CSE 20
DISCRETE MATH

WINTER 2016

http://cseweb.ucsd.edu/classes/wi16/cse20-ab/

Today's learning goals

- Explain the steps in a proof by (strong) mathematical induction
- Use (strong) mathematical induction to prove
  - correctness of identities and inequalities
  - properties of algorithms
  - properties of geometric constructions
- Represent functions in multiple ways
- Define and prove properties of: domain of a function, image of a function, composition of functions
- Determine and prove whether a function is one-to-one, onto, bijective
- Apply the definition and properties of floor function, ceiling function, factorial function
- Define and compute the cardinality of a set: Finite sets, countable sets, uncountable sets
- Use functions to compare the sizes of sets
- Use functions to define sequences: arithmetic progressions, geometric progressions
- Use and prove properties of recursively defined functions and recurrence relations (using induction)
- Use and interpret Sigma notation

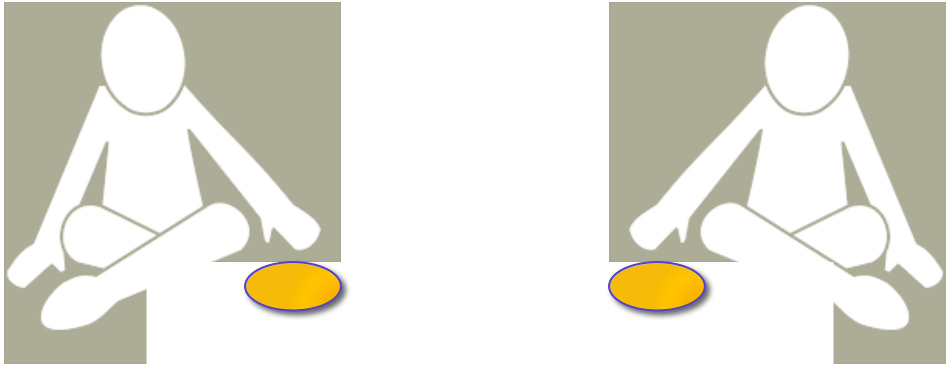
Nim

Two players take turns removing **any positive # of jellybeans** at a time from one of two piles in front of them. **The piles start out with equal #s.** The player who removes the **last jellybean wins** the game.

A. The first player has a strategy to always win.
B. The second player has a strategy to always win.
C. One of the players has a strategy to always win, but which player depends on how many jellybeans there are.
D. Who wins is random.
E. None of the above.

Nim

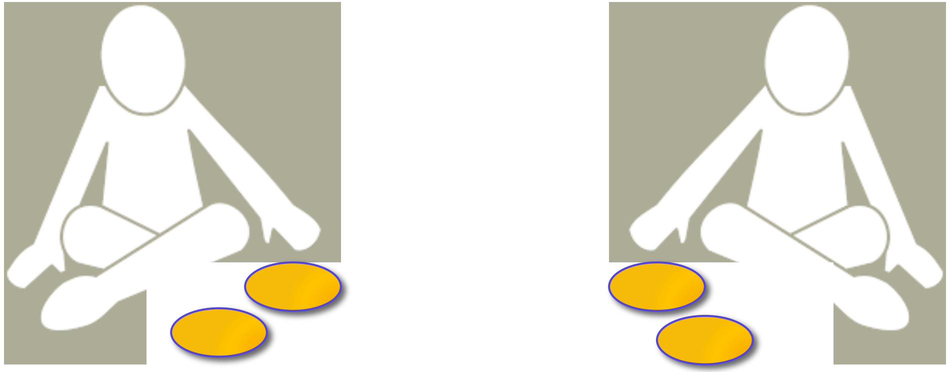
Two players take turns removing **any positive # of jellybeans**
at a time from one of two piles in front of them. **The piles start out with equal #s.**
The player who removes the **last jellybean wins** the game.



n=1 *Who wins?*

Nim

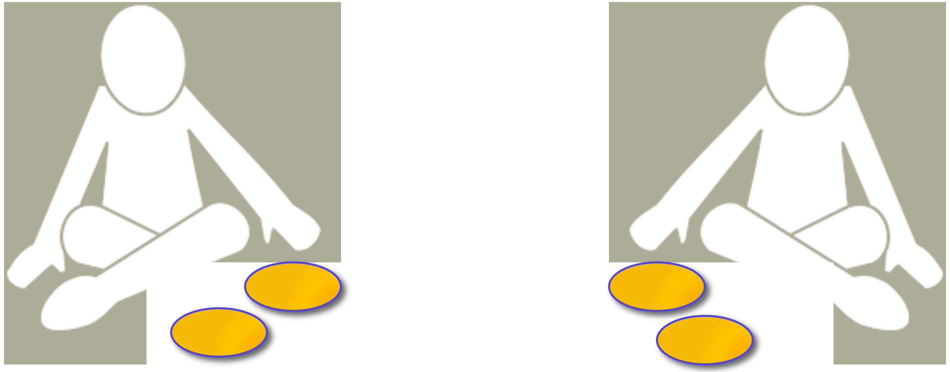
Two players take turns removing **any positive # of jellybeans** at a time from one of two piles in front of them. **The piles start out with equal #s.** The player who removes the **last jellybean wins** the game.



