CS 145 – Foundations of Computer Science

Professor Eric Aaron

Lecture – W F 1:30pm
Lab – F 3:30pm

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

Business

• HW7 due Dec. 5 / Dec. 6 (see HW sheet)
  – (We will have covered all the needed material by the end of class today!)
  – NOTE the atypical deadline!
  – Turn in to Jennie Colabella—see HW sheet for details

• HW5 and HW6 grading update
• A note about the Final Exam
• Scheduling the Final Exam review session

• Reading: Makinson, Ch. 6
  – We may not cover all the material in chapter 6, but it’s worth reading anyway
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 \( \sum f(s) \cdot p(s) \), 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 \( \sum f(s) \cdot p(s) \), 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 the 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)