The *Abracadabra Problem*: Solution, Illustration and Generalization

Bernardo Lembi Ramalho Maciel

Professor: Victor Filipe Martins-da-Rocha

Measure Theory Winter Course - FGV/EESP - 2021

1 Problem and motivation

When studying the Borel-Cantelli results, a recurrent illustrative example is that of a monkey hitting random keys on a keyboard. We assume that the monkey hits the keys independently and with uniform probability. In particular, using the Second Borel-Cantelli Lemma, we conclude that the probability of the monkey producing infinitely many copies of the complete works of Shakespeare is 1. The problem now is: using the 26 uppercase letters of the keyboard, what is the expected time until the monkey types the string `ABRACADABRA`?

I believe this problem is worth studying for three main reasons:

1. It uses measure theory (in particular, martingale theory) to solve in a very elegant way a problem that in a first glance seems like a problem of combinatorics;
2. It produces a counterintuitive result;
3. The solution extends to arbitrary countably finite strings and countably finite alphabets;

2 Solution

2.1 Main idea

The approach towards the solution consists in:

- Constructing a lottery in which gamblers gains come from correctly predicting the digits of the desired string. In this way, we essentially map the problem into a martingale process.

- The first period when the string has been (completely) typed will be a stopping time for the lottery.

- Therefore, we can try to use Doob's Optional-Stopping Theorem to establish an equality involving the expectation of gains and the expectation of losses.

- Use the expectation of gains (which we know) to find the expectation of losses (which we don't know, and want to find).

- But the expectation of losses will be constructed in such a way as to be exactly the expected time for the monkey to type the word!

2.2 Creating a lottery

For simplification, consider first an alphabet of 26 upper-case letters. The monkey hits each of these keys independently and with the same probability. At each period, it hits only one key. Now, we construct a lottery and a strategy to transform the problem into a problem of measure theory. Suppose that a gambler j arrives at each period $j \in \mathbb{N}$ and pays \$1 to play the following game (with the following strategy):

- Agent j bets that the monkey will type `A` at j. If the monkey does, j wins 26\$[1]. If the monkey does not, j wins nothing and quits the game.

- If j was succesful at $t = j$, j uses his previous earnings (26\$) to bet again that the letter typed at $t = j + 1$ will be `B`. If it is `B`, j earns 26^2\$ and continues in this fashion in the following periods, betting in the subsequent characters in `ABRACADABRA` using all his previous gains. If he is not successful, he earns zero and quits the game.

- If `ABRACADABRA` is typed, the lottery ends.

- **Important:** In the case of failure for gambler j, since he had bet everything he won previously, j quits the game "empty-handed". This means that j has a net loss of 1\$

[1] This construction follows from the fact that there are 26 letters, and that we wish to construct a fair game to create an equality between expected gains and expected losses.

(since he initially paid 1\$ to join the game, and leaves with 0\$ gains). Thus, the only possibility for the gambler to leave the game with positive net gains is if he got at least one letter right, never got a letter wrong, and `ABRACADABRA` has been typed by the monkey.

2.3 Constructing a martingale

Define $(\Omega, \mathcal{F}, (\mathcal{F}_n)_{n\in\mathbb{Z}^+}, \mathbb{P})$ as our filtered space[2]. Let S denote the string `ABRACADABRA`. We denote by $\#S = 11$ its lenght, and S_l the l-th letter.

Let's define, as Williams [1], C_n^j as j's stake at period $n \in \mathbb{N}$. It will be given by:

$$C_n^j = \begin{cases} 0, \quad \text{if} \quad n < j \\ 1, \quad \text{if} \quad n = j \\ 26^k, \quad \text{if } j \text{ successfully guessed the first } k \text{ letters } S_1, ..., S_k, \text{ with } n = j + k \\ 0, \quad \text{otherwise} \end{cases}$$

We can think of the stake as the amount the gambler j has available for betting at period n, *prior* to making his bet at n. Whenever he wins, his stakes for the next period increase: the gambler always adds all previous gains to the stake, betting everything he has!