n=2 *Who wins?*

Nim

Two players take turns removing **any positive # of jellybeans** at a time from one of two piles in front of them. **The piles start out with equal #s.** The player who removes the **last jellybean wins** the game.



Idea: 2nd player takes the same amount 1st player took but from opposite pile.

*…Game reduces to **same setup** but with fewer jellybeans.*

Strong induction

Rosen p. 334

To show that some statement P(n) is true about all positive integers n,

1. Verify that P(1) is true.
2. Let k be an arbitrary positive integer. Show that

$$[P(1) \wedge P(2) \wedge \cdots \wedge P(k)] \rightarrow P(k+1)$$

is true.

Strong induction hypothesis

Nim

Two players take turns removing **any positive # of jellybeans** at a time from one of two piles in front of them. **The piles start out with equal #s.** The player who removes the **last jellybean wins** the game.

Theorem: the second player can always guarantee a win.

Proof: By Strong Mathematical Induction, on # jellybeans in each pile.

1. Basis step WTS if piles each have 1, then 2nd player can win.
2. Strong Induction hypothesis Let k be a positive integer. Assume that 2nd player can win whenever there are j jellybeans in each pile, for each j between 1 and k (inclusive).
3. Induction step WTS 2nd player has winning strategy when start with k+1 jellybeans in each pile.

Fibonacci numbers

Rosen p. 158, 347

$$f_0 = 1,\ f_1 = 1,\ f_n = f_{n-1} + f_{n-2}$$

Fibonacci numbers

Rosen p. 158, 347

$$f_0 = 1,\ f_1 = 1,\ f_n = f_{n-1} + f_{n-2}$$

Theorem: For each integer n>=2, f_n >= 1.5^{n-2}

Proof: By Strong Mathematical Induction, on n>=2.

1. Basis step WTS f_2 >= 1.5^{2-2}.
2. Strong Induction hypothesis Let k be an integer, k>=2. Assume inequality is true for each **integer j, 2<=j<=k.**
3. Induction step WTS statement is true about f_{k+1}.

Fibonacci numbers

Rosen p. 158, 347

$$f_0 = 1, \; f_1 = 1, \; f_n = f_{n-1} + f_{n-2}$$

1. Basis step WTS $f_2 >= 1.5^{2-2}$.

LHS = f_2 = 1 + 1 = 2.

RHS = $1.5^{2-2} = 1.5^0 = 1$.

Since 2>1, LHS>RHS so, in particular, LHS>= RHS ☺

Fibonacci numbers

Rosen p. 158, 347

$$f_0 = 1,\ f_1 = 1,\ f_n = f_{n-1} + f_{n-2}$$

Induction step Let k be an integer with k>=2.

Assume as the **strong induction hypothesis** that

$$f_j >= 1.5^{j-2}$$

for each integer j with 2<=j<=k.

WTS that $\mathbf{f_{k+1} >= 1.5^{(k+1)-2}}$

By definition of Fibonacci numbers, since k+1 > 1, $f_{k+1} = f_k + f_{k-1}$.

Therefore, LHS = $f_{k+1} = f_k + f_{k-1}$.

Idea: apply strong induction hypothesis to k and k-1. Can we do it?

Fibonacci numbers

Rosen p. 158, 347

…

Case 1: k=2 and WTS that $\mathbf{f_3 >= 1.5^{(3)-2}}$

Case 2: k>2 and WTS that $\mathbf{f_{k+1} >= 1.5^{(k+1)-2}}$ and strong IH applies to k, k-1 (because both k, k-1 are greater than or equal to 2 and less than k+1).

So let's prove each of these cases in turn:

Case 1: k=2 and WTS that $\mathbf{f_3 >= 1.5^{(3)-2}}$

By definition of Fibonacci numbers, LHS = $f_3=f_2+f_1=2+1=3$.

By algebra, RHS = $1.5^{3-2}=1.5$ Since 3>1.5, LHS>RHS ☺

Fibonacci numbers

Rosen p. 158, 347

…

Case 2: k>2 and WTS that $\mathbf{f_{k+1} >= 1.5^{(k+1)-2}}$ and strong IH applies to k, k-1 (because both k, k-1 are greater than or equal to 2 and less than k+1).

