coin_toss_sequences

June 14, 2022

1 Analysis of sequences of coin tosses

1.1 Background

The origin of this notebook comes from the paper Perception of Randomness: Subjective Probability of Alternation, which makes some claims about probabilities and expectations regarding sequences of coin tosses. In particular, they discuss (but do not prove) the difference between tossing coins until the pattern THH (tail-head-head) has been observed, and tossing coins until HHH (head-head-head) has been observed. For THH, they claim 8 throws are needed on average, whereas 14 throws are needed on average for HHH. Furthermore, they claim that if a single sequence is halted when either THH or HHH is first observed, then a sequence ending in THH is 7 times more likely than one ending in HHH.

In this notebook, we seek to prove (or disprove) these claims, using theory wherever possible but also numerical computations where this becomes too difficult (such as this author not being clever enough!).

1.2 Experimental Setup

1.2.1 Experimenter X

Consider an experimenter X who is inside a locked room performing a sequence of independent tosses of an unbiased coin. Suppose that the coin tosses continue until the first occurrence of the subsequence (or pattern) S occurs, whereupon the experiment halts. Let $\texttt{H}$ be the predicate that the experiment has halted, and let N be a variable representing the total number of trials (coin tosses) performed so far during the experiment.

1.2.2 Observer Y

Suppose that sitting outside of the locked room is observer Y. We further suppose that observer Y knows all about the experiment being performed, including S, but does not observe the outcome of any of the trials. However, after each coin toss, experimenter X calls out the number n of trials performed so far, which observer Y hears through a small grill. Thus, observer Y does not know if the experiment has halted or not, but does know that if $N = n$ trials have been performed, then the experiment cannot have halted before then.

Consequently, observer Y computes the probability of the experiment halting at the nth trial as

$$\mathbb{P}(\texttt{H} \mid N = n, S, Y) \quad = \quad \mathbb{P}(t_{n-m+1} \dots t_n = S \mid N = n) = \frac{1}{2^m}\,, \tag{1}$$

where $m = |S|$ is the length of pattern S, and $t_k \in \{T, H\}$ represents the outcome of the kth trial. Clearly, this only holds for $n \geq |S|$, and thus $\mathbb{P}(\mathtt{H} \mid N = n, S, Y) = 0$ for $n < |S|$.

We shall show in a later section below that if the halting pattern is simply $S =$T (or H), then the sequence follows a geometric distribution, such that the probability $p_n \doteq \mathbb{P}(N = n \mid \mathtt{H}, S)$ of the experiment halting with a length-n sequence is $p_n = p^{n-1}q$. Here we have observed that the probability q of halting the sequence is the constant $q = \frac{1}{2^m}$ (with $p = 1 - q$), and hence we might suppose that every experiment, for arbitrary S, is a geometric sequence of independent Bernoulli trials.

However, do not be fooled. The decision of experimenter X to halt the experiment is not a stochastic one, independent of previous trials, but is in fact purely deterministic, depending entirely on the observed sequence $t_1 \ldots t_n$ of outcomes and the halting pattern S.

1.2.3 Observer Z

Now suppose that both experimenter X and observer Y are inside a locked building. Outside of this building waits observer Z, who also knows the nature of the experiment. Only when the experiment is halted does experimenter X unlock and leave the inner room (passing by observer Y), and then unlock and open the outer door of the building, telling observer Z the number N of trials performed. Consequently, observer Z knows that the experiment has halted when told the number n of trials, such that $\mathbb{P}(\mathtt{H} \mid N = n, S, Z) = 1$ for $n \geq |S|$, with $\mathbb{P}(\mathtt{H} \mid N = n, S, Z) = 0$ for $n < |S|$.

Note that during the experiment, observer Y has access to different information than observer Z, and thus computes different probability estimates. However, once experiment X has told observer Z the total number N of trials, then both observers Y and Z have the same information. Consequently, we shall drop the explicit dependence of the observer when it makes no difference. If in doubt, the reader should adopt the viewpoint of observer Z.

Whilst waiting for the experiment to finish, observer Z plays a game of trying to estimate the number of trials. Upon hearing the outer door being unlocked, but before knowing the answer, observer Z computes $\mathbb{P}(N \mid \mathtt{H}, S)$ and thus $\mathbb{E}[N \mid \mathtt{H}, S]$.

1.3 Theoretical Support

The theoretical analyses of later sections will involve counting the number c_n of distinct length-n sequences in which the pattern S occurs (one or more times), out of all 2^n length-n sequences of coin tosses (trials). It will become apparent that the Fibonacci numbers F_n play a significant role in this counting. Hence, we digress here to derive some useful information about the Fibonacci sequence.

The Fibonacci numbers satisfy the recurrence relation

$$F_n \quad = \quad F_{n-1} + F_{n-2} \tag{2}$$

for $n \geq 2$, with boundary conditions $F_0 = 0$ and $F_1 = 1$. Thus, the first few numbers in the sequence are $0, 1, 1, 2, 3, 5, 8, \ldots$.

It turns out that there is a closed formula for computing the nth Fibonacci number F_n, for $n \geq 0$. We deduce this by substituting the supposed solution $F_n = \lambda^n$ into the recurrence relation to obtain

the so-called characteristic equation. Hence,

$$\begin{aligned} F_n &= F_{n-1} + F_{n-2} && (3)\\ \Rightarrow \lambda^n &= \lambda^{n-1} + \lambda^{n-2} && (4)\\ \Rightarrow 0 &= \lambda^2 - \lambda - 1 && (5)\\ \Rightarrow \lambda &= \frac{1 \pm \sqrt{5}}{2}\,. && (6) \end{aligned}$$

The positive root is the Golden Ratio ϕ, and the negative root is $\bar{\phi} = -\frac{1}{\phi}$. Hence, the general solution is

$$F_n = \alpha\phi^n + \beta\bar{\phi}^n\,, \tag{7}$$

for coefficients α and β to be determined. The appropriate boundary conditions are $F_0 = 0$ and $F_1 = 1$, such that

$$\begin{aligned} F_0 = \alpha + \beta = 0 \quad &\Rightarrow \quad \beta = -\alpha\,, && (8)\\ F_1 = \alpha(\phi - \bar{\phi}) = 1 \quad &\Rightarrow \quad \alpha = \frac{1}{\sqrt{5}}\,. && (9) \end{aligned}$$

Hence, the closed form of the Fibonacci numbers is given by

$$F_n = \frac{1}{\sqrt{5}}\left\{\left(\frac{1+\sqrt{5}}{2}\right)^n - \left(\frac{1-\sqrt{5}}{2}\right)^n\right\}\,. \tag{10}$$

[1]:
```
from math import sqrt

_root5 = sqrt(5)
_phi = (1 + _root5) / 2
_phi_bar = (1 - _root5)/2

def fibonacci(n):
    rval = (_phi**n - _phi_bar**n) / _root5
    ival = int(rval)
    return (
        # Allow for numerical error:
        ival if rval - ival < 0.5  # real = integer + epsilon
        else ival + 1              # real = integer - espilon
    )

print(list(fibonacci(n) for n in range(21)))
```

```
[0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584,
4181, 6765]
```

[2]:
```
def recursive_fibonacci(n):
    if n == 0:
        return 0
```

```
    elif n == 1:
        return 1
    else:
        return recursive_fibonacci(n-1) + recursive_fibonacci(n-2)

print(list(recursive_fibonacci(n) for n in range(21)))
```

```
[0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584,
4181, 6765]
```

Another property of Fibonacci numbers that we shall need is that

$$\sum_{j=1}^{n} F_j \quad = \quad F_{n+2} - 1\,. \tag{11}$$

Observe that this is true for $n = 0$, since (by definition) the sum on the left-hand side is zero, and the right-hand side gives $F_2 - 1 = 1 - 1 = 0$. Hence, by induction, we have

$$\sum_{j=1}^{n+1} F_j \quad = \quad \sum_{j=1}^{n} F_j + F_{n+1} \tag{12}$$

$$= \quad F_{n+2} - 1 + F_{n+1} = F_{n+3} - 1\,, \tag{13}$$

using the Fibonacci recurrence relation above.

