# Algorithm for Bioinformatics Exercise 4 David Riemer, Usama Mehmood

### Problem 1

30 points

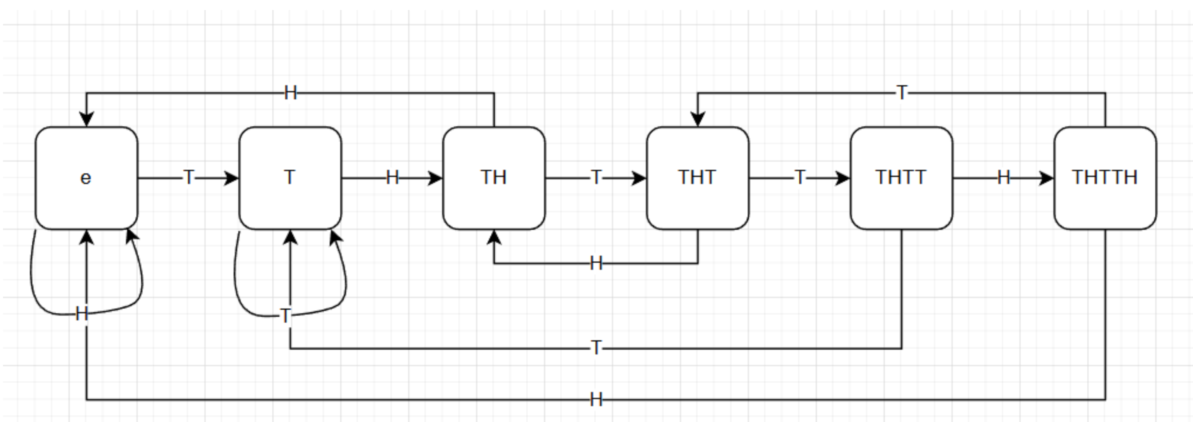
Consider the following biased coin: With probability $p$, it shows H (heads) and with probability $q := 1 - p$, it shows T (tails).

Let $X$ be the random number of coin tosses until the pattern THTTH appears for the first time.

Compute the probability generating function $G_X(z)$ for $X$ and determine the expected value and the variance of $X$ as a function of $p$.

## Solution 1

We create a automaton that visualizes the system of equations we will solve later on.




The system of equations is described we will solve is described as:

$$
\begin{gathered}
F_5(z) = 1 \\
F_0(z) = z\big(pF_0 + qF_1\big) \\
F_1(z) = z\big(pF_2 + qF_1\big) \\
F_2(z) = z\big(pF_0 + qF_3\big) \\
F_3(z) = z\big(pF_2 + qF_4\big) \\
F_4(z) = z\big(p + qF_1\big)
\end{gathered}
$$

Here $F_i$ is the state the coin toss is in and $p$ the probability that the next toss will be a $H$ and $q$ the probability that the next toss is a $T$. So $F_0$ is with probabilty $p$ goes back ot $F_0$ or with probability $q$ to $F_1$. We then leverage GPT-5 to write code using sympy to solve this system of equations to get the pgf, as well as the expected value and variance as a function in relation tp $p$.

```
from sympy import symbols, Eq, solve, simplify, together, diff, limit, Rational

# 1) Define symbols
p, z = symbols('p z')
q = 1 - p

# F_i(z) = E[z^X | start in state i], states 0...5 (5 is absorbing)
F0, F1, F2, F3, F4, F5 = symbols('F0 F1 F2 F3 F4 F5')

# 2) Define the system in "expression = 0" form
#    Automaton for w = THTTH with transitions:
#    0 --H→ 0, 0 --T→ 1
#    1 --H→ 2, 1 --T→ 1
#    2 --H→ 0, 2 --T→ 3
#    3 --H→ 2, 3 --T→ 4
#    4 --H→ 5, 4 --T→ 1
equations = [
    Eq(F5 - 1, 0),
    Eq(F0 - z*(p*F0 + q*F1), 0),
    Eq(F1 - z*(p*F2 + q*F1), 0),
    Eq(F2 - z*(p*F0 + q*F3), 0),
    Eq(F3 - z*(p*F2 + q*F4), 0),
    Eq(F4 - z*(p*F5 + q*F1), 0),
]

# 3) Variables to solve for
variables = (F0, F1, F2, F3, F4, F5)

# 4) Solve the system
solution = solve(equations, variables, dict=True)[0]

# 5) PGF G_X(z) = F0(z)
G = simplify(together(solution[F0]))

print("Probability Generating Function G_X(z)")
print("G_X(z) =", G)
```

```
# 6) Expected value and Variance2
#    For PGF: E[X]=G'(1),  Var[X]=G''(1) + G'(1) - (G'(1))^2
G1 = diff(G, z)
G2 = diff(G, z, 2)

EX  = simplify(limit(G1, z, 1))
Var = simplify(limit(G2, z, 1) + EX - EX**2)

print("E[X] =", EX)
print("Var[X] =", Var)

OUTPUT:
PS C:\Users\David> & C:/Users/David/AppData/Local/Microsoft/WindowsApps/python3.13.ex
e c:/Users/David/Desktop/solver.py
Probability Generating Function G_X(z)
G_X(z) = p**2*z**5*(p**3 - 3*p**2 + 3*p - 1)/(p**5*z**5 - 3*p**4*z**5 + 3*p**3*z**5 + p**
3*z**4 - p**3*z**3 - p**2*z**5 - 2*p**2*z**4 + 2*p**2*z**3 + p*z**4 - p*z**3 + z - 1)
E[X] = (-p**3 + 2*p**2 - p - 1)/(p**2*(p**3 - 3*p**2 + 3*p - 1))
Var[X] = (3*p**8 - 15*p**7 + 31*p**6 - 25*p**5 - 6*p**4 + 22*p**3 - 12*p**2 + 2*p + 1)/(p*
*4*(p**6 - 6*p**5 + 15*p**4 - 20*p**3 + 15*p**2 - 6*p + 1))
```

Leading to the equations:

$$G_X(z) = \frac{p^2 z^5 \left(p^3 - 3p^2 + 3p - 1\right)}{p^5 z^5 - 3p^4 z^5 + 3p^3 z^5 + p^3 z^4 - p^3 z^3 - p^2 z^5 - 2p^2 z^4 + 2p^2 z^3 + pz^4 - pz^3 + z - 1}$$

$$\mathbb{E}[X] = \frac{-p^3 + 2p^2 - p - 1}{p^2(p^3 - 3p^2 + 3p - 1)}$$

$$\mathrm{Var}(X) = \frac{3p^8 - 15p^7 + 31p^6 - 25p^5 - 6p^4 + 22p^3 - 12p^2 + 2p + 1}{p^4(p^6 - 6p^5 + 15p^4 - 20p^3 + 15p^2 - 6p + 1)}$$