CSE 312

Foundations of Computing II

Lecture 1: Introduction & Counting

https://cs.washington.edu/312

Instructor

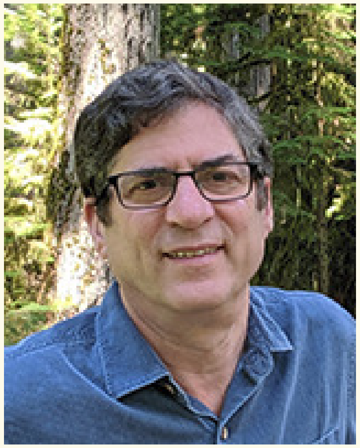
Paul Beame [he/him]

beame@cs

Specialty: **Complexity**

https://homes.cs.washington.edu/~beame

Office: CSE 668



A Team of fantastic TAs

Xinyue Chen

Lukshya Ganjoo

Viraj Goyal

Vlad Murad

Sela Navot

Khanh Nguyen

Morgan Putnam

Edward Qin

Emily Robinson

Tanmay Shah

Claris Winston

Ben Zhang

See https://cs.washington.edu/312/staff.html to learn more about their backgrounds and interests!

Lectures and Sections

- **Lectures MWF (Anderson Hall 223)**
  - 1:30-2:20pm
  - Classes will be in person
  - Lecture recording but no live streaming because of classroom
    - Panopto manual recording - not automatically uploaded. Please bear with me!
  - Annotated slides also uploaded.
- **Poll Everywhere**
  - We will sometimes use Poll Everywhere during class
  - You sign up directly

  Go to https://www.polleverywhere.com/login and login using YOURNETID@uw.edu
- **Sections Thu (starts this week)**
  - Not recorded
  - Will prepare you for problem sets!

Questions and Discussions

- **Office hours throughout the week (starting this Friday)**
  - See https://cs.washington.edu/312/staff.html
- **Ed Discussion**
  - You should have received an invitation (synchronized with the class roaster)
  - Material (resources tab)
  - Announcements (discussion tab)
  - Discussion (discussion tab)

Use Ed discussion forum as much as possible. You can make private posts that only the staff can view! Email instructors for personal issues.

Engagement

- **"Concept checks" after each lecture 5-8 %**
  - Must be done (on Gradescope) before the next lecture by 1:00 pm.
  - Simple questions to reinforce concepts taught in each class
  - Keep you engaged throughout the week, so that homework becomes less of a hurdle
- **9 Problem Sets (Gradescope) 45-50 %**
  - Solved individually. Discussion with others allowed but separate solutions
  - Generally due Wednesdays starting next week, except for midterm week but Fridays after Thanksgiving
  - First problem set posted later today
- **Midterm 15-20 %**
  - In class on **Wednesday, Nov 2**
- **Final Exam 30-35 %**
  - **Monday, December 12** at **2:30-4:20 pm** in this room (as in UW Autumn Quarter Exam Schedule)

Check out the syllabus for policies on late submission for checkpoints and HW

COVID-19

UW's policies are pretty much what they were last spring

- Masking is still recommended, in particular in a crowded settings
- Masking is strongly recommended for the first 2 weeks of the quarter
- Stay home 5+ days with Covid

For more details see

Course Webpage **https://cs.washington.edu/312**

Foundations of Computing II

=

Introduction to Counting, Probability & Statistics

for computer scientists

What is probability??

