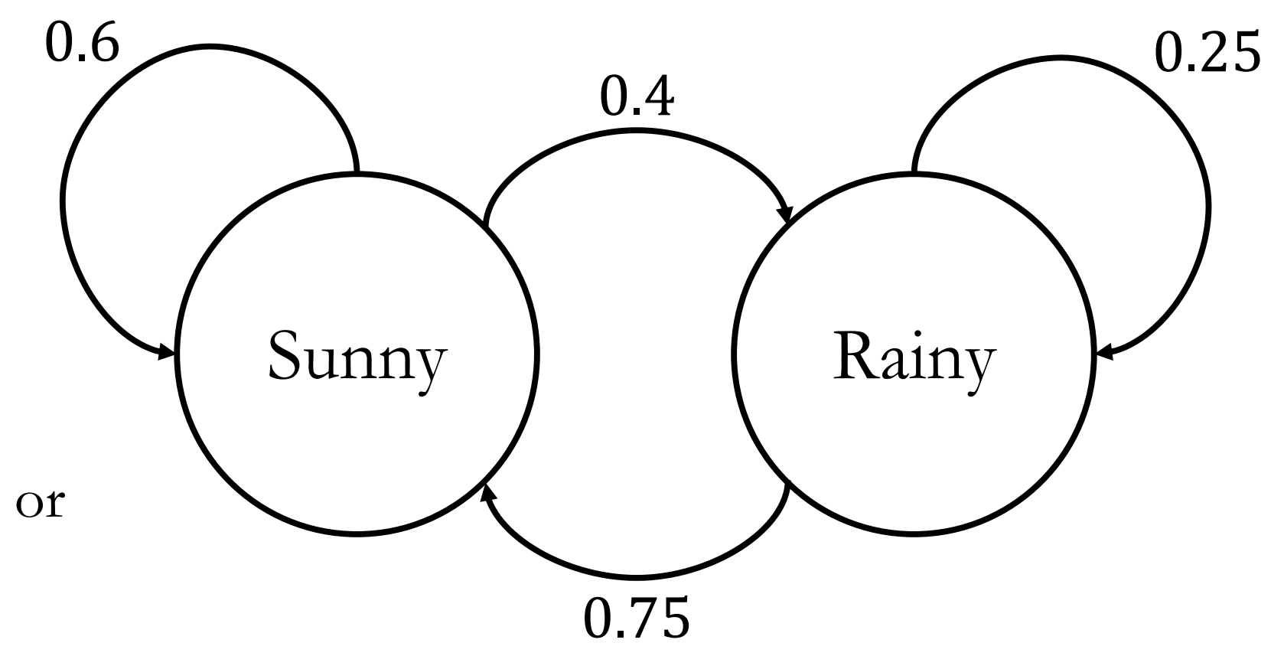


Figure 1: Example of transition matrix and Markov chain. Here, we give the probabilities that the next day's weather is either Sunny (S) or Rainy (R) given the current day's weather.

Note that there is a special type of state called an *absorbing state*. A state i is absorbing if and only if $P_{ii} = 1$. It follows that all other entries in row i are 0.

Markov Chain Application

In this project, the appearance of target strings in a random sequence of characters were represented as Markov chains. First, consider an alphabet Σ, the set of possible characters that can occur in a random string of characters r typed by our hypothetical monkey. We denote the target string by s, where $s = c_1 \ldots c_n$. Next, let $Q = \{q_0, q_1, \ldots, q_l\}$, where $q_i = c_1 \ldots c_i$. Note that q_0 is the empty string, denoted by ε. For $t \geq 0$, if s has not appeared as a substring of $r_1\, r_2 \ldots r_t$, then the state of the Markov chain at time t equals the largest suffix of $r_1\, r_2 \ldots r_t$ in $Q - \{q_l\}$. If, however, s has appeared as a substring of $r_1\, r_2 \ldots r_t$, then the Markov chain at time t is in state q_l. An example of the representation of a target string as a Markov chain is seen at the beginning of the second column of text, in Figure 2. Here, $\Sigma = \{A, B, C\}$, $s =$ "ABA", and $Q = \{\varepsilon, A, AB, ABA\}$. We use these parameters in future examples as well. Here, we also give an example string of random characters r, and which state the Markov chain is in as we traverse each character. Observe that state ABA is absorbing.