Then the process C^j is previsible for all j, since its value depends on events contained on the sigma-algebra of period $n-1$, i.e.,

$$\forall n \in \mathbb{N}, C_n^j \in m\mathcal{F}_{n-1}.$$

In other words, we can determine C_n^j with information available at n (i.e., how did j perform in all *past* periods).

Now let X_n^j denote the net lottery payoff for j up to (and including) period n:

$$X_n^j = \begin{cases} 0, \quad \text{if} \quad n \leq j \\ C_{n+1}^j - 1, \quad \text{otherwise} \end{cases}$$

Now I will show X_n^j is in fact a martingale. Since $\forall n \in \mathbb{N}, C_{n+1}^j \in m\mathcal{F}_n$, then $\forall n \in \mathbb{Z}^+, X_n^j \in m\mathcal{F}_n$, which means $(X_n^j)_{n\in\mathbb{Z}^+}$ is adapted (i). Besides, $\forall n, j, \quad \mathbb{E}(|X_n^j|) < \infty$ (ii). To see this, notice that X_n^j is strictly increasing on the number of correct characters j guesses, which in turn

[2] I denote by $\mathbb{Z}^+$ the set of non-negative integers.

is bounded above by $\#S$, implying $X_n^j \leq 26^{\#S} = 26^{11} \in \mathbb{R}$. To see that $\mathbb{E}[X_n|\mathcal{F}_{n-1}] = X_{n-1}$, notice that:

$$n \leq j \implies X_n^j = X_{n-1}^j$$

$$n > j, \quad j \quad \text{typed a letter wrong at } n-1 \implies X_n^j = X_{n-1}^j$$

$$n > j, \text{typed the first } k \text{ letters correctly at } n-1 \implies$$

$$X_n^j = \begin{cases} 26^{k+1} - 1 \quad \text{with probability } 1/26 \\ -1 \quad \text{with probability } 25/26 \end{cases} \implies \mathbb{E}[X_n^j|\mathcal{F}_n] = 26^k - 1 = X_{n-1}^j$$

In any case, we have established $\mathbb{E}[X_n^j|\mathcal{F}_{n-1}] = X_{n-1}^j$ (iii). By facts (i), (ii) and (iii), $(X_n^j)_{n\in\mathbb{Z}^+}$ is a martingale.

2.4 Doob's Optional Stopping Theorem

I will reproduce below Doob's Optional Stopping Theorem for martingales below, for convenience (taken from the class slides):

Theorem 2.1 (Doob's Optional Stopping Theorem)**.** *Let X be a martingale and T a stopping time. Then X_T is integrable and*

$$\mathbb{E}(X_T) = \mathbb{E}(X_0)$$

in each of the following situations:

(i) T is bounded;
(ii) X is bounded and T is a.s. finite;
(iii) $\mathbb{E}(T) < \infty$ and for some $K > 0$,

$$\left|X_n(\omega) - X_{n-1}(\omega)\right| \leqslant K, \quad \forall(n, \omega)$$

Define $Y_n = \sum_{j=0}^n X_n^j$. Y_n denotes **all profits** (gains minus the losses, including the entry fee) accrued to all gamblers that joined the lottery up to period n. Then $(Y_n)_{n\in\mathbb{Z}^+}$ is a martingale. Indeed, first notice that:

$$\mathbb{E}[Y_n|\mathcal{F}_{n-1}] = \mathbb{E}\left[\left(\sum_{j=1}^n X_n^j\right)|\mathcal{F}_{n-1}\right] = \sum_{j=1}^n \mathbb{E}[X_n^j|\mathcal{F}_{n-1}] = \sum_{j=1}^n X_{n-1}^j = Y_{n-1}$$

The fact that it is adapted follows from $m\mathcal{F}_n$ being a vector space, and $\mathbb{E}(|Y_n|) < \infty$ follows from $\mathcal{L}^1$ being a vector space. Thus, $(Y_n)_{n \in \mathbb{Z}^+}$ satisfies the three requirements of a martingale.

Now let T be $\inf\{t \in \mathbb{N} : \text{monkey types } \texttt{ABRACADABRA}\}$. We will apply Theorem 2.1 to Y and T by proving condition (iii) of the Theorem.