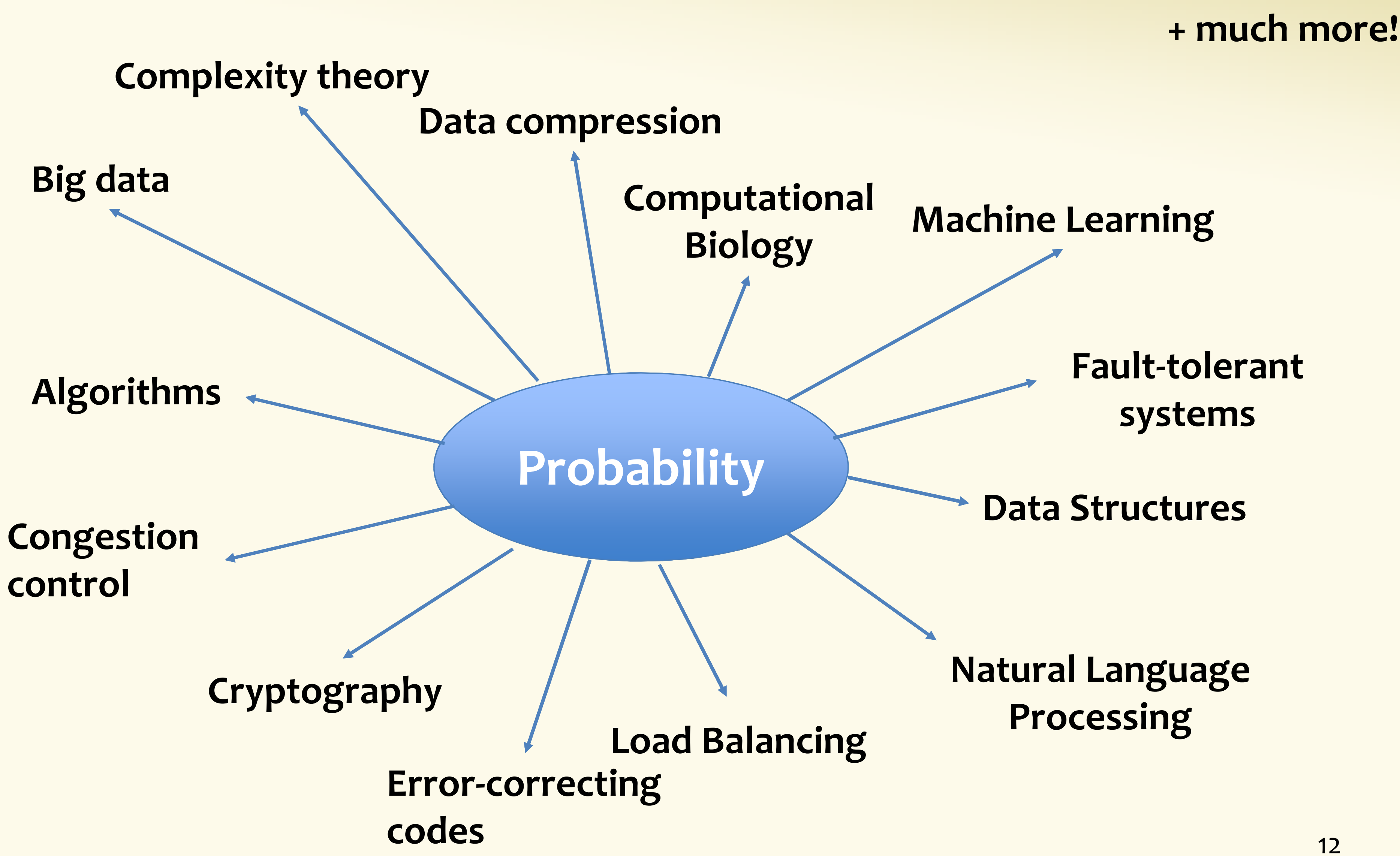
Why probability?!

Probability is our tool for understanding random processes

- Randomness in nature and the sciences/social sciences
  - At the quantum level, everything is random
  - Best way to understand and simulate behavior of complex systems
  - A way to design and understand experiments, observations
    - In the lab, the field, medical trials, surveys
- In Computer Science, randomness has these kinds of roles but also important new ones…

Probability and randomness in Computer Science

- Understanding/modelling the inputs to and behavior of our algorithms
  - In ML, program testing/fuzzing, algorithm analysis, …
- Experiments to validate our designs
  - In user studies, HCI, CS applications in other fields, …
- A tool for hiding information, protecting against adversaries/failures
  - Cryptography, privacy, fault tolerance, computer security, …
- A tool for simpler and more efficient design
  - Data structures, algorithms, ML, …
- …