$r =$ **BABCBABCCCBABBCBACCBABACBAABC**…

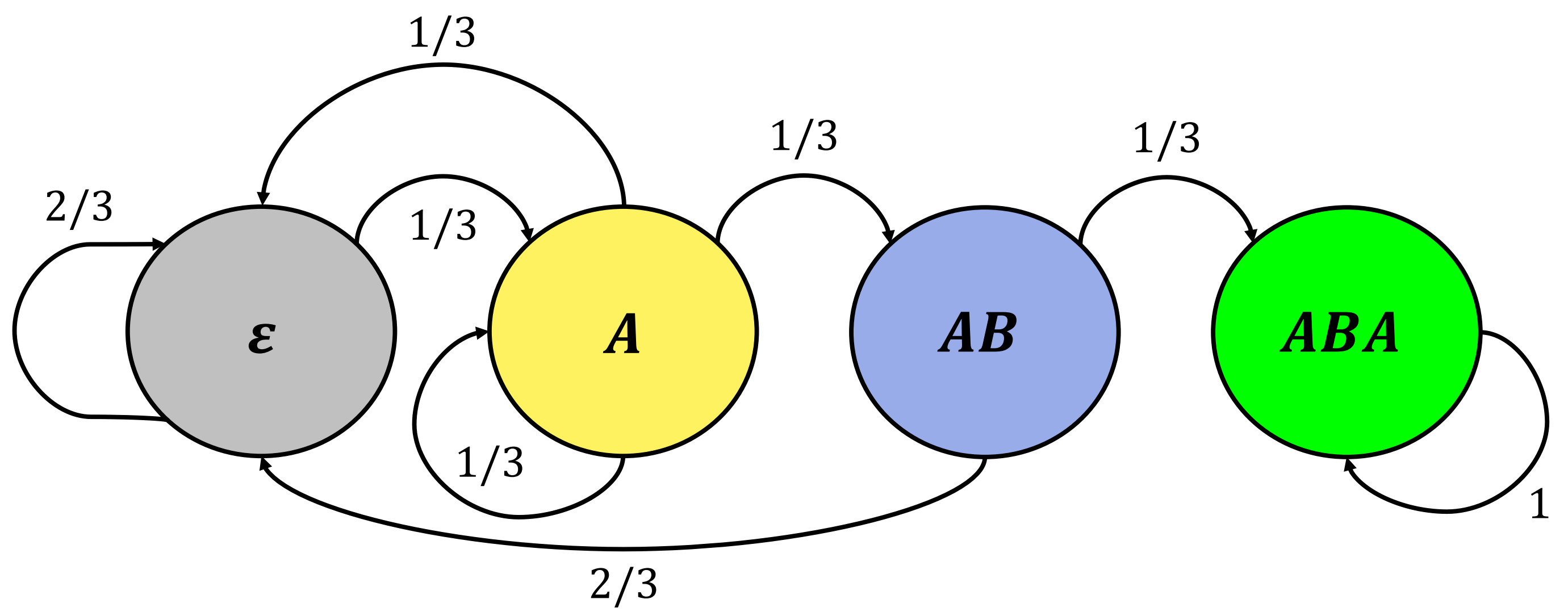


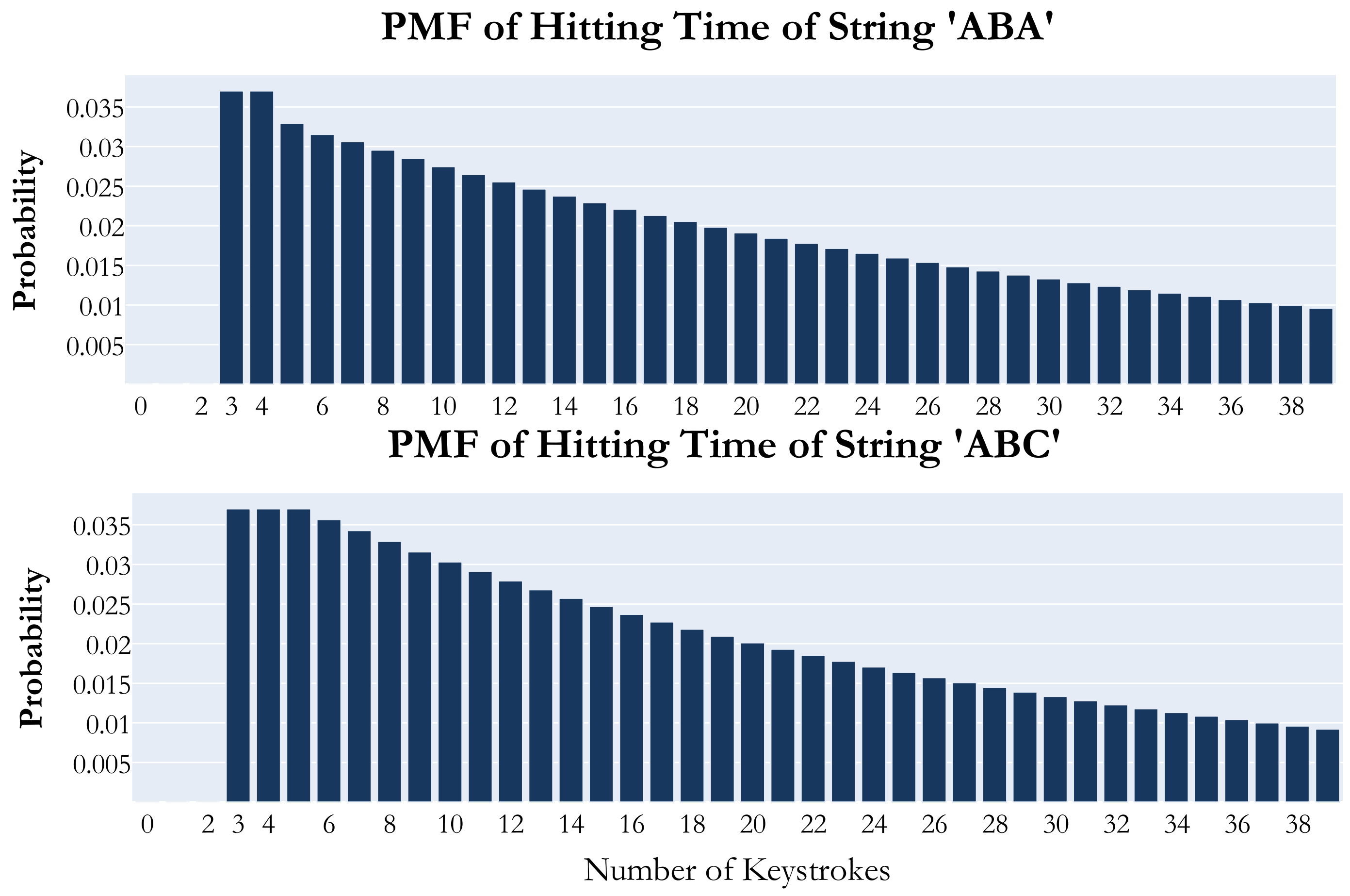
Figure 2: Markov chain representing the given target string "ABA". The color of each letter in r corresponds to the color of the state that the Markov chain transitions to after that letter is typed.

Transition Matrix and PMF Algorithms

To derive the transition matrix of a string's Markov chain, like the one above, Σ and s were traversed. For each letter in s, we examined what state the associated Markov chain would transition to if a given letter of Σ appeared next. This allowed us to populate P. To the right, in Figure 3, we see the transition matrix using the same parameters found in the "Markov Chain Application" section. Using this, we calculated the target string's HT PMF by using the formula: $\boldsymbol{\lambda_0 P^{t-1}(P - I)\overrightarrow{e_l}}$. Here, λ_0 represents the initial distribution and was set equal to the first row of the $(l + 1) \times (l + 1)$ identity matrix, I. As well, $\overrightarrow{e_l}$ represents the last column of the I, used to return only the probability that we are in the final (absorbing) state of s's Markov chain after t random keystrokes. A plot of the HT PMF of "ABA" is seen below in Figure 4. We compare this to the PMF of "ABC" to highlight the difference.

$$P = \begin{bmatrix} 2/3 & 1/3 & 0 & 0 \\ 1/3 & 1/3 & 1/3 & 0 \\ 2/3 & 0 & 1/3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Figure 3: Transition matrix of the target string "ABA".