First: the fact that $\mathbb{E}(T) < \infty$ follows from William's [1] Lemma from Section 10.11:

Lemma 2.2. *Suppose that T is a stopping time such that for some N in $\mathbb{N}$ and some $\varepsilon > 0$, we have, for every n in $\mathbb{N}$:*

$$\mathbb{P}\big(T \leq n + N \mid \mathcal{F}_n\big) > \varepsilon, \qquad a.s.$$

Then $\mathrm{E}(T) < \infty$.

The probability that the monkey types ABRACADABRA between n (inclusive) and $n + (\#S) - 1 = n + 10$ is $26^{-\#S} = 26^{-11} > 0$. Thus, by taking $N = 10, \varepsilon = 26^{-11}/2$, we have:

$$\forall n \in \mathbb{N}, \quad \mathbb{P}\big(T \leq n + N \mid \mathcal{F}_n\big) > \varepsilon$$

And we conclude that $\mathbb{E}(T) < \infty$.

Now we proceed with the second condition of (iii). We show $\exists K > 0 : \forall (n, \omega), |Y_n(\omega) - Y_{n-1}(\omega)| \leq K$. To see this, notice that X_n^j (and Y_n too) is increasing and convex on the number of successful characters guessed. Thus, the variation $|Y_n - Y_{n-1}|$ is maximized when the number of subsequent correct characters guessed by all gamblers achieves its maximum, i.e., $\#S$, at n.

Thus, I claim

Proposition 2.3. $|Y_n - Y_{n-1}| \leq 26^{11} + 26^4 + 26$

To better understand the expression in Proposition 2.3, consider the event where all letters from the string were correctly typed from $n - 10$ to n. As argued earlier, by the convexity of the lottery on subsequent correct guesses, this event maximizes the variation of net gains (in particular, the maximum will be attained between $n - 1$ and n). It may be useful to define

$$G_n^j = \begin{cases} 0, \quad \text{if} \quad n \leq j \\ X_n^j + 1, \text{otherwise} \end{cases}$$

as the **gross gains** accrued to j by playing the lottery up to period n (i.e., ignoring the entry costs of 1\$). Then

$$\sum_{j=n-10}^{n} G_n^j = G_n^{n-10} + G_n^{n-3} + G_n^n = 26^{11} + 26^4 + 26 \tag{2.1}$$

represents the total gross gains from the agents that didn't quit when the full string is typed by the monkey[3]. Indeed, $j \notin \{n-10, n-3, n\}$ will guess a letter wrong at some point, while $j \in \{n-10, n-3, n\}$ will never guess a letter wrong with their strategy. This is best seen in the diagram below, which illustrates the streaks of characters associated with the gamblers who were successful:

$$\overbrace{ABRACAD\underbrace{ABR\overbrace{A}^{1}}_{4}}^{11}$$

In red are the js that get a letter wrong at some point and will have quit at n. Since the gamblers quit when they get a letter wrong, they cannot increase their gains between n and $n-1$. Their gross gains at n are zero: they lost everything they put at stake at some point.

In blue, are the js that got all letters right, and inside the brackets are the lenght of the streak they get. These will be the ones having positive gross gains at n: $j = n-10$ guesses 11 letters right, and his gross gains are 26^{11} at n; $j = n-3$ guesses 4 letters right, and his gross gains are 26^4 at n; and $j = n$ only 1 letters right, and his gross gains are 26^1 at n. This explains the expression in Equation (2.1).

Since the variation in aggregate net payoffs, $|Y_n - Y_{n-1}|$, must be bounded from above by the maximum gross gains these gamblers can get at between $n-1$ and n, Proposition 2.3 is proved. Therefore, (iii) from Theorem 2.1 is proved, and $\mathbb{E}(Y_T) = \mathbb{E}(Y_0)$, where Y_T is the stopped process associated to T.