LHS = $f_{k+1} = f_k + f_{k-1} >= 1.5^{k-2} + 1.5^{(k-1)-2} = 1.5^{k-3}(1.5+1) = 1.5^{k-3}(2.5)$

$> 1.5^{k-3}(2.25) = 1.5^{k-3}1.5^2 = 1.5^{k-1} = 1.5^{(k+1)-2}$ = RHS.

Def of Fibonacci numbers

Strong induction hypothesis

Fulfilling promises

- We now have all the tools we need to rigorously prove
  - Correctness of **greedy change-making algorithm** with quarters, dimes, nickels, and pennies *Proof by contradiction, Rosen p. 199*
  - The **division algorithm** is correct *Strong induction, Rosen p. 341*
  - **Russian peasant multiplication** is correct *Induction*
  - Largest **n-bit binary** number is 2^n-1 *Induction, Rosen p. 318*
  - Correctness of **base b conversion** (Algorithm 1 of 4.2), *Strong induction*
  - Size of the **power set** of a finite set with n elements is 2^n *Induction, Rosen p. 323*
  - Any int greater than 1 can be written as **product of primes** *Strong induction, Rosen p. 323*
  - There are infinitely many **primes** *Proof by contradiction, Rosen p. 260*
  - **Sum** of geometric progressions $\sum_{j=0}^{n} ar^j = \frac{ar^{n+1} - a}{r-1}$ when $r \neq 1$, *Induction, Rosen p. 318*

Cautionary tales

- The **basis step** is absolutely necessary … and might need more than one!
- Make sure to stay in the **domain**.

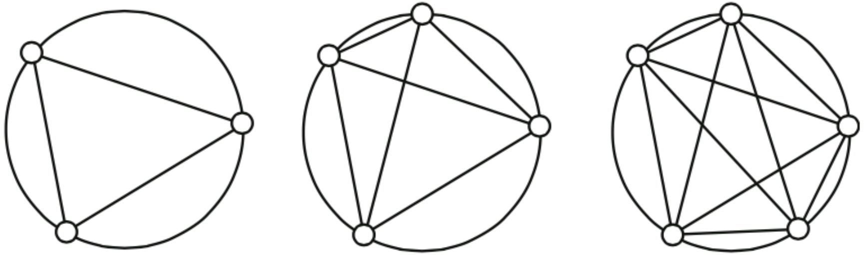
  Recommended practice

  Section 5.1 #49, 50, 51

  Section 5.2 #32

- A few **examples** do not guarantee a pattern:

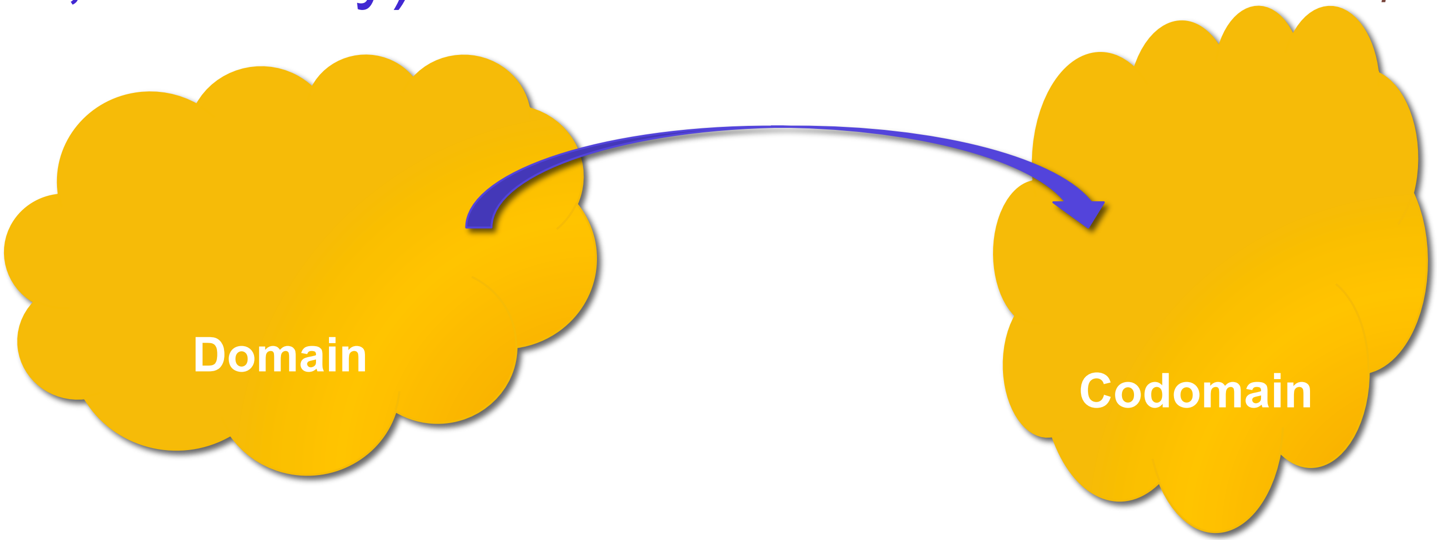
  cake cutting conundrum. Join all pairs of points among N marked on circumference of cake.