+ much more!
Complexity theory
Data compression
Big data
Computational Biology
Machine Learning
Algorithms
Fault-tolerant systems
Probability
Data Structures
Congestion control
Cryptography
Natural Language Processing
Load Balancing
Error-correcting codes

Content

- Counting (basis of discrete probability)
  - Counting, Permutation, Combination, inclusion-exclusion, Pigeonhole Principle
- What is probability
  - Probability space, events, basic properties of probabilities, conditional probability, independence, expectation, variance
- Properties of probability
  - Various inequalities, Zoo of discrete random variables, Concentration, Tail bounds
- Continuous Probability
  - Probability Density Functions, Cumulative Density Functions, Uniform, Exponential, Normal distributions, Central Limit Theorem, Estimation
- Applications
  - A sample of randomized algorithms, differential privacy, learning …

Today: A fast introduction to counting so you will have enough to work on in section tomorrow…

We are interested in counting the number of objects with a certain given property.

"How many ways are there to assign 7 TAs to 5 sections, such that each section is assigned to two TAs, and no TA is assigned to more than two sections?"

"How many positive integer solutions (x, y, z) *does the equation* $x^3 + y^3 = z^3$ *have?"*

Generally: Question boils down to computing <u>cardinality</u> $|S|$ of some given set S.

(Discrete) Probability and Counting are Twin Brothers

"What is the probability that a random student from CSE312 has black hair?"

$$= \frac{\#\ students\ with\ black\ hair}{\#students}$$

Today – Two basic rules

- Sum rule
- Product rule

Sum Rule

If you can choose from

- EITHER one of n options,
- OR one of m options with NO overlap with the previous n

then the number of possible outcomes of the experiment is

$$n + m$$

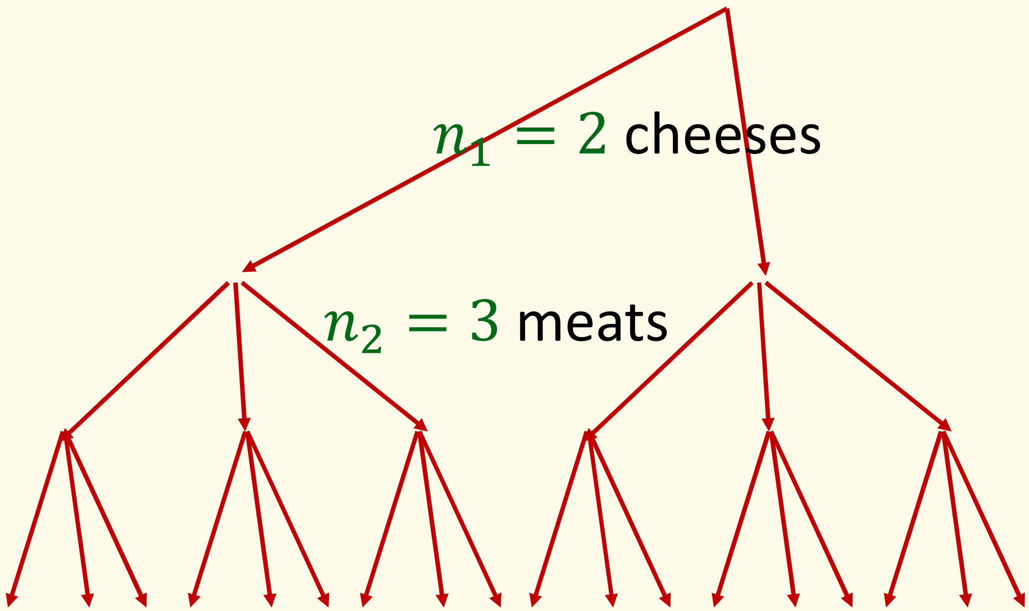
Counting "lunches"

If a lunch order consists of **either** one of 6 soups **or** one of 9 salads, how many different lunch orders are possible?

Product Rule: In a sequential process, there are

- n_1 choices for the first step,
- n_2 choices for the second step (given the first choice), …, and
- n_m choices for the m^{th} step (given the previous choices),

then the total number of outcomes is $n_1 \times n_2 \times \cdots \times n_m$



$n_3 = 3$ veggies

Example: "How many subs?"