Now we use Theorem 2.1 to conclude that, since

$$\mathbb{E}(Y_0) = 0$$

(indeed, $Y_0 = X_0^0 = 0$, since all cumulative earnings start deterministically at zero), then

$$\mathbb{E}(Y_T) = 0,$$

[3] You could also bound $|Y_n(\omega) - Y_{n-1}(\omega)|$ more easily by 11×26^{11}, the maximum gross earnings that these individuals $j = n-10, ..., n$ can get by guessing all the letters from `ABRACADABRA` right the first time, since 26^{11} is an upper bound for what any given individual can win at any period, and 11 is an upper bound for the number of 'surviving' individuals betting from $t = n-10$ to $t = n$. However, it will be useful to use the other bound $(26^{11} + 26^4 + 26)$ later on.

which means that the expected aggregate gross losses from our lottery (stopping when `ABRACADABRA` is first typed) will equal expected aggregate gross gains. We know total gross losses from the gamblers are $\mathbb{E}(T)$ (every gambler that joins the game pays 1$). By construction, the only gamblers that survive are the ones who started betting from $t = T - 10$ to $t = T$; all the others have already quit the game, and have **zero** cumulative gross earnings (-1 cumulative net earnings each). Thus, the only gamblers contributing to the expected cumulative gross earnings are the ones who started betting when the first `ABRACADABRA` appears. In particular:

- the one who starts betting when the 1st 'A' is typed ($j = T - 10$)
- the one who starts betting when the 4th 'A' is typed ($j = T - 3$)
- the one who starts betting when the 5th and last 'A' appears ($j = T$)

We already showed that, for $j = T - 10$ to $t = T$, the cumulative gross gains will be precisely $26^{11} + 26^4 + 26$. Since we must have that, on average, cumulative gross gains equal cumulative gross losses, so that the equality established in the theorem is maintained, we must have

$$\mathbb{E}(Y_T) = \mathbb{E}(26^{11} + 26^4 + 26 - T) = 0,$$

and we conclude that

$$\mathbb{E}(T) = 26^{11} + 26^4 + 26 \tag{$\star$}$$

And that is the answer to the problem posed: the expected time for the monkey to type `ABRACADABRA` will be $26^{11} + 26^4 + 26$ periods (which is equal to $3,670,344,487,444,778$).

3 A (slightly) more general solution to a (slightly) more general problem

We can go a little bit further and characterize our problem more broadly, by asking ourselves: what will be the expected time for the monkey to first type a string S of finite length containing characters from a countably finite set (alphabet) A, given that all characters can be typed with uniform probability? Notice that nothing essentially changes in our proof: we can still apply Theorem 2.1, if we slightly change our lottery to give $\#A$ dollars in case of success and zero in case of failure. The strategy remains the same: the gambler bets all his previous gross gains in case of victory, and quits, leaving empty-handed, in case of failure.

The difficulty, then, resides in finding an analytical expression for the gross gains of the gamblers that are betting (didn't quit) when the string is first typed.

3.1 Left and right slices

I will represent string S as a vector $(S_1, S_2, ..., S_{\#S})$ of characters such that

$$\forall i = 1, ..., \#S, \quad S_i \in A.$$

I define a left slice of length $n \leq \#S$ from string S (denoted by $L_n(S)$) as the **first** n characters in the string:

$$L_n(S) = (S_1, S_2, ..., S_n)$$

Similarly, I define a right slice of length $n \leq \#S$ from string S (denoted by $R_n(S)$) by the **last** n characters in the string:

$$R_n(S) = (S_{\#S-n+1}, ..., S_{\#S-1}, S_{\#S})$$

Let's say string S is first typed starting at $t = n^*$ and ending at $t = n^* + \#S - 1$. Each gambler arriving will be associated with a right slice: the one arriving at $j = n^*$ will bet (if he gets to keep playing) on $R_{\#S}(S) = S$. The one arriving at $j = n^* + 1$ will bet (if he gets to keep playing) on $R_{\#S-1}(S) = (S_2, S_3, ..., S_{\#S})$, and so on. Notice I stressed *if he gets to keep playing*: he will only bet on every character from the right slice (i.e., he will only maintain positive gross earnings at the terminal date) **if** that right slice is also a left slice; otherwise, he will necessarily make a wrong bet at some period, and quit the game (having 0 gross earnings at the terminal date).