PMF of Hitting Time of String 'ABA'

PMF of Hitting Time of String 'ABC'

Figure 4: PMF of the Hitting Time of String "ABA" and "ABC". Note that they have a different distribution, despite having the same length and alphabet.

Average HT and Multiplayer Game Algorithms

We calculate the HT by traversing each of s's prefixes found at the end of s letter-by-letter. The product of the probabilities of each letter in a prefix occurring is then calculated, with the reciprocal of the product being taken. These reciprocals are added together, returning the average HT of s. Our target string of "ABA" has an average HT of **30 keystrokes**, with "ABC" only having an average HT of **27 keystrokes**. To find the winning probabilities of a related multiplayer game involving n players each picking a target string and seeing which string(s) occur(s) first, let each state q_i representing the first i characters of s, it is now given by the coordinate $(q_{1m_1}, q_{2m_2}, \ldots, q_{nm_n})$, where q_{im_j} gives the first m_j characters of string i. To derive the associated transition matrix, we first obtained a list of *valid states* representing all progressions through the n target strings that are possible. After finding the *valid states*, the associated n-player transition matrix was derived using a similar process as the transition matrix derivation for one target string. An example Markov chain for a 2-player game is seen below in Figure 5. Here, let $\Sigma =$ {A,B}, $s_1 =$ "ABA", and $s_2 =$ "BAA".