1.4 Computational Support

Since we are considering sequences of independent, unbiased coin tosses, it will be useful to provide some machinery for both deterministically and stochastically creating such sequences. This machinery will allow us to computationally check our theoretical results, or to bypass theoretical results if they are intractable to prove.

[3]:
```
def binary_sequence(value, N=None):
    """
    Converts the decimal value into a size-N vector of bits, using zero-padding
    or truncation (both on the left) as necessary.

    Inputs:
        - value (int): The decimal value.
        - N (int, optional): The size of the binary vector.
    Returns:
        - vec (array): The size-N binary vector.
    """
    b = bin(value)[2:]
    if N is not None:
        b = b[-N:]
        if len(b) < N:
            b = "0" * (N - len(b)) + b
    return b
```

```
def deterministic_sequence(value, n):
    """
    Computes the distinct, length-n coin toss sequence
    corresponding to the given decimal value.
    """
    seq = binary_sequence(value, n)
    return seq.replace('0', 'T').replace('1', 'H')


def deterministic_sequences(n):
    """
    Computes all distinct, length-n sequences of coin tosses.
    """
    return list(
        deterministic_sequence(v, n)
        for v in range(2**n)
    )
```

[4]:
```
print(deterministic_sequences(2))
print(deterministic_sequences(3))
print(deterministic_sequences(4))
```

```
['TT', 'TH', 'HT', 'HH']
['TTT', 'TTH', 'THT', 'THH', 'HTT', 'HTH', 'HHT', 'HHH']
['TTTT', 'TTTH', 'TTHT', 'TTHH', 'THTT', 'THTH', 'THHT', 'THHH', 'HTTT', 'HTTH',
'HTHT', 'HTHH', 'HHTT', 'HHTH', 'HHHT', 'HHHH']
```

[5]:
```
def count_pattern(n, pattern):
    """
    Counts the number of distinct, length-n sequences of coin tosses
    that contain the given pattern.
    """
    f = lambda s: 1 if s.find(pattern) >= 0 else 0
    return sum(f(s) for s in deterministic_sequences(n))
```

[6]:
```
print(list(count_pattern(n, "TH") for n in range(21)))
print(list(count_pattern(n, "THH") for n in range(21)))
```

```
[0, 0, 1, 4, 11, 26, 57, 120, 247, 502, 1013, 2036, 4083, 8178, 16369, 32752,
65519, 131054, 262125, 524268, 1048555]
[0, 0, 0, 1, 4, 12, 31, 74, 168, 369, 792, 1672, 3487, 7206, 14788, 30185,
61356, 124308, 251199, 506578, 1019920]
```

[7]:
```
from random import random
```

```
def stochastic_sequence(pattern):
    """
    Performs one experiment of repeated coin tosses that halts
    when the given pattern is first observed.
    Returns the observed sequence.
    """
    seq = ""
    while True:
        throw = 'T' if random() < 0.5 else 'H'
        seq += throw
        if seq.endswith(pattern):
            return seq


def length_stochastic_sequence(pattern):
    """
    Performs one experiment of repeated coin tosses that halts
    when the given pattern is first observed.
    Returns the length of the observed sequence.
    """
    seq = 'X' * (len(pattern) - 1)
    n = 0
    while True:
        n += 1
        throw = 'T' if random() < 0.5 else 'H'
        seq += throw
        if seq == pattern:
            return n
        seq = seq[1:]
```

[8]:
```
print(stochastic_sequence("THH"))
print(stochastic_sequence("THH"))
print(stochastic_sequence("THH"))
```

```
HTTHTTHTHH
HHHHTHTHTTTTHTHH
HHHHTTHH
```

[9]:
```
print(length_stochastic_sequence("THH"))
print(length_stochastic_sequence("THH"))
print(length_stochastic_sequence("THH"))
```

```
6
13
6
```

[10]:
```
def simulate_experiment(num_experiments, pattern):
    """
    Simulates the experiment the specified number of times, and returns the list
    of observed sequence lengths.
    """
    return list(
        length_stochastic_sequence(pattern)
        for i in range(num_experiments)
    )
```

[11]:
```
import numpy as np

print(np.mean(simulate_experiment(1000, "T")))
print(np.mean(simulate_experiment(10_000, "TH")))
print(np.mean(simulate_experiment(100_000, "THH")))
```

```
2.037
4.0248
8.00481
```

[12]:
```
def index_length_stochastic_sequence(patterns):
    """
    Performs a modified experiment that halts when any one of the given
    patterns is first observed. Returns the index of the matching pattern
    and the length of the observed sequence.
    """
    max_len = max(len(p) for p in patterns)
    seq = 'X' * (max_len - 1)
    n = 0
    while True:
        n += 1
        throw = 'T' if random() < 0.5 else 'H'
        seq += throw
        for i, pattern in enumerate(patterns):
            if seq.endswith(pattern):
                return i, n
        seq = seq[1:]
```

[13]:
```
patterns = deterministic_sequences(3)
print(index_length_stochastic_sequence(patterns))
print(index_length_stochastic_sequence(patterns))
print(index_length_stochastic_sequence(patterns))
```

```
(4, 3)
(5, 3)
(2, 3)
```

[14]:
```
def simulate_combined_experiment(num_experiments, patterns):
    """
    Simulates the modified experiment the specified number of times.
    The modified experiment halts when any one of the given patterns
    is first observed.
    Returns a list of lists of observed sequence lengths,
    one for each pattern.
    """
    lists = list(list() for p in patterns)
    for i in range(num_experiments):
        j, n = index_length_stochastic_sequence(patterns)
        lists[j].append(n)
    return lists
```

[15]:
```
patterns = deterministic_sequences(2)
res = simulate_combined_experiment(10_000, patterns)
for p, r in zip(patterns, res):
    print("%s: %d" % (p, len(r)))
```

```
TT: 2493
TH: 2476
HT: 2588
HH: 2443
```

1.5 Experiment 1 - T vs H

The probability of throwing a T is $\mathbb{P}(T) = \frac{1}{2}$, and so we might expect it to take 2 throws on average to obtain a sequence halting with T. Similarly for H.