To see this is true for all gamblers, note that it is immediate for the first one, arriving at $j = n^*$, since $R_{\#S}(S) = L_{\#S}(S) = S$. Gambler $j = n^* + k$, (with $k > 0, j \leq n^* + \#S - 1$) will only bet on every character if $S_{k+1} = S_1, S_{k+2} = S_2, ..., S_{\#S} = S_{\#S-k}$, which is precisely equivalent to

$$L_{\#S-k}(S) = (S_1, S_2, ..., S_{\#S-k}) = (S_{k+1}, S_{k+2}, ..., S_{\#S}) = R_{\#S-k}(S).$$

For convenience, let $m(j) = \#S - k$, map the length of the right slice associated with the gambler arriving at $j = n^* + k$. Each gambler arriving at $j = n^*, ..., n^* + \#S - 1$ will have positive gross gains at the terminal period $t = n^* + \#S - 1$ **iff** $L_{m(j)}(S) = R_{m(j)}(S)$. In this case, the gross gains will be equal to $\#A^{m(j)}$ Therefore, the expected time for the monkey to type S using A will be:

$$\sum_{j=n^*}^{n^*+\#S-1} (\#A)^{m(j)} \mathbf{1}\left\{R_{m(j)}(S) = L_{m(j)}(S)\right\} \tag{3.1}$$

where $\mathbf{1}\left\{R_i(S) = L_i(S)\right\} = 1$ if $R_i(S) = L_i(S)$, 0 otherwise. Notice that this agrees with our ABRACADABRA example:

- $R_1(S) = L_1(S) = (A)$ (ABRACADABRA)
- $R_4(S) = L_4(S) = (A, B, R, A)$ (ABRACADABRA)
- $R_{11}(S) = L_{11}(S) = (A, B, R, A, C, A, D, A, B, R, A)$ (ABRACADABRA)

For $i \notin \{1, 4, 11\}, R_i(S) \neq L_i(S)$. Thus, our expression is, since $\#A = 26$:

$$\sum_{i=1}^{\#S} (\#A)^i \mathbf{1}\left\{R_i(S) = L_i(S)\right\} = \sum_{i \in \{1,4,11\}} 26^i = 26^{11} + 26^4 + 26^1$$

the same as the one in $(\star)$.

3.2 A counterintuitive result

The following ideas and dicussion lead, then, to the following counterintuitive result[4]: two words of the **same length** may take **different** expected average times to be typed by the monkey! Indeed, given two strings of the same length ℓ, the expected time to type any one of them will be greater the greater (everything else constant):

1. The number of left slices of the string that are also right slices of the string ($\#\{i = 1, .., \#S : L_i(S) = R_i(S)\}$). This is maximized when the string consists of a single repeating character, so that every left slice is also a right slice.
2. The lenght of the left slices that are also right slices.

In particular, AA takes longer, on average, than AB to be typed using an alphabet of only upper-case letters. It takes even longer if we use an alphabet of upper and lower-case letters (the difference is increasing in $\#A$, everything else constant).

[4] Or, at least, it was quite counterintuitive to me!

3.3 A computational exercise

To illustrate our previous conclusions, I run 300,000 Monte Carlo simulations with strings. The program draws random letters and counts the time until the string is typed. I do the same exercise separately for strings `AA` and `AB`. The histograms of these distributions in Figure 1 show that, visually, they are quite similar (indiscernible even). However, the moments in Table 1 reveal that, not only is the mean time taken to type `AB` lower than the one to type `AA`, all quartiles are also lower. The expected times from expression (3.1) give $26^2 + 26 = 702$ for `AA` and $26^2 = 676$ for `AB`, which are pretty close to the results from the simulations (704 and 674, respectively).

	AA	*AB*
mean	704	674
std	703	673
min	2	2
25%	203	195
50%	490	467
75%	977	933
max	9228	8733

Table 1: Moments from the Monte Carlo simulations, rounded to the next integer. 300,000 simulations where used.