$\square \times \square \times \square = \square$

Product rule examples – Strings

How many strings of length 5 over the alphabet $\{A, B, C, \ldots, Z\}$ are there?

- E.g., AZURE, BINGO, TANGO, STEVE, SARAH, …

$\square \times \square \times \square \times \square \times \square = \square$

How many binary strings of length n over the alphabet $\{0,1\}$?

- E.g., $0 \cdots 0, 1 \cdots 1, 0 \cdots 01$, …

$\square \times \square \times \square \times \cdots \times \square = \square$

Product rule example – Cartesian Product

Definition. The **cartesian product** of two sets S, T is

$$S \times T = \{(a, b): a \in S, b \in T\}$$

Called a 2-sequence
Order matters! $(a, b) \neq (b, a)$

$$|S \times T| = \square \times \square$$

$$|A_1 \times A_2 \times \cdots \times A_n| = \square \times \square \times \square \times \cdots \times \square$$

Product rule example – Power set

Definition. The **power set** of S is the set of all subsets of S,

$$\{X : X \subseteq S\}.$$

Notations: $\mathcal{P}(S)$ or simply 2^S (which we will use).

Example.
$2^{\{\bigstar,\spadesuit\}} = \{\emptyset, \{\bigstar\}, \{\spadesuit\}, \{\bigstar, \spadesuit\}\}$
$2^{\emptyset} = \{\emptyset\}$
…

How many different subsets of S are there if $|S| = n$?

Product rule example – Power set

set $S = \{e_1, e_2, e_3, \cdots, e_n\}$

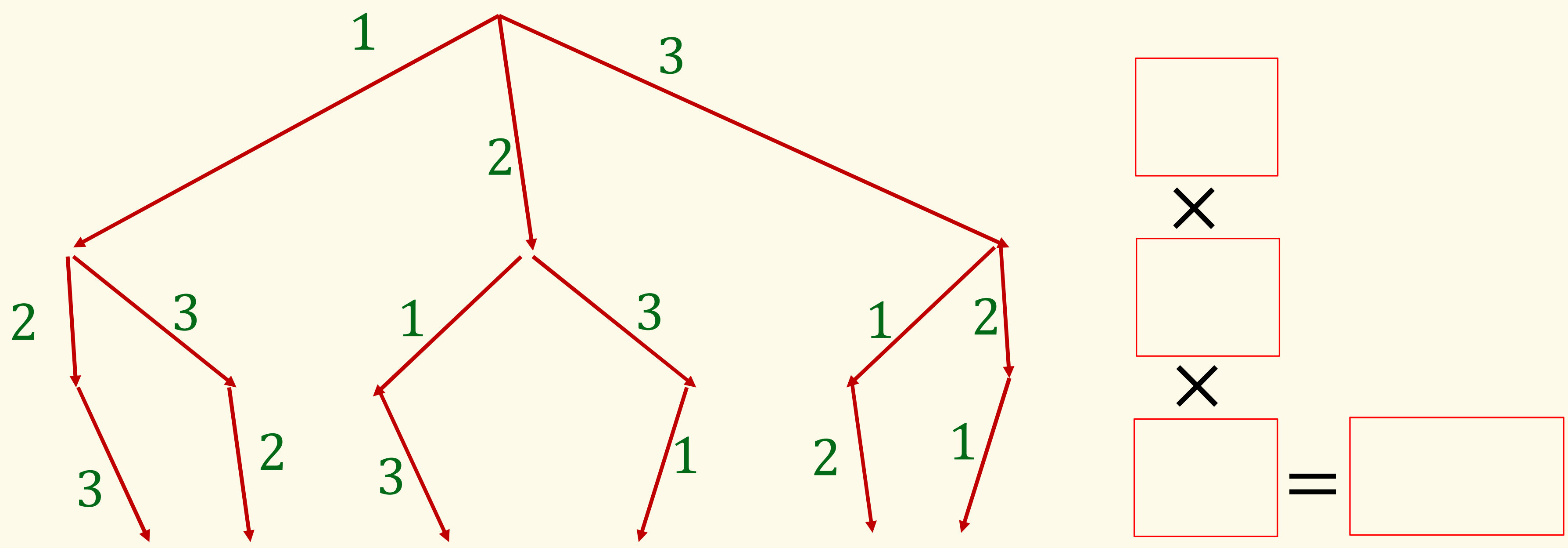
subset $X = \{ \qquad\qquad\qquad\qquad \}$

