CS 145 – Foundations of Computer Science

Professor Eric Aaron

**Lecture** – M W 10:30am  
**Lab** – F 3:10pm

**Lecture Meeting Location:** SP 105  
**Lab Meeting Location:** SP 309

---

**Business**

- CEQs today

- HW7:  
  - Programming and printouts: due already  
  - Non-programming exercises:  
    - Given to me before 4pm today; or  
    - Given to Linda Wood in the main CS office before 3pm on May 10

- Grading update
- Reading: Makinson, Ch. 6  
  - We may not cover all the material, but it’s worth reading anyway

- **Purely optional** Lab8 available at course website  
  - (With a very cool result about expected values and biased coins!)

- Scheduling review session for final exam  
  - Early next week? Late this week?

---

*Last day for Coaching Hours this semester is Tuesday, May 10*
Expected Values

- When the elements in a sample space have probabilities associated with them (e.g., the chance that a lottery ticket is a winner)
- ... and those outcomes have values (or payoffs) associated with them,
- ... we can talk about the expected value (or expectation) of an outcome—the probability-weighted average value of the payoff function
- More formally
  - Let $S$ be a sample space, $p$ be a probability distribution on $S$
  - ... and let $f$ be a payoff function that maps each element of $S$ to a real number (it could be any real number, positive or negative)
  - Then the expected value of $f$ given $p$ is $\sum f(s)p(s)$, summed over all $s$ in $S$
  - Example: What is the expected value of a roll of a fair 6-sided die?

Expected Values

- We can talk about the expected value (or expectation) of an outcome—the probability-weighted average value of the payoff function
  - The expected value of $f$ given $p$ is $\sum f(s)p(s)$, summed over all $s$ in $S$

- Example: A fair coin is flipped three times.
  - Let $S$ be the sample space of the eight possible outcomes, and
  - ... let $f$ be the payoff function that maps each element of $S$ to the number of heads in that outcome
  - What is the expected value of $f$?
Digression: What’s A Number?

- Actually, the question is “How do we represent numbers?”
- Several possibilities – We’re used to decimal (i.e., base 10), using digits 0 through 9, but there are also:
  - Binary (base 2): using digits 0 and 1
  - Octal (base 8): using digits 0 through 7
  - Hexadecimal (base 16): using “digits” 0 through “15”?!
    - (we use letters A-F to represent numbers 10-15)
  - (Why are these three possibilities well-suited for computers?)
- In all cases, each digit represents a number of (base-position)
  - e.g., in base 10: 209 = 2 * 10^2 + 0 * 10^1 + 9 * 10^0
    - (in base 100: 29_{100} = 2 * 100^1 + 9 * 100^0 = 209)
  - in base 2: 1010_2 = 1 * 2^3 + 0 * 2^2 + 1 * 2^1 + 0 * 2^0 = ??
  - in base 16: A3E_{16} = 10 * 16^2 + 3 * 16^1 + 14 * 16^0 = ??
**Digression: Converting Between Number Bases**

- We indicate the base of a number by a subscript
  - e.g., $1101_2$ is in binary, $1101_{16}$ is in hexadecimal
  - We omit subscript for decimal numbers
- How to convert between bases?
  - $47 = ??_2$; $51 = ??_{16}$; $51_8 = ??$
  - $29_{100} = ??$
  - $10100010_2 = ??_{16}$; $1F_{16} = ??_2$
    - What’s the trick with these last two?
    - (Can double-check conversions by going through base 10)
So, about that lab...

• Simulating a biased coin with a fair coin
• Based on representing a fraction / decimal in binary representation
  – If we see a binary representation like the mystery values from lab
    • 0.101010101010101010101010101....
    • 0.011011011011011011011011011....
  – … how could we analytically compute what they are?
• Computing the expected value of tosses of a fair coin that it would take to simulate a biased coin
  – How could we analytically compute that expected value, rather than just simulating it with Gen-And-Test?

So, about that lab pt. 2

• Simulating a biased coin with a fair coin
• Based on representing a fraction / decimal in binary representation
  – If we see a binary representation like the mystery values from lab
    • 0.101010101010101010101010101....
    • 0.011011011011011011011011011....
  – … how could we analytically compute what they are?
So, about that lab pt. 2

- Simulating a biased coin with a fair coin
- Based on representing a fraction/decimal in binary representation
  - If we see a binary representation like the mystery values from lab
    - $0.101010101010101010101010101\ldots$
    - $0.011011011011011011011011011\ldots$
  - … how could we analytically compute what they are?
    - For the first one, let $x = 0.101010101010101010101010101\ldots$
    - What does $x/2$ equal, in binary?
    - What’s $(x + x/2)$ equal, in binary?
    - What’s $x$ equal, in decimal?

So, about that lab pt. 3

- Simulating a biased coin with a fair coin
- Computing the expected value of tosses of a fair coin that it would take to simulate a biased coin
  - How could we analytically compute that expected value, rather than just simulating it with Gen-And-Test?
  - What is the formula for the expected value?
    - (This is a formula that you could derive, based on concepts covered in lab and today in lecture!)
  - And then, how could we calculate the value from that formula?
    - (This… this is a bit tricky to see, if you’ve never seen it before….)

Does the expected value depend on the bias?
Isn’t that kinda amazing?!?