[16]:
```
print(np.mean(simulate_experiment(1000, "T")))
print(np.mean(simulate_experiment(1000, "H")))
```

```
1.988
1.989
```

This is in fact what we observe.

More formally, suppose on a single coin toss we throw a tail with probability $p_T = \mathbb{P}(T)$ or a head with probability $p_H = \mathbb{P}(H)$. Then a length-n sequence that halts with the first T has probability

$$\mathbb{P}(N = n \mid \mathtt{H}, S = T) \quad = \quad p_H^{n-1} p_T \,, \tag{14}$$

where

$$\sum_{n=1}^{\infty} p_H^{n-1} p_T \quad = \quad (1 + p_H + p_H^2 + \cdots)\, p_T = \frac{p_T}{1 - p_H} = 1 \,. \tag{15}$$

The expectation is therefore given by

$$\mathbb{E}[N = n \mid \mathtt{H}, S = T] = \sum_{n=1}^{\infty} n p_H^{n-1} p_T = p_T \frac{\partial}{\partial p_H} \sum_{n=0}^{\infty} p_H^n \tag{16}$$

$$= p_T \frac{\partial}{\partial p_H} \frac{1}{1-p_H} = \frac{p_T}{(1-p_H)^2} = \frac{1}{p_T}\,. \tag{17}$$

We conclude for the simple halting pattern S =T that the distribution of permissible sequence lengths is geometric.

1.6 Experiment 2 - TT vs TH

Suppose now we run two simultaneous experiments, with the first experiment halting on S =TT, and the second halting on S =TH.

[17]:
```
patterns = deterministic_sequences(2)
for pattern in patterns:
    mu = np.mean(simulate_experiment(10_000, pattern))
    print("%s: %f" % (pattern, mu))
```

```
TT: 5.996300
TH: 3.995800
HT: 3.978500
HH: 6.002800
```

We now observe an interesting result. The probability of obtaining TT on two consecutive trials is $\frac{1}{2^2}$, as is that of obtaining TH. However, experimentally it takes 6 throws on average to first obtain TT, whereas it only takes 4 throws on average to obtain TH. By symmetry, we expect HT to behave like TH, and HH to behave like TT, and we observe that is indeed the case.

Alternatively, suppose we run a modified experiment that halts on first observing either TH or TT.

[18]:
```
half_patterns = patterns[0 : (len(patterns)//2)]
res = simulate_combined_experiment(10_000, half_patterns)
for p, r in zip(half_patterns, res):
    print("%s: %d %f" % (p, len(r), np.mean(r)))
```

```
TT: 4965 3.004632
TH: 5035 3.021450
```

We observe that number of sequences halting with TH compared to those halting with TT are in about the ratio 1:1. However, given the above results, we might have anticipated the ratio 3:2 of TH to TT. I find the discrepancy between observation and "intuition" to be fascinating.

To explain the results of the modified experiment, we note that the first coin toss will either be T with probability 0.5 or H with probability 0.5. If it is a T, then a second throw will guarantee a sequence that ends in either TT or TH. However, if it is an H, then we have effectively reset the experiment except for having 1 extra throw. Thus, if we let $\bar{N} \doteq \mathbb{P}(N \mid \mathtt{H}, S = T*)$, where $*$ here is

a wildcard denoting either T or H, we have

$$\bar{N} = \frac{1}{2}(1+1) + \frac{1}{2}(1+\bar{N}) = \frac{3}{2} + \frac{1}{2}\bar{N} \tag{18}$$

$$\Rightarrow \bar{N} = 3\,. \tag{19}$$

Equivalently, if it takes 2 throws on average to first obtain T (from the previous section), then it will take a third throw to obtain TH or TT. Also, once we have thrown the first T, the two possible endings of TT or TH are equally likely.

1.6.1 Halting with $S =$TH

To explain the two simultaneous experiments, the situation is less straightforward. Let us first consider the pattern $S =$TH. For the experiment to return a sequence of length n, the sequence must end in TH and not contain TH within the length-$(n-1)$ pre-sequence. However, since the pattern TH cannot overlap with itself, then the length-$(n-1)$ pre-sequence, $t_1 t_2 \ldots t_{n-1}$, cannot end with $t_{n-2}t_{n-1} =$TH since we require $t_{n-1}t_n =$TH, and hence we need only consider the length-$(n-2)$ pre-sequence, $t_1 t_2 \ldots t_{n-2}$. Letting c_n be the number of distinct length-n sequences that contain the pattern TH, we have

$$\begin{aligned}
\mathbb{P}(N = n \mid \mathtt{H}, S = TH) &= \mathbb{P}(t_{n-1}t_n = TH \wedge TH \notin t_1 \ldots t_{n-1}) && (20)\\
&= \mathbb{P}(t_{n-1}t_n = TH)\,\mathbb{P}(TH \notin t_1 \ldots t_{n-1} \mid t_{n-1}t_n = TH) && (21)\\
&= \mathbb{P}(t_{n-1}t_n = TH)\,\left[1 - \mathbb{P}(TH \in t_1 \ldots t_{n-2})\right] && (22)\\
&= \frac{1}{2^2}\left[1 - \frac{c_{n-2}}{2^{n-2}}\right]. && (23)
\end{aligned}$$

More neatly, if we further define $\bar{c}_n \doteq 2^n - c_n$ to be the number of distinct length-n sequences that do **not** contain the pattern S, then we have

$$p_n = \frac{\bar{c}_{n-2}}{2^n}\,. \tag{24}$$

To find c_n, consider the complete length-n sequence, denoted by $\langle t_1 t_2 \ldots t_n \rangle$. We suppose at the outset that n is sufficiently large enough for the following analysis.

Now suppose that the sequence ends with T or H, denoted by $T\rangle$ or $H\rangle$, respectively. If $t_n = T$, then $t_{n-1}t_n \neq TH$, and so TH can only possibly occur in the length-$(n-1)$ pre-sequence, in c_{n-1} ways. However, if $t_n = H$, then it is possible that $t_{n-1}t_n = TH$ or that TH occurs earlier in the sequence. Thus, we define $c_n^{H\rangle}$ to be the number of distinct length-n sequences ending with H that contain the pattern S. Therefore, we obtain the recursion

$$c_n = c_n^{H\rangle} + c_{n-1}\,, \tag{25}$$

with boundary conditions $c_0 = c_1 = 0$ by construction. We may unwind this recursion to obtain

$$c_n = \sum_{j=2}^{n} c_j^{H\rangle}\,. \tag{26}$$

To find $c_n^{H\rangle}$, we see that either $t_{n-1}t_n = TH$ or $t_{n-1}t_n = HH$. For the former case, there are 2^{n-2} distinct length-n sequences of the form $\langle t_1 t_2 \ldots t_{n-2} TH \rangle$. For the latter case, the length-$(n-1)$

pre-sequence $\langle t_1 t_2 \ldots t_{n-2} H)$ could contain TH in $c_{n-1}^{H\rangle}$ ways. Consequently, we obtain the further recursion

$$c_n^{H\rangle} = 2^{n-2} + c_{n-1}^{H\rangle}\,, \tag{27}$$

with $c_0^{H\rangle} = c_1^{H\rangle} = 0$ by construction. Unwinding this recursion gives

$$c_n^{H\rangle} = \sum_{k=0}^{n-2} 2^k = 2^{n-1} - 1\,. \tag{28}$$

Combining the two results gives

$$c_n = \sum_{j=2}^{n} \left(2^{j-1} - 1\right) = 2^n - n - 1\,, \tag{29}$$

$$\Rightarrow \bar{c}_n = 2^n - c_n = n + 1\,, \tag{30}$$

both of which are valid for $n \geq 0$. Finally, we obtain the probability of the experiment returning a length-n sequence as

$$p_n = \frac{\bar{c}_{n-2}}{2^n} = \frac{n-1}{2^n}\,, \tag{31}$$

which is valid for $n \geq 1$.