$\square \times \square \times \square \times \cdots \times \square = \square$

Proposition. $|2^S| = 2^{|S|}$

Note: Sequential process for product rule works even if the sets of options are different at each point

"How many sequences in $\{1,2,3\}^3$ *with no repeating elements?"*



Factorial

"How many ways to order elements in S, where $|S| = n$?" **Permutations**

Answer = $n \times (n-1) \times (n-2) \times \cdots \times 2 \times 1$

Definition. The **factorial function** is

$$n! = n \times (n-1) \times \cdots \times 2 \times 1$$

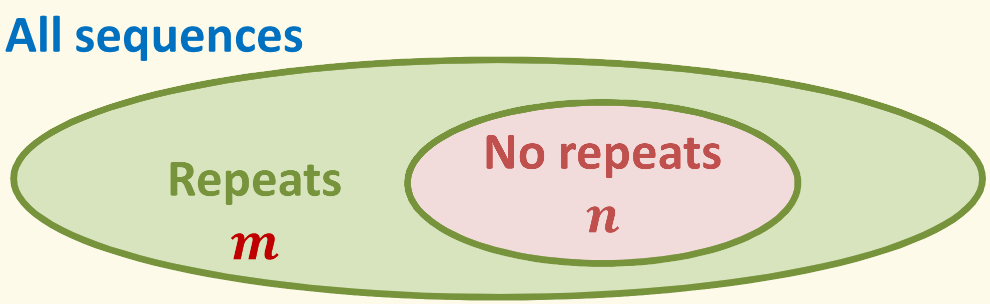
Note: $0! = 1$

Nice use of sum rule: Counting using complements

"How many sequences in $\{1,2,3\}^3$ *have repeating elements?"* m

"# of sequences in $\{1,2,3\}^3$ *with no repeating elements"* $n =$ [ ]

"# of sequences in $\{1,2,3\}^3$ $3^3 = 27$ $= m + n$ *by the sum rule*



$m = 27 - n =$ [ ]

Distinct Letters

"How many sequences of 5 ***distinct*** *alphabet letters from* $\{A, B, \ldots, Z\}$*?"*

E.g., AZURE, BINGO, TANGO. But not: STEVE, SARAH

Answer: $26 \times 25 \times 24 \times 23 \times 22 = 7893600$

In general

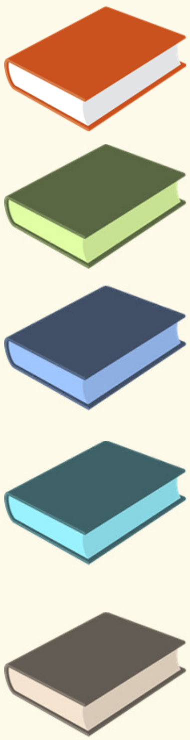
Aka: k-permutations

Fact. # of k-element sequences of distinct symbols from an n-element set is

$$P(n,k) = n \times (n-1) \times \cdots \times (n-k+1) = \frac{n!}{(n-k)!}$$

Product rule – One more example

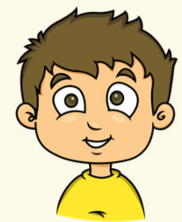
5 books



“How many ways are there to distribute 5 books among Alice, Bob, and Charlie?”

Every book to one person, everyone gets ≥ 0 books.