3.4 Intuition

Now that we have proved the result formally, using measure theory, and illustrated it with simulations, it may still appear counterintuitive that two strings of the same length may take different expected times to be typed. In order to get some intuition, let's return to the simple examples of string $S_1 = AB$ *versus* string $S_2 = AA$, and ask ourselves what is the probability that the monkey typed each of them with 4 keystrokes or less. Let "-" denote the monkey typing any letter. The situations in which this happens will be:

$$\text{For AB:} \quad \begin{bmatrix} A & B & - & - \\ - & A & B & - \\ - & - & A & B \end{bmatrix}$$

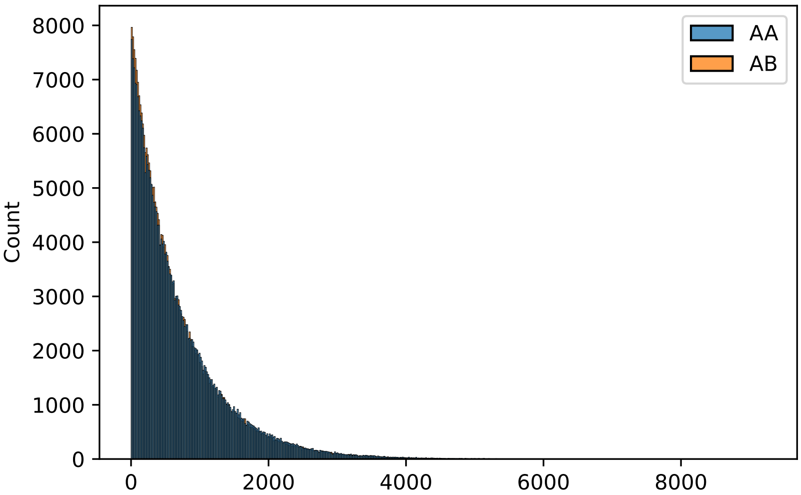

Figure 1: Histogram of Monte Carlo simulations. Notice the distributions are visually overlapped.

$$\text{For AA:} \quad \begin{bmatrix} A & A & - & - \\ - & A & A & - \\ - & - & A & A \end{bmatrix}$$

For AB, there are no overlaps in the strings; therefore, the probability of each event (each row) occurring is $1/26^2$, and the probability of the union of these three events is $3/26^2$, since they are mutually exclusive. The same does not happen, however, for AA: for instance, all of the rows contain the sequence (A, A, A, A). Therefore, we can't simply add the probabilities of these three events: we must subtract the probability of (A, A, A, A) happening. There are other overlaps in this example; for instance, rows 1 and 2 contain the sequence $(A, A, A, -)$, and rows 2 and 3 contain the sequence $(-, A, A, A)$.

After subtracting these overlaps (which will happen with strictly positive probability) appropriately, from the probability of the union of the 3 rows, we end up with a probability strictly lower than $3/26^2$. Thus, we conclude that the probability of the monkey typing AB in 4 keystrokes or less is higher than that of it typing AA in 4 keystrokes or less, providing some intuition for our main result.

4 Conclusion

In conclusion, we applied our results of measure theory and martingales to solve a seemingly complicated problem of combinatorics[5] that at the first glance seems to have nothing to do with martingales.

By noticing that the previous steps of our proof could be extended to any countably finite string S and any countably finite alphabet A, and that only remaining issue was limited to finding an expression for the gains of the last gamblers in the lottery (the ones who didn't quit), we were able to generalize an analytical expression for the expected time of any such string and any such alphabet[6] to be typed by the monkey.

We applied this expression to arrive at the counterintuitive result that strings of the same cardinality may take different expected times to be typed by our monkey. Finally, we illustrated this conclusion with computer simulations and provided a simple example that provided intuition for this seemingly counterintuitive result.

References

[1] D. Williams. *Probability with Martingales.* Cambridge University Press, feb 1991. doi: 10.1017/cbo9780511813658.

[5] If you are curious: for a solution with combinatorics, see this discussion.

[6] Given that the monkey still has independent and uniform probability of hitting each key.