Clearly, we do **not** have a geometric distribution, as we anticipated earlier.

A sanity check is in order here - do these supposed probabilities sum to unity? Letting $p = \frac{1}{2}$ for convenience, we see that

$$\sum_{n=1}^{\infty} p_n = \sum_{n=1}^{\infty} (n-1)p^n \tag{32}$$

$$= p^2 \frac{\partial}{\partial p} \sum_{n=1}^{\infty} p^{n-1} = p^2 \frac{\partial}{\partial p} \frac{1}{1-p} \tag{33}$$

$$= \frac{p^2}{(1-p)^2} = 1\,, \tag{34}$$

since $p = \frac{1}{2}$. Thus, we may proceed to finding the expected value $\bar{N} = \mathbb{E}[N \mid \mathtt{H}, S = TH]$ as

$$\bar{N} = \sum_{n=1}^{\infty} n p_n = \sum_{n=1}^{\infty} n(n-1)p^n \tag{35}$$

$$= p^2 \frac{\partial^2}{\partial p^2} \sum_{n=2}^{\infty} p^{n-2} = p^2 \frac{\partial^2}{\partial p^2} \frac{1}{1-p} \tag{36}$$

$$= \frac{2p^2}{(1-p)^3} = 4\,, \tag{37}$$

as expected from the simulation results above.

Let us further check these theoretical solutions by computation.

[19]:
```
def c_n(n):
    return 2**n - n - 1


theoretical_c_n_list = list(c_n(n) for n in range(21))
computed_c_n_list = list(count_pattern(n, "TH") for n in range(21))
print(computed_c_n_list)
for observed, expected in zip(computed_c_n_list, theoretical_c_n_list):
    assert observed == expected
```

```
[0, 0, 1, 4, 11, 26, 57, 120, 247, 502, 1013, 2036, 4083, 8178, 16369, 32752,
65519, 131054, 262125, 524268, 1048555]
```

We observe that brute force counting agrees exactly with the theoretical counts.

[20]:
```
def p_n(n):
    return (
        0 if n <= 0
        else (n - 1) / 2**n
    )


theoretical_p_n_list = list(p_n(n) for n in range(21))
print(theoretical_p_n_list)
```

```
[0, 0.0, 0.25, 0.25, 0.1875, 0.125, 0.078125, 0.046875, 0.02734375, 0.015625,
0.0087890625, 0.0048828125, 0.002685546875, 0.00146484375, 0.00079345703125,
0.00042724609375, 0.0002288818359375, 0.0001220703125, 6.4849853515625e-05,
3.4332275390625e-05, 1.811981201171875e-05]
```

[21]:
```
res = simulate_experiment(100_000, "TH")
lens, counts = np.unique(res, return_counts=True)
probs = counts / np.sum(counts)
```

[22]:
```
print(lens)
```

```
[ 2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23]
```

[23]:
```
print(probs)
```

```
[2.4923e-01 2.4692e-01 1.8807e-01 1.2596e-01 7.8990e-02 4.8000e-02
 2.6990e-02 1.5820e-02 9.1000e-03 5.1200e-03 2.5000e-03 1.5300e-03
 7.0000e-04 4.9000e-04 2.2000e-04 1.6000e-04 1.0000e-04 3.0000e-05
 3.0000e-05 1.0000e-05 1.0000e-05 2.0000e-05]
```

We observe that the simulated probabilites agree very well numerically with the theoretical probabilities.

1.6.2 Halting with $S =$TT

Now consider the other experiment for which the sequence of coin tosses halts when $S =$TT is first observed. Following the reasoning above, we have

$$\begin{aligned}
\mathbb{P}(N = n \mid \mathtt{H}, S = TT) &= \mathbb{P}(t_{n-1}t_n = TT \wedge TT \notin t_1 \dots t_{n-1}) && (38)\\
&= \mathbb{P}(t_{n-1}t_n = TT)\, \mathbb{P}(TT \notin t_1 \dots t_{n-1} \mid t_{n-1}t_n = TT) && (39)\\
&= \mathbb{P}(t_{n-1}t_n = TT)\, [1 - \mathbb{P}(TT \in t_1 \dots t_{n-2}T)] && (40)\\
&= \frac{1}{2^2}\left[1 - \frac{c_{n-1}^{T\rangle}}{2^{n-2}}\right] = \frac{\bar{c}_{n-1}^{T\rangle}}{2^n}\,, && (41)
\end{aligned}$$

where $c_n^{T\rangle}$ is the number of distinct length-n sequences ending in T that contain $S =$TT, and $\bar{c}_n^{T\rangle} = 2^{n-1} - c_n^{T\rangle}$ is the number of length-n sequences ending in T that do not contain $S =$TT.

Once again we assume in advance that n is large enough for our purposes. Thus, a length-n sequence ending in T either ends with TT or HT. If it ends with TT then there are 2^{n-2} such sequences. However, if it ends with HT then TT can occur in the length-$(n-2)$ pre-sequence in c_{n-2} ways, where we have now (perhaps confusingly) repurposed c_n to count sequences containing TT (rather than TH as before). Therefore, we obtain

$$c_n^{T\rangle} = 2^{n-2} + c_{n-2}\,, \tag{42}$$

with $c_0^{T\rangle} = c_1^{T\rangle} = 0$ as boundary conditions.

For c_n, the length-n sequence either ends with T or H, giving

$$\begin{aligned}
c_n &= c_n^{T\rangle} + c_{n-1} && (43)\\
&= 2^{n-2} + c_{n-1} + c_{n-2}\,, && (44)
\end{aligned}$$

with $c_0 = c_1$ by construction.

Unwinding this sequence manually, we observe that

$$\begin{aligned}
c_2 &= 2^0 + c_1 + c_0 = 2^0\,, && (45)\\
c_3 &= 2^1 + c_2 + c_1 = 2^1 + 2^0\,, && (46)\\
c_4 &= 2^2 + c_3 + c_2 && (47)\\
&= 2^2 + (2^1 + 2^0) + (2^0) = 2^2 + 2^1 + 2.2^0\,, && (48)\\
c_5 &= 2^3 + c_4 + c_3 && (49)\\
&= 2^3 + (2^2 + 2^1 + 2.2^0) + (2^1 + 2^0) && (50)\\
&= 2^3 + 2^2 + 2.2^1 + 3.2^0\,, && (51)\\
c_6 &= 2^4 + c_5 + c_4 && (52)\\
&= 2^4 + (2^3 + 2^2 + 2.2^1 + 3.2^0) + (2^2 + 2^1 + 2.2^0) && (53)\\
&= 2^4 + 2^3 + 2.2^2 + 3.2^1 + 5.2^0\,, && (54)\\
c_7 &= 2^5 + c_6 + c_5 && (55)\\
&= 2^5 + (2^4 + 2^3 + 2.2^2 + 3.2^1 + 5.2^0) + (2^3 + 2^2 + 2.2^1 + 3.2^0) && (56)\\
&= 2^5 + 2^4 + 2.2^3 + 3.2^2 + 5.2^1 + 8.2^0\,. && (57)
\end{aligned}$$

It would appear that the decreasing powers of 2 have coefficients that match the Fibonacci sequence, namely $1, 1, 2, 3, 5, 8, \ldots$ (from the section on Theoretical Support).

