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

• HW5 due Nov. 14 / Nov. 15
  – (See assignment sheet for deadlines, except with Nov. 14 for Nov. 9, and Nov. 15 for Nov. 10)
• HW6 out today, due Nov. 21 / Nov. 22
• Lab in today’s 3:30pm class meeting
  – Due by end of day Nov. 16
• Exams back today (after lab)

• Reading: Ch. 5

• Document on structural induction available from course website (follow the Additional Notes link)
A Note On Scheme:
(apply append (…))

- In Scheme, the *apply* function takes a function F and a list L = (L1 .. Lk), and it applies function F with arguments L1 ... Lk.
  - That is, the elements of the list become the arguments of F
- This can be useful when F is the append function—then, (apply append (…)) can “flatten” a list
- Examples:
  - (apply append '((a b c) (d e f) (g h) (i) (j k l)))
A Note On Scheme:
(apply append (…))

• In Scheme, the *apply* function takes a function F and a list L = (L1 .. Lk), and it applies function F with arguments L1 ... Lk.
  – That is, the elements of the list become the arguments of F
• This can be especially useful when working with the result of the map function, which returns a list
• Example:
  – `(apply append (map (lambda (x) (list x (* 10 x))) '(1 2 3 4 5)))`

A Note On Scheme:
(apply append (…))

• In Scheme, the *apply* function takes a function F and a list L = (L1 .. Lk), and it applies function F with arguments L1 ... Lk.
  – That is, the elements of the list become the arguments of F
• This can be especially useful when working with the result of the map function, which returns a list
• Example:
  – `(apply append (map (lambda (listy)
                           (map (lambda (elt)
                                 (list elt elt))
                           listy))
                '((1 2 3) (4 5) (6 7 8))))`
Proving Statements about Recursively Defined Sets

• A related kind of induction is called structural induction
  – Base case(s): Show that P holds for every element in the basis for the recursive definition.
  – Inductive case(s): Show that every constructor in the definition preserves property P.

• Recall our recursive definition of propositional logic expressions
  – Base: Given an initial set A of propositional letters (e.g., p, q, r, …), all elements of A are propositional logic expressions
  – Induction: If P, Q are propositional logic expressions, then the following are also propositional logic expressions (note that the parentheses are part of the expressions)
    • (¬P); (P ^ Q); (P v Q); (P → Q); (P ↔ Q)  (Note: 5 constructors)

• Claim: All propositional logic expressions contain an even number of parentheses. (We consider 0 to be an even number.) Prove by structural induction.
Selection: Order and Repetition

• Sometimes it’s useful to consider the number of ways to select (or choose) \( k \) items out of \( n \) items

• Two factors to consider:
  – Order: does the order matter when considering the elements? (E.g., is selecting \( a, b \) in that order different from selecting \( b, a \) in that order?)
  – Is repetition permitted? (Can the same element be selected multiple times)

• How many ways can we choose 2 letters from \{a, b, c, d, e\}? 
  – If order matters and repetition is permitted?
  – If order matters and repetition is not permitted?
  – If order doesn’t matter and repetition is permitted?
  – If order doesn’t matter and repetition is not permitted?

Selection: Permutations

• The operations for selecting \( k \) out of \( n \) items (for given \( k \leq n \)) are particularly important when repetition is not permitted
  – Corresponds to working with sets, where elements are not repeated

• Those operations are often called combination and permutation
  – Difference: the order in which they’re selected matters in permutations, not in combinations

• Permutations:
  – Number of ways to select \( k \) out of \( n \) items, where order matters = …?
Selection: Permutations

• The operations for selecting \( k \) out of \( n \) items (for given \( k \leq n \)) are particularly important when repetition is not permitted
  – Corresponds to working with sets, where elements are not repeated
• Those operations are often called *combination* and *permutation*
  – Difference: the order in which they’re selected matters in permutations, not in combinations
• Permutations:
  – Number of ways to select \( k \) out of \( n \) items, where order matters =
    \[
    P(n, k) = \frac{n!}{(n-k)!}
    \]
  – How is this formula derived?
    It may not be necessary to memorize this formula; it could be derived when needed.

Selection: Combinations

• The operations for selecting \( k \) out of \( n \) items (for given \( k \leq n \)) are particularly important when repetition is not permitted
  – Corresponds to working with sets, where elements are not repeated
• Those operations are often called *combination* and *permutation*
  – Difference: the order in which they’re selected matters in permutations, not in combinations
• Combinations:
  – Number of ways to select \( k \) out of \( n \) items, where order doesn’t matter = ...
  \[
  \text{...}
  \]
Selection: Combinations

- The operations for selecting k out of n items (for given k ≤ n) are particularly important when repetition is not permitted
  - Corresponds to working with sets, where elements are not repeated
- Those operations are often called combination and permutation
  - Difference: the order in which they're selected matters in permutations, not in combinations
- Combinations:
  - Number of ways to select k out of n items, where order doesn’t matter = …?
    \[ C(n, k) = \frac{n!}{k! \cdot (n-k)!} \]
  - How is this formula derived?

  It may not be necessary to memorize this formula; it could be derived when needed.

Counting 1

- Using the counting ideas from combinations and permutations…

- Exercises
  - Your investment advisor gives you a list of 8 stocks that seem like good investments. You decide to invest in 3 of them. How many different selections are possible?
  - Same scenario, except you decide to invest $1,000 in one stock, $2,000 in another, and $4,000 in a third. How many different selections are possible?
Counting 2—Repetition

• It can be useful to think about situations in terms of possibly repeated elements being counted, as well as possible orderings of elements

• Using the counting ideas from combinations and permutations…

• Exercises
  – A restaurant has five flavors of ice cream, and you can order one, two, or three scoops. How many different ice cream orders could you place?

Counting 3—Repetition

• Using the counting ideas from combinations and permutations…

• Exercises
  – How many ways are there to rearrange the letters in the word banana?

(As always, explain your answer, including the way you model the situation!)