COMPSCI 575/MATH 513
Combinatorics and Graph Theory

Lecture #1: Combinatorics Overview
David Mix Barrington
7 September 2016

Combinatorics Overview

- Mechanics of the Course
- Outline of the Course
- What is Combinatorics?
- The Mastermind Game
- Basic Counting Problems
- Derangements
- Generating Functions

Mechanics of the Course

- Three sections (CS ugrad, CS grad, MATH) all treated the same, one Moodle course site.
- Two evening midterms (20% each), final (30%), seven homeworks, (5% each, one dropped).
- Prereq is CS 250 with a B, or later CS 311, MATH 411, or comparable math course.
- Counts as upper-level elective for CS or math, fits in some menus for tracks.

More Mechanics

- Textbook is Tucker's *Applied Combinatorics*, ordered with Amazon virtual bookstore.
- Homework problems will be mostly from the book, so you will need it.
- Lectures will 80-90% follow the book, lecture slides will be posted on Moodle.
- I will post practice exams a week before each real exam.

Outline of the Course

- Part I: Graph Theory. Definitions, path theorems, coloring, algorithms for paths, spanning trees, network flow, matching.
- Part II: Basic Combinatorics. Counting problems, generating functions, recurrences, inclusion/exclusion.
- Part III: Advanced Combinatorics. Connections to algebra, Conway game theory.

What is Combinatorics?

- The study of finite discrete structures
- **Enumerative** combinatorics: How many structures of a particular type and of size n meet a particular constraint?
- The answer f(n) may be described in **closed form**, as a **recurrence**, or by a **generating function**.

The Fibonacci Sequence

- How many strings of a's and b's, of length n, begin with a and have no aa substring? (The strings in the language $a(b+ba)^*$.)
- $f(0) = 0, f(1) = 1, f(2) = 1, f(3) = 2, f(4) = 3, \ldots$
- For $n \geq 2$, $f(n) = f(n-1) + f(n-2)$
- $f(n) = (1/r)(((1+r)/2)^n + ((1-r)/2)^n)$ where $r = \sqrt{5}$.
- $\Sigma\, f(n)x^n = x/(1 - x - x^2)$, a generating function

Why Combinatorics?

- Math is beautiful. Mathematicians build abstract structures that help describe the real world, then expand into structures motivated by their own beauty.
- Computers lead us to consider vast arrays of possibilities, organized in logical ways. To consider all the possibilities, we need to count them. In particular, we often want to know about the possible paths in a graph.

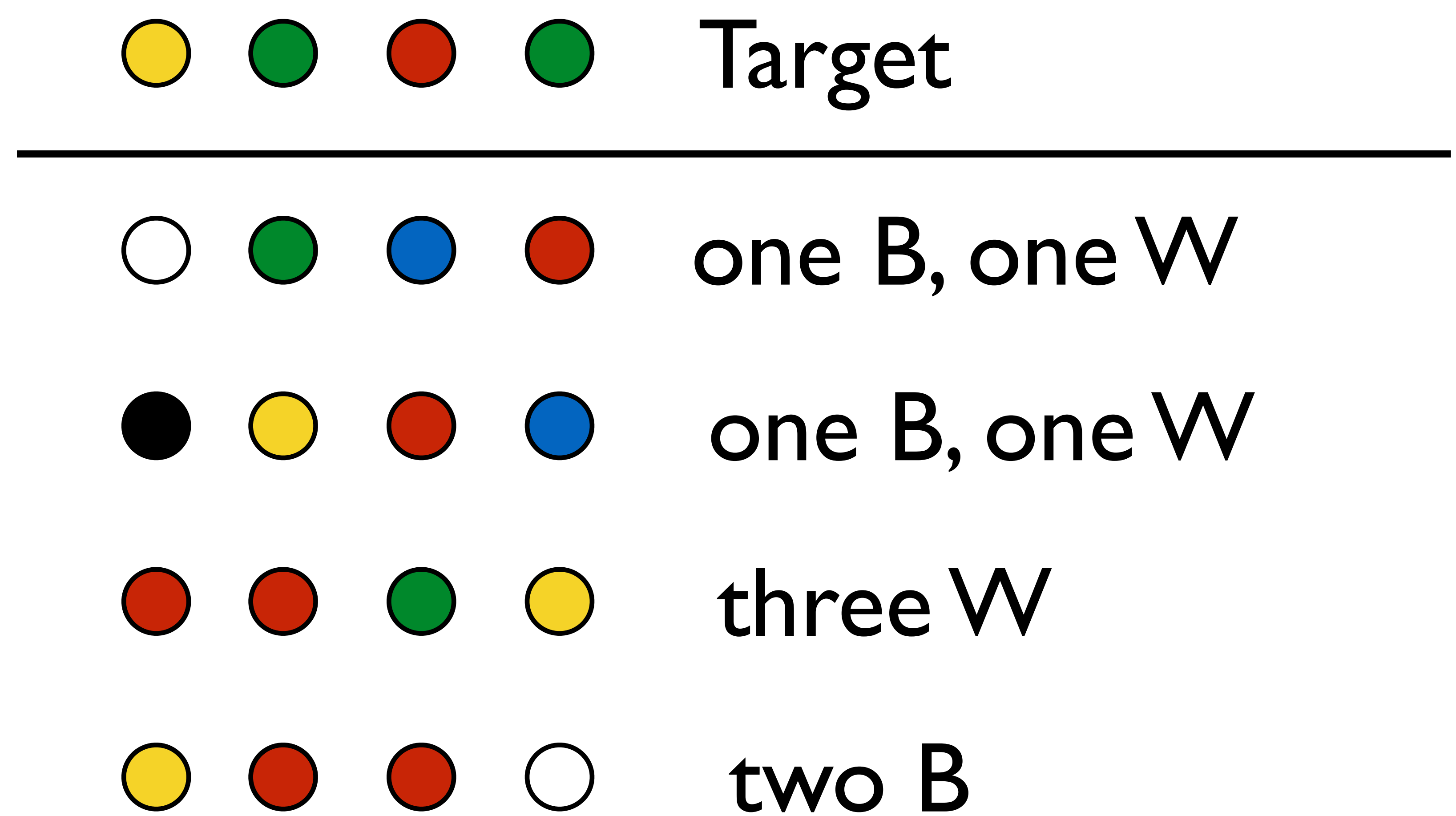
Why Combinatorics?