To prove this in general for $n \geq 2$, we postulate the hypothesis

$$H_n : \qquad c_n = \sum_{k=0}^{n-2} F_{n-k-1}.2^k \,. \tag{58}$$

Thus, from the recursion formula above, we have

$$\begin{aligned} c_{n+1} &= 2^{n-1} + c_n + c_{n-1} & (59)\\ &= 2^{n-1} + \sum_{k=0}^{n-2} F_{n-k-1}.2^k + \sum_{k=0}^{n-3} F_{n-k-2}.2^k \,, & (60) \end{aligned}$$

using hypotheses H_n and H_{n-1}. Consequently, we have

$$\begin{aligned} c_{n+1} &= 2^{n-1} + F_1.2^{n-2} + \sum_{k=0}^{n-3} \left\{F_{n-k-1} + F_{n-k-2}\right\}.2^k & (61)\\ &= F_1.2^{n-1} + F_2.2^{n-2} + \sum_{k=0}^{n-3} F_{n-k}.2^k & (62)\\ &= \sum_{k=0}^{n-1} F_{n-k}.2^k = \sum_{k=0}^{n+1-2} F_{n+1-k-1}.2^k \,, & (63) \end{aligned}$$

from the Fibonacci recursion, using the fact that $F_1 = F_2 = 1$. However, this is exactly hypothesis H_{n+1}, which we have now proven by induction, since $c_2 = F_{2-0-1}.2^0 = F_1 = 1$, agreeing with the unwinding results above.

Consequently, for $n \geq 5$ we have

$$\begin{aligned} \bar{c}_{n-1}^{T\rangle} &= 2^{n-2} - c_{n-1}^{T\rangle} = 2^{n-2} - 2^{n-3} - c_{n-3} & (64)\\ &= 2^{n-3} - \sum_{k=0}^{n-5} F_{n-k-4}.2^k \,. & (65) \end{aligned}$$

For $3 \leq n < 5$, we instead find $\bar{c}_{n-1}^{T\rangle}$ via

$$\begin{aligned} \bar{c}_3^{T\rangle} &= 2^2 - c_3^{T\rangle} = 4 - (2^1 + c_1) = 2 \,, & (66)\\ \bar{c}_2^{T\rangle} &= 2^1 - c_2^{T\rangle} = 2 - (2^0 + c_0) = 1 \,, & (67) \end{aligned}$$

since $c_0 = c_1 = 0$ by construction. Lastly, for $n = 2$ we find $\bar{c}_{n-1}^{T\rangle}$ via

$$\bar{c}_1^{T\rangle} = 2^0 - c_1^{T\rangle} = 1 \,, \tag{68}$$

since $c_1^{T\rangle} = 0$ by construction.

Consequently, the first few terms in our probability distribution are

$$p_0 = p_1 = 0\,, \tag{69}$$

$$p_2 = \frac{\bar{c}_1^{T\rangle}}{2^2} = \frac{1}{4}\,, \tag{70}$$

$$p_3 = \frac{\bar{c}_2^{T\rangle}}{2^3} = \frac{1}{8}\,, \tag{71}$$

$$p_4 = \frac{\bar{c}_3^{T\rangle}}{2^4} = \frac{2}{16} = \frac{1}{8}\,, \tag{72}$$

with

$$p_n = 2^{-3} - \sum_{k=0}^{n-5} F_{n-k-4}.2^{-(n-k)}\,, \tag{73}$$

for $n \geq 5$.

Let us now check our counts numerically.

[24]:
```
def c_n(n):
    if n < 2:
        return 0
    return sum(fibonacci(n-k-1) * 2**k for k in range(n-1))


theoretical_c_n_list = list(c_n(n) for n in range(21))
computed_c_n_list = list(count_pattern(n, "TT") for n in range(21))
print(computed_c_n_list)
for observed, expected in zip(computed_c_n_list, theoretical_c_n_list):
    assert observed == expected
```

```
[0, 0, 1, 3, 8, 19, 43, 94, 201, 423, 880, 1815, 3719, 7582, 15397, 31171,
62952, 126891, 255379, 513342, 1030865]
```

Next, let us check our probabilities numerically.

[25]:
```
def bar_cT_n(n):
    if n <= 0:
        return 0
    elif n == 1:
        return 1
    else:
        return 2**(n-2) - c_n(n-2)


def p_n(n):
    return (
        0 if n <= 1
        else bar_cT_n(n-1) / 2**n
    )
```

```
theoretical_p_n_list = list(p_n(n) for n in range(51))
print(theoretical_p_n_list)
```

```
[0, 0, 0.25, 0.125, 0.125, 0.09375, 0.078125, 0.0625, 0.05078125, 0.041015625,
0.033203125, 0.02685546875, 0.021728515625, 0.017578125, 0.01422119140625,
0.011505126953125, 0.009307861328125, 0.00753021240234375, 0.006092071533203125,
0.0049285888671875, 0.003987312316894531, 0.003225803375244 1406,
0.002609729766845703, 0.0021113157272338867, 0.0017080903053283691,
0.0013818740844726562, 0.0011179596185684204, 0.0009044483304023743,
0.0007317140698432922, 0.0005919691175222397, 0.0004789130762219429,
0.0003874488174915313 7, 0.0003134526778012514, 0.0002535885432735085 5,
0.0002051574410870671 3, 0.0001659758563619107, 0.0001342772872845272213,
0.0001086326083168387 4, 8.78856262715999e-05, 7.110096521500964e-05,
5.7521889175404795e-05, 4.6536185891454807e-05, 3.76485652395786e-05,
3.0458329092653003e-05, 2.4641305856221152e-05, 1.9935235201273827e-05,
1.61279440646922e-05, 1.3047780832664557e-05, 1.0555876432505329e-05,
8.539883424418804e-06, 6.908910820335734e-06]
```

[26]:
```
sum(theoretical_p_n_list)
```

[26]:
```
0.9999707333841146
```

[27]:
```
np.sum(np.arange(51) * theoretical_p_n_list)
```

[27]:
```
5.998383427215481
```

[28]:
```
res = simulate_experiment(200_000, "TT")
lens, counts = np.unique(res, return_counts=True)
probs = counts / np.sum(counts)
```

[29]:
```
print(lens)
```

```
[ 2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 47 48 50 51
 52 53 56 68 72]
```