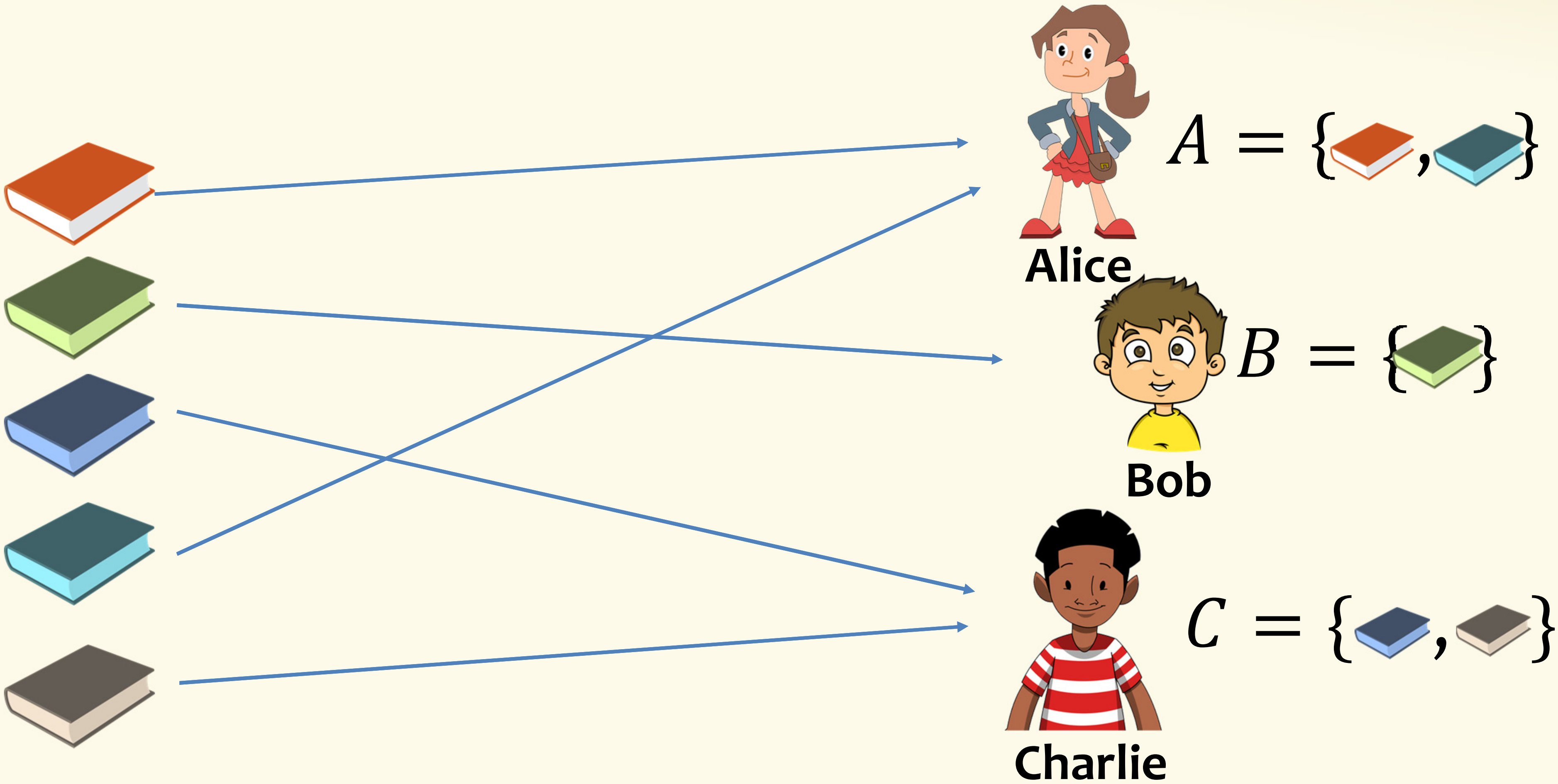
Alice



Bob

Charlie

Example Book Assignment



Book assignment – Modeling

Correct?