- Combinatorics underlies probability, as you've probably seen. When there are a large number of equally possible cases, the probability of an event is defined in terms of the number of cases in the event.
- The analysis of algorithms in terms of time complexity frequently turns on counting problems and their solutions.

The Mastermind Game

- Basic object: Sequence of four colors taken from six possibilities {Bk, Bu, G, R, W, Y}
- Given two sequences, they have a black match if the same color is in the same place, and a white match if the same color is in a different place after black matches are counted.
- The game is to guess a hidden sequence by using the feedback from guessed sequences.

A Mastermind Game



- This is enough to deduce the target (see page xii of Tucker)

Mastermind Counting

- How many possible sequences? Each place has six choices, for $6^4 = 1296$.
- How many have four different colors? Six choices for first place, five for second, four for third, three for fourth, for 6*5*4*3 = 720.
- Suppose the target has four different colors? How many sequences have four total matches? (4! = 24).
- How many have four **white** matches?

Derangements

- Given a set of n objects, a **permutation** is an ordering of them, and there are n! total permutations. We can write a permutation by given the order of the elements, e.g., 31524, where numbers give the original order.
- A **derangement** is a permutation that has no fixed points. 31524 is a derangement.
- D(n) is the number of derangements of n elements. D(0) = 1, D(1) = 0, D(2) = 1, D(3) = 2, and D(4) = 9 by inspection.
- How could we compute D(n) for larger n?

Counting Derangements

- It turns out that D(n) is the closest integer to n!/e, except when n = 0.
- We might try to analyze derangements of n in terms of derangements of smaller numbers, getting a recurrence.
- Later we'll solve this problem using inclusion/exclusion.
- We can generalize to count the permutations where a fixed set of assignments doesn't occur.

Generating Functions

- It turns out to be useful to think of a sequence f(0), f(1), f(2),… as a single mathematical object $F = \Sigma f(i)x^i$.
- For the Fibonaccis, $F = x + x^2 + 2x^3 + 3x^4 + 5x^5 + \ldots$
- If we compute $F(1 - x - x^2)$, by cancellation it turns out we get x. So F can be thought of as the **rational function** $x/(1 - x - x^2)$. (I'll do this on the board.)

More Generating Functions

- It turns out that familiar operations on polynomials and rational functions, like multiplication and taking derivatives, have combinatorial meanings.
- There are combinatorial problems that can be solved using generating functions, without necessarily giving us a **combinatorial proof** of the answer.
- In that case we often look for a combinatorial proof because it gives us more insight into why.