[30]:
```
print(probs[0:49])
```

```
[2.49930e-01 1.25695e-01 1.24830e-01 9.27300e-02 7.83750e-02 6.26250e-02
 5.12350e-02 4.07800e-02 3.32050e-02 2.76100e-02 2.15350e-02 1.74600e-02
 1.40800e-02 1.18550e-02 9.25000e-03 7.45500e-03 5.96500e-03 4.72500e-03
 4.08500e-03 3.24500e-03 2.38000e-03 2.03000e-03 1.55500e-03 1.49000e-03
 1.06500e-03 8.35000e-04 8.05000e-04 6.25000e-04 4.95000e-04 4.65000e-04
 2.95000e-04 2.05000e-04 1.70000e-04 1.80000e-04 1.20000e-04 1.25000e-04
 1.20000e-04 4.00000e-05 7.00000e-05 7.00000e-05 6.00000e-05 3.50000e-05
```

```
1.00000e-05 2.00000e-05 1.00000e-05 5.00000e-06 1.00000e-05 1.00000e-05
1.00000e-05]
```

It appears to be beyond this author's modest powers to directly establish $\bar{N} = \mathbb{E}[N \mid \texttt{H}, S = TT]$ from $p_n = \mathbb{P}(N = n \mid \texttt{H}, S = TT)$. However, we can attempt the trick used in the previous section, by reasoning indirectly.

Thus, we begin with the first coin toss. With probability $\frac{1}{2}$ the result is $t_1 = H$, using up 1 throw. However, we cannot directly stop now with TT, leaving this up to the expected $\bar{N}$ further throws to acheive. Alternatively, with probability $\frac{1}{2}$ the result is $t_1 = T$, using up 1 throw. Now we proceed to the second coin toss. With probability $\frac{1}{2}$ the result is $t_2 = H$ (giving $t_1 t_2 = TH$), using up 1 throw and requiring $\bar{N}$ further throws (on average) to observe TT. Alternatively, with probability $\frac{1}{2}$ the second result is $t_2 = T$ (giving $t_1 t_2 = TT$), using up 1 throw, whereupon the sequence halts. Mathematically, this gives

$$\begin{aligned} \bar{N} &= \frac{1}{2}\left(1 + \bar{N}\right) + \frac{1}{2}\left(1 + \frac{1}{2}\left(1 + \bar{N}\right) + \frac{1}{2}(1)\right) && (74) \\ &= \frac{3}{2} + \frac{3}{4}\bar{N} && (75) \\ \Rightarrow \bar{N} &= 6\,. && (76) \end{aligned}$$

This is indeed the expected sequence length that we have observed computationally.

1.7 Experiment 3 - THH vs HHH

We now extend the experiment to halt upon the first occurrence of a length-3 subsequence. In particular, we are interested in S =THH and S =HHH.

[31]:
```
patterns = deterministic_sequences(3)
for pattern in patterns:
    mu = np.mean(simulate_experiment(100_000, pattern))
    print("%s: %f" % (pattern, mu))
```

```
TTT: 13.934850
TTH: 7.989040
THT: 10.039580
THH: 7.988940
HTT: 8.000460
HTH: 10.050080
HHT: 8.012750
HHH: 14.009410
```

As noted in the Background section, the expected sequence length for S =THH (and HTT by symmetry) is $\bar{N} = 8$, and for S =HHH (and TTT) is $\bar{N} = 14$. We also see that $\bar{N} = 10$ for S =THT and HTH, and lastly $\bar{N} = 8$ for S =TTH and HHT.

For the halting sequence S =THH, let us consider the situation indirectly, as we did in the previous section. Briefly, if $t_1 = H$ then we reset the experiment, expecting a futher $\bar{N}$ throws to halt. If $t_1 = T$ then we proceed to the second throw. At this point, let us define $\bar{N}^{\langle T}$ as the expected number of coin tosses required to terminate (with S =THH) a sequence that starts with T. Further

letting $p_H \doteq \mathbb{P}(H)$ and $p_T \doteq \mathbb{P}(T)$ to help the reader distinguish coin tosses (and thus match mathematical terms with verbal description), we have

$$\bar{N} = p_H\left(1+\bar{N}\right) + p_T\bar{N}^{\langle T}\,. \tag{77}$$

Continuing to the second coin toss, if $t_2 = T$ (i.e. $t_1t_2 = TT$), then we reset the experiment, expecting a futher $\bar{N}^{\langle T}$ throws to halt. However, if $t_2 = H$ (i.e. $t_1t_2 = TH$), then we proceed to the third throw. If $t_3 = T$ (i.e. $t_1t_2t_3 = THT$), then we reset the experiment, expecting a futher $\bar{N}^{\langle T}$ throws to halt. Only if $t_3 = H$ (i.e. $t_1t_2t_3 = THH$) do we halt the sequence.

Mathematically, this description leads to

$$\begin{aligned}
\bar{N}^{\langle T} &= 1 + p_T\bar{N}^{\langle T} + p_H\left(1 + p_T\bar{N}^{\langle T} + p_H(1)\right) && (78)\\
&= \left(1 + p_H + p_H^2\right) + \left(p_T + p_Hp_T\right)\bar{N}^{\langle T} && (79)\\
\Rightarrow \bar{N}^{\langle T} &= \frac{1 + p_H + p_H^2}{1 - p_T - p_Hp_T} && (80)\\
&= \frac{1 - p_H^3}{p_H^2p_T}\,, && (81)
\end{aligned}$$

using the fact that $p_H + p_T = 1$. Substituting this into the equation above gives

$$\begin{aligned}
\bar{N} &= p_H + p_H\bar{N} + p_T\left(\frac{1 - p_H^3}{p_H^2p_T}\right) && (82)\\
\Rightarrow \bar{N} &= \frac{1}{p_Tp_H^2} = \frac{1}{\mathbb{P}(THH)}\,. && (83)
\end{aligned}$$

Clearly, this gives $\bar{N} = 8$ for $p_H = p_H = \frac{1}{2}$.

For the alternative halting pattern, S =HHH, the analysis is more straightforward, since any occurrance of T simply resets the experiment. Thus, we have

$$\begin{aligned}
\bar{N} &= p_T\left(1+\bar{N}\right) + p_H\left\{1 + p_T\left(1+\bar{N}\right) + p_H\left[1 + p_T\left(1+\bar{N}\right) + p_H(1)\right]\right\} && (84)\\
&= \left(p_T + p_H + p_Hp_T + p_H^2 + p_H^2p_T + p_H^3\right) + \left(p_T + p_Hp_T + p_H^2p_T\right)\bar{N} && (85)\\
&= \left\{p_T\left(1 + p_H + p_H^2\right) + p_H\left(1 + p_H + p_H^2\right)\right\} + p_T\left(1 + p_H + p_H^2\right)\bar{N} && (86)\\
&= \frac{1 - p_H^3}{p_T} + \left(1 - p_H^3\right)\bar{N} && (87)\\
\Rightarrow \bar{N} &= \frac{1 - p_H^3}{p_Tp_H^3}\,. && (88)
\end{aligned}$$

For $p_H = p_T = \frac{1}{2}$, this gives $\bar{N} = 14$, as anticipated from the simulations above.