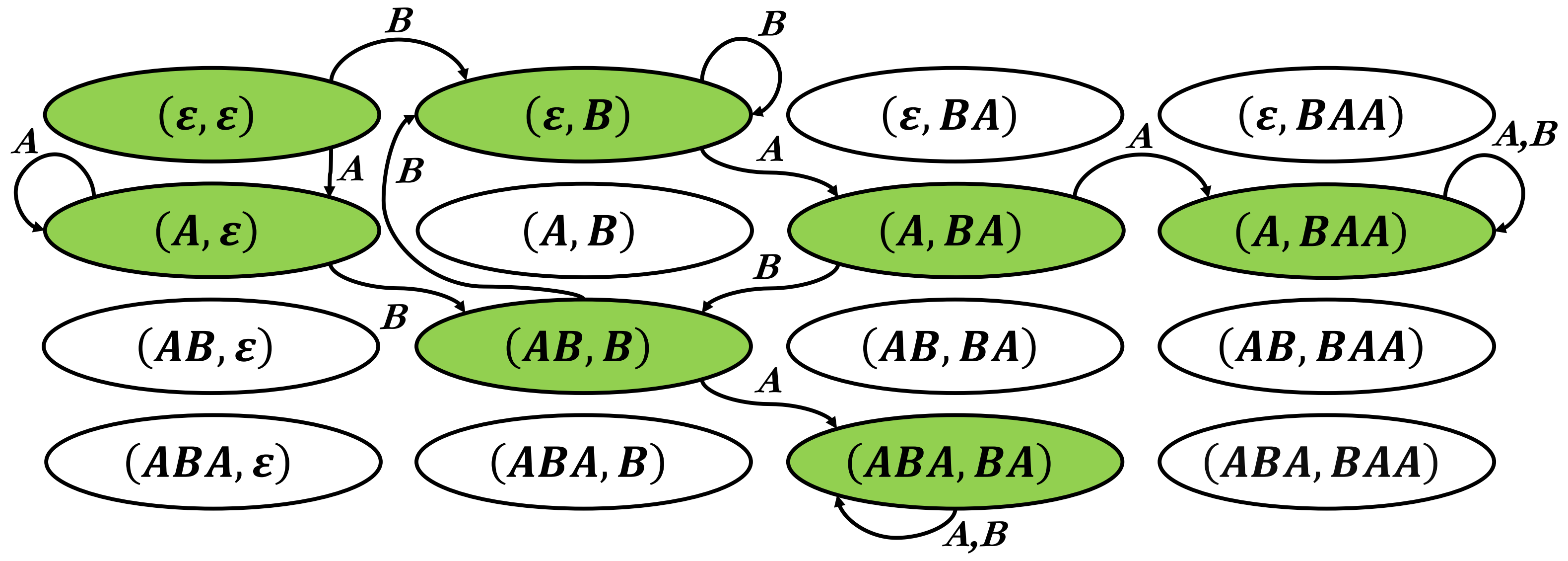


Figure 5: Markov chain that represents an example 2-player game. Note that the "valid states" are highlighted in green, and the letters that result in each transition, rather than the associated transition probabilities, are given.

Conclusion & Future Work

The motivation for the work completed in this project came from the Infinite Monkey Theorem, and the idea of the generation of long sequences of random characters. The goal of this project was to develop and implement algorithms to represent a target string as a Markov chain's transition matrix, calculate a target string's PMF and average hitting time, and play a related multiplayer game. Due to the explorative nature of this project, there is a plethora of possible future work to expand on the work done. Algorithms developed can be improved for further efficiency and readability. As well, a front-end application can be created that allows for easy interaction with the back-end code that was written. Such a tool can be useful for gamblers playing a game that involves a random sequence of outcomes either at a casino or online. It can also be useful for someone who simply wants to learn more about probability and Markov chains.

References

Christopher R. S. Banerji, Toufik Mansour, and Simone Severini. 2013. A notion of graph likelihood and an infinite monkey theorem. Journal of Physics A: Mathematical and Theoretical 47, 3 (2013). https://doi.org/10.1088/1751-8113/47/3/035101

J. R. Norris. 1998. *Markov chains.* Cambridge University Press, Cambridge, England