Poll:

A. Correct
B. Overcount
C. Undercount
D. No idea

pollev.com/paulbeame028

$2^5 = 32$ options

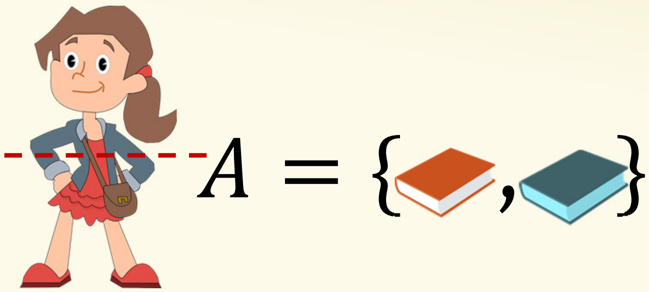
$\times$

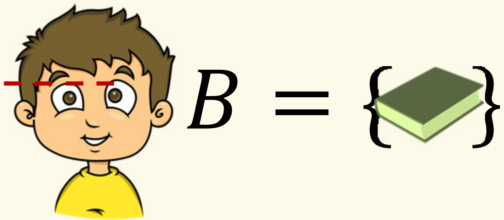
$2^5 = 32$ options

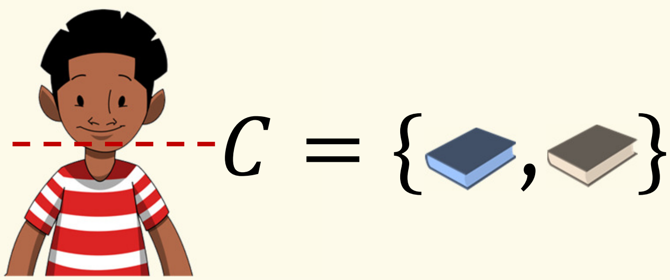
$\times$

$2^5 = 32$ options

$= 32^3$ assignments



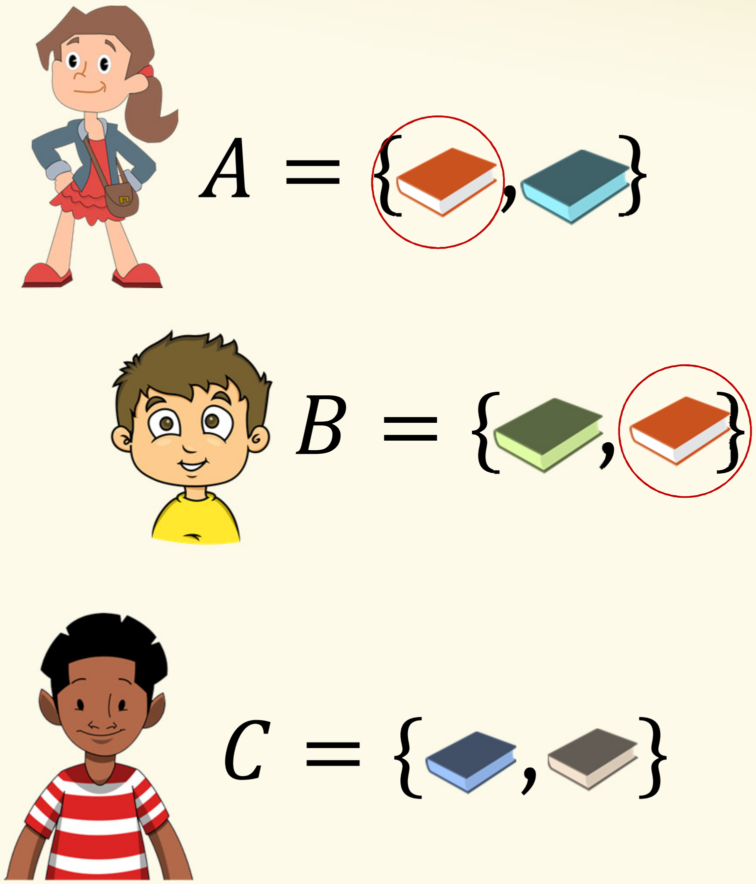




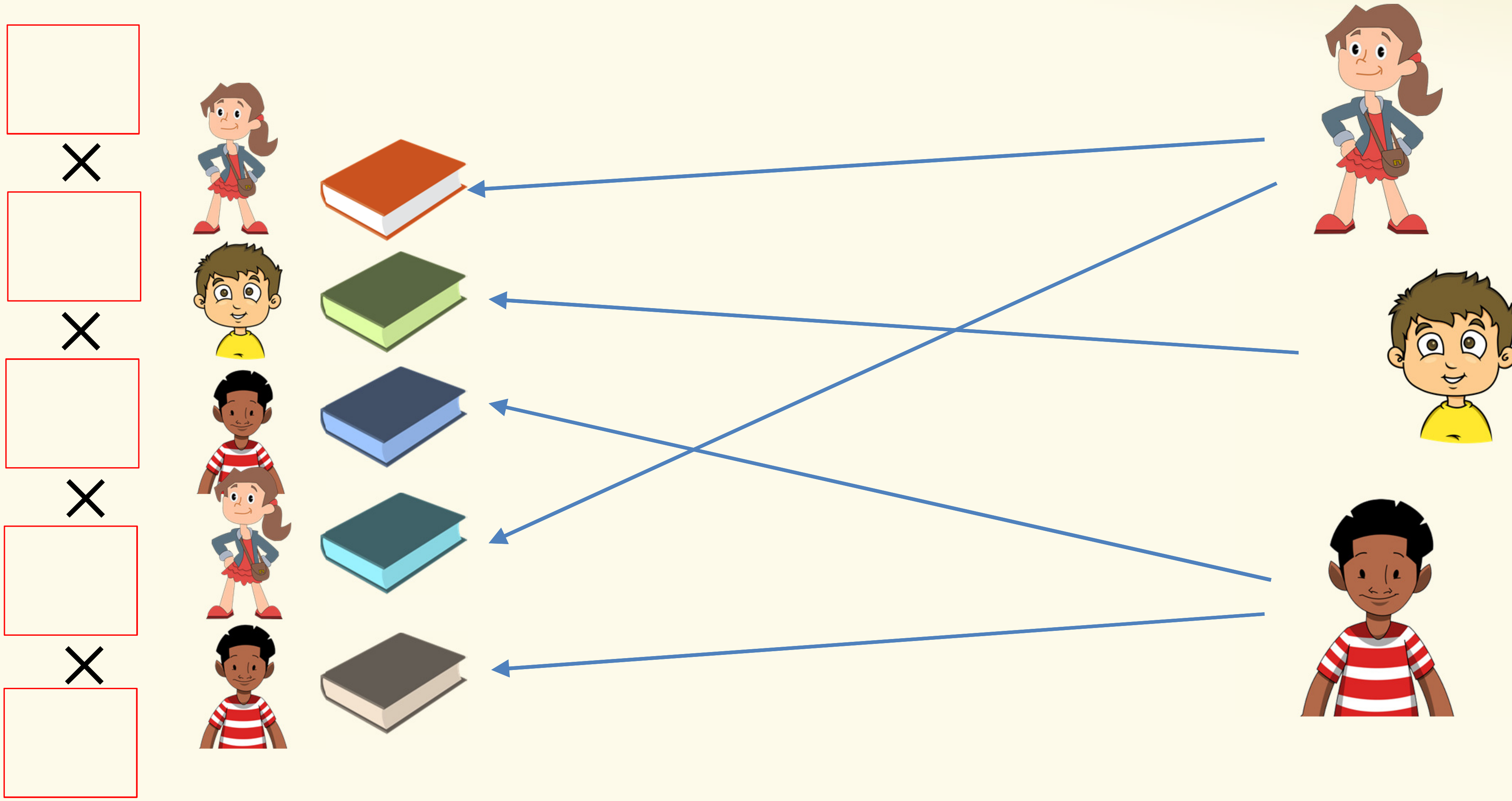
Problem – Overcounting

Problem: We are counting some <u>invalid</u> assignments!!!
→ **overcounting**!

What went wrong in the sequential process?
After assigning A to Alice,
B is no longer a valid option for Bob



Book assignments – A Clever Approach



Lesson: Representation of what we are counting is very important!

Tip: Use different methods to double check yourself
Think about counter examples to your own solution.

The first concept check is out and due at 1:00pm before Friday's lecture

The concept checks are meant to help you immediately reinforce what is learned.

Students from previous quarters have found them really useful!