Finally, for the sake of completeness, let us see if we can derive the expected length for sequences that halt with S =THT. Clearly, we reset the experiment in its entirety if $t_1 = H$ or $t_1t_2t_3 = THH$, and if $t_1t_2 = TT$ then we reset the experiment with a sequence starting with T. Thus, we have

$$\begin{aligned}
\bar{N} &= p_H\left(1+\bar{N}\right) + p_T\bar{N}^{\langle T}\,, && (89)\\
\bar{N}^{\langle T} &= 1 + p_T\bar{N}^{\langle T} + p_H\left(1 + p_T(1) + p_H\left(1+\bar{N}\right)\right)\,. && (90)
\end{aligned}$$

The latter equation gives

$$p_H\bar{N}^{\langle T} = 1 + p_H + p_Hp_T + p_H^2 + p_H^2\bar{N}\,, \tag{91}$$

and the former equation gives

$$\begin{aligned} p_T\bar{N} &= p_H + \frac{p_T}{p_H}\left(1 + p_H + p_H p_T + p_H^2 + p_H^2\bar{N}\right) && (92)\\ \Rightarrow \left(p_H p_T - p_T p_H^2\right)\bar{N} &= p_H^2 + p_T + p_T p_H + p_T^2 p_H + p_T p_H^2 && (93)\\ \Rightarrow p_H p_T^2 \bar{N} &= 1 + p_T p_H && (94)\\ \Rightarrow \bar{N} &= \frac{1}{p_H p_T^2} + \frac{1}{p_T} = \frac{1}{\mathbb{P}(THT)} + \frac{1}{\mathbb{P}(T)}\,. && (95) \end{aligned}$$

Thus, we obtain exactly $\bar{N} = 10$ for $p_H = p_T = \frac{1}{2}$, which agrees with the simulation.

1.7.1 Halting with $S =$THH

We have derived above the expected length of sequences that halt with pattern $S =$THH, namely $\bar{N} = 8$. This is sufficient for our purposes of comparing expected length for different halt patterns.

However, we might as well derive the distribution of lengths, just for a final example.

Due to the fact that the pattern THH does not overlap itself, a length-n sequence halting on THH must have the last three coins toss as THH, and the previous $n-3$ coin tosses, for $n \geq 3$, must not contain THH. Thus

$$\begin{aligned} p_n &= \mathbb{P}(t_{n-2}t_{n-1}t_n = THH \wedge THH \notin t_1 \cdots t_{n-3}) && (96)\\ &= \mathbb{P}(t_{n-2}t_{n-1}t_n = THH)\,\mathbb{P}(THH \notin t_1 \cdots t_{n-3} \mid t_{n-2}t_{n-1}t_n = THH) && (97)\\ &= \mathbb{P}(t_{n-2}t_{n-1}t_n = THH)\,\left[1 - \mathbb{P}(THH \in t_1 \cdots t_{n-3})\right] && (98)\\ &= \frac{1}{2^3}\left[1 - \frac{c_{n-3}}{2^{n-3}}\right] = \frac{\bar{c}_{n-3}}{2^n}\,, && (99) \end{aligned}$$

where c_n is the number of distinct length-n sequences containing THH, and $\bar{c}_n = 2^n - c_n$ is the number of distinct length-n sequences not containing THH.

Just to provide some contrast from our previous probabilistic analyses, let us examine starting sequences instead of ending sequences. Thus, let $c_n^{\langle T}$ be the number of length-n sequences starting with T that contain the pattern THH.

For $n \geq 1$, any sequence must start with T or H. If it starts with H, then it cannot start with THH, but it will contain THH if the length-$(n-1)$ post-sequence (i.e. $(t_2 \cdots t_n\rangle$) contains THH, which can occur in c_{n-1} ways. Alternatively, if the sequence starts with T, then $c_n^{\langle T}$ of the 2^{n-1} distinct length-n sequences starting with T will contain THH. Thus, we obtain

$$c_n = c_n^{\langle T} + c_{n-1}\,. \qquad (100)$$

Unwinding this recursion gives

$$c_n = c_n^{\langle T} + c_{n-1}^{\langle T} + \cdots + c_3^{\langle T} + c_2 = \sum_{m=3}^{n} c_m^{\langle T}\,, \qquad (101)$$

since $c_0 = c_1 = c_2 = 0$ by construction.

For $n \geq 2$, any sequence starting with T must start with TT or TH. If it starts with TT, then it cannot start with THH, but the second T might start a length-$(n-1)$ post-sequence containing THH, which can occur in $c_{n-1}^{\langle T}$ ways.

Alternatively, if the sequence starts with TH, then for $n \geq 3$ it either starts with THH or THT. If it starts with THH, then it doesn't matter what the length-$(n-3)$ post-sequence is, and there are 2^{n-3} such completions, giving 2^{n-3} distinct sequences of length n that start with THH. However, if the sequence starts with THT, then the length-$(n-2)$ post-sequence starting with $t_3 = T$ might contain THH, which can occur in $c_{n-2}^{\langle T}$ ways. Consequently, for $n \geq 3$, we obtain the recursion

$$c_n^{\langle T} = 2^{n-3} + c_{n-1}^{\langle T} + c_{n-2}^{\langle T} \,. \tag{102}$$

By construction, we also have $c_0^{\langle T} = c_1^{\langle T} = c_2^{\langle T} = 0$.

We met a similar recursion earlier, except that the leading term was 2^{n-2} then, whereas it is 2^{n-3} now, since $|S| = |\text{THH}| = 3$. It can be shown by induction that the recursion satisfies

$$c_n^{\langle T} = \sum_{k=0}^{n-3} F_{n-k-2}.2^k \,. \tag{103}$$

Therefore, we can substitute this formula for $c_n^{\langle T}$ into the formula for c_n, obtaining

$$\begin{aligned}
c_n &= \sum_{m=3}^{n} \sum_{k=0}^{m-3} F_{m-k-2}.2^k && (104)\\
&= \sum_{k=0}^{n-3} \left\{ \sum_{m=k+3}^{n} F_{m-k-2} \right\} 2^k && (105)\\
&= \sum_{k=0}^{n-3} \left\{ \sum_{j=1}^{n-k-2} F_j \right\} 2^k && (106)\\
&= \sum_{k=0}^{n-3} (F_{n-k} - 1) 2^k && (107)\\
&= 1 - 2^{n-2} + \sum_{k=0}^{n-3} F_{n-k}.2^k \,, && (108)
\end{aligned}$$

which is clearly valid for $n \geq 3$. Note that the progression from the third to fourth line, regarding the sum of Fibonacci numbers, uses the result we derived in the section on Theoretical Support.

Let us now check these results numerically.

[32]:
```
def c_n(n):
    if n < 3:
        return 0
    s = sum(fibonacci(n-k) * 2**k for k in range(n-2))
    return s + 1 - 2**(n-2)


theoretical_c_n_list = list(c_n(n) for n in range(21))
computed_c_n_list = list(count_pattern(n, "THH") for n in range(21))
print(computed_c_n_list)
for observed, expected in zip(computed_c_n_list, theoretical_c_n_list):
```

```
    assert observed == expected
```

```
[0, 0, 0, 1, 4, 12, 31, 74, 168, 369, 792, 1672, 3487, 7206, 14788, 30185,
61356, 124308, 251199, 506578, 1019920]
```

[33]:
```
def p_n(n):
    return (
        0 if n <= 2
        else (2**(n-3) - c_n(n-3)) / 2**n
    )


theoretical_p_n_list = list(p_n(n) for n in range(51))
print(theoretical_p_n_list)
```

```
[0, 0, 0, 0.125, 0.125, 0.125, 0.109375, 0.09375, 0.078125, 0.064453125,
0.052734375, 0.04296875, 0.034912109375, 0.0283203125, 0.02294921875,
0.018585205078125, 0.015045166015625, 0.012176513671875, 0.009853363037109375,
0.00797271728515625, 0.006450653076171875, 0.005218982696533203,
0.004222393035888672, 0.0034160614013671875, 0.002763688564300537,
0.002235889434814453, 0.0018088817596435547, 0.001463420689105975,
0.00118393450975418, 0.0009578242897987366, 0.0007748967036604881,
0.0006269048899412155, 0.0005071768537163734, 0.0004103147657588124,
0.0003319516545161605, 0.0002685545478016138, 0.0002172652020817625,
0.0001757712452672422, 0.0001422019267920405, 0.0001150437765318202,
9.307237087341491e-05, 7.529713002440985e-05, 6.091665795793233e-05,
4.928261159875546e-05, 3.9870470345704234e-05, 3.225588100962696e-05,
2.609556165111826e-05, 2.111175285790523e-05, 1.7079766845284894e-05,
1.3817821638895111e-05, 1.1178852531656958e-05]
```

[34]:
```
print("sum{p_n} =", sum(theoretical_p_n_list))
print("sum{n.p_n} =", np.sum(np.arange(len(theoretical_p_n_list)) *␣
 ↪theoretical_p_n_list))
```

```
sum{p_n} = 0.9999526456207626
sum{n.p_n} = 7.997384330289407
```

[35]:
```
res = simulate_experiment(200_000, "THH")
lens, counts = np.unique(res, return_counts=True)
probs = counts / np.sum(counts)
```

[36]:
```
print(lens)
```

```
[ 3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 50 51
 52 53 54 57 58 59 60 62]
```

[37]:
```
print(probs)
```

```
[1.25960e-01 1.23985e-01 1.24375e-01 1.09330e-01 9.42500e-02 7.88200e-02
 6.39200e-02 5.25750e-02 4.33600e-02 3.55500e-02 2.83700e-02 2.23600e-02
 1.87700e-02 1.45400e-02 1.21850e-02 9.69500e-03 7.78000e-03 6.42000e-03
 5.00000e-03 4.35500e-03 3.52000e-03 2.90500e-03 2.12000e-03 1.82500e-03
 1.63500e-03 1.18500e-03 9.40000e-04 7.80000e-04 6.45000e-04 5.60000e-04
 3.80000e-04 4.10000e-04 3.15000e-04 2.00000e-04 1.95000e-04 1.40000e-04
 1.10000e-04 1.05000e-04 6.50000e-05 4.00000e-05 7.00000e-05 6.00000e-05
 4.00000e-05 2.00000e-05 2.50000e-05 3.00000e-05 1.00000e-05 5.00000e-06
 2.00000e-05 1.00000e-05 1.00000e-05 5.00000e-06 5.00000e-06 5.00000e-06
 5.00000e-06 5.00000e-06]
```

1.7.2 Halting with $S =$THH or $S =$HHH

We now consider the modified experiment where the sequence halts upon the first occurrence of either THH or HHH. We note that these stopping patterns are rather special in that HHH is overlapped to its left by both THH and HHH.

Initially, the experiment can halt on $S =$HHH with a length $n = 3$ sequence with probability $\mathbb{P}(HHH) = \frac{1}{8}$, or on $S =$THH with probability $\mathbb{P}(THH) = \frac{1}{8}$. However, when we look for a length $n = 4$ sequence $\langle t_1 HHH\rangle$, we see that if $t_1 = T$ then the sequence would have previously halted with $t_1t_2t_3 = THH$, whereas if $t_1 = H$ then the sequence would have halted with $t_1t_2t_3 = HHH$. Consequently, no sequence for $n \geq 4$ may halt on HHH. Thus, the probability that the experiment halts on HHH is

$$\mathbb{P}(S = HHH \mid \mathtt{H}, S = HHH \vee S = THH) = \frac{1}{8}, \tag{109}$$

$$\Rightarrow \mathbb{P}(S = THH \mid \mathtt{H}, S = HHH \vee S = THH) = \frac{7}{8}, \tag{110}$$

giving rise to the 7:1 ratio.

We further deduce that any coin toss of T results in a subsequence starting with T that halts with $S =$THH, with an expected length $\bar{N}^{\langle T}$, where

$$\bar{N}^{\langle T} = 1 + p_T\bar{N}^{\langle T} + p_H\left\{1 + p_T\bar{N}^{\langle T} + p_H(1)\right\} \tag{111}$$

$$\Rightarrow \bar{N}^{\langle T} = \frac{1 + p_H + p_H^2}{1 - p_T - p_Hp_T} = \frac{1 - p_H^3}{p_Tp_H^2}, \tag{112}$$

giving $\bar{N}^{\langle T} = 7$ for $p_H = p_T = \frac{1}{2}$.

Hence, the overall expected length $\bar{N}$ is given by

$$\bar{N} = p_T\bar{N}^{\langle T} + p_H\left\{1 + p_T\bar{N}^{\langle T} + p_H\left[1 + p_T\bar{N}^{\langle T} + p_H(1)\right]\right\} \tag{113}$$

$$= p_T(1 + p_H + p_H^2)\bar{N}^{\langle T} + p_H(1 + p_H + p_H^2) \tag{114}$$

$$= (1 - p_H^3)\bar{N}^{\langle T} + \frac{p_H(1 - p_H^3)}{p_T} \tag{115}$$

$$= (1 - p_H^3)\frac{1 - p_H^3}{p_Tp_H^2} + p_H^3\frac{1 - p_H^3}{p_Tp_H^2} = \frac{1 - p_H^3}{p_Tp_H^2}, \tag{116}$$

also giving $\bar{N} = 7$. We now confirm these results numerically via simulation.

[38]:
```
patterns = ["THH", "HHH"]
res = simulate_combined_experiment(500_000, patterns)
for p, r in zip(patterns, res):
    print("%s: %d %f" % (p, len(r), np.mean(r)))
```

```
THH: 437119 7.560083
HHH: 62881 3.000000
```

We observe that the sequences halting with S =HHH all have length 3, as anticipated above.

[39]:
```
print("THH:HHH ratio =", len(res[0]) / len(res[1]))
```

```
THH:HHH ratio = 6.951527488430528
```

The ratio of the number of sequences halting with S =THH compared to S =HHH is about 7:1, as expected.

[40]:
```
print("overall mean =", np.mean(res[0] + res[1]))
```

```
overall mean = 6.986598
```

Finally, the mean length of all sequences produced by the modified experiment is approximately 7, as we deduced above.