N=3 N=4 N=5

And now for something completely different …
(well, not really)

Rosen p. 138



Function **Mapping** **Transformation**

$$\forall a(a \in D \rightarrow \exists! b\,(b \in C \wedge f(a) = b))$$

To specify a function

(1) Domain (2) Codomain (3) Assignment

All examples below have domain = codomain = **N**

- Formula $f(n) = n^2$, $f(n) = \lfloor \sqrt{n} \rfloor$, $f(n) = \lceil \log_2(n+1) \rceil$
- Recursive definition

$$f(n) = \begin{cases} 0 & \text{if } n = 0 \\ nf(n-1) & \text{if } n > 0 \end{cases}, \quad f(n) = \begin{cases} n/2 & \text{if } n \textbf{ mod } 2 = 0 \\ 3n+1 & \text{if } n \textbf{ mod } 2 = 1 \end{cases}$$

With this basis step, f(n) is the constant zero function. If we change the output at input 0 to be 1, get the factorial function.

To specify a function

(1) Domain (2) Codomain (3) Assignment

If domain and codomain happen to be finite…

- Formula
- Table / diagram of values
- Relation: set of (input,output pairs)

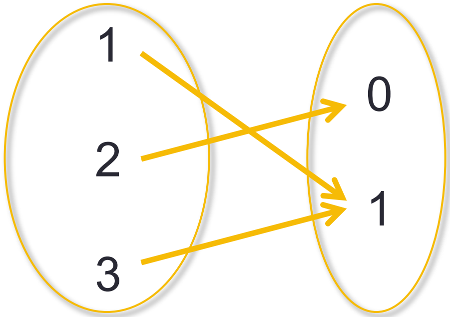
To specify a function

(1) D = {1,2,3} (2) C = {0,1} (3) Assignment

Which of the following specifications of a function is not equal to the rest?

A. f(x) = x **mod** 2

B.



C. f(x) = 2-x

D. {(1,1), (2,0), (3,1)}

E. None of the above (they're all equal)

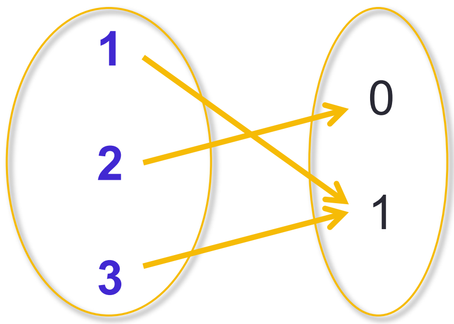
Properties of functions

- A function f is **well-defined** means **exactly one image for every input**

$$\forall a(a \in D \to \exists! b\,(b \in C \land f(a) = b))$$

- Two functions f_1, f_2 are **equal** means (1) they have the same domain, and (2) they have the same codomain, and (3)

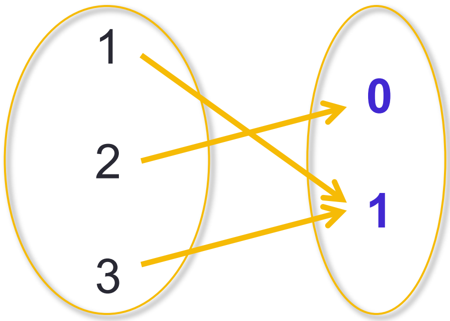
$$\forall a(a \in D \to f_1(a) = f_2(a))$$



Properties of functions

possible

- A function f is ……….. means **at least one input for every output**



How can we formalize this?

A. $\forall a(a \in D \rightarrow \exists! b\,(b \in C \wedge f(a) = b))$

B. $\forall b(b \in C \rightarrow \exists! a\,(a \in D \wedge f(a) = b))$

C. $\forall a \forall b((a \in D \wedge b \in C) \rightarrow f(a) = b)$

D. $\forall b(b \in C \rightarrow \exists a\,(a \in D \wedge f(a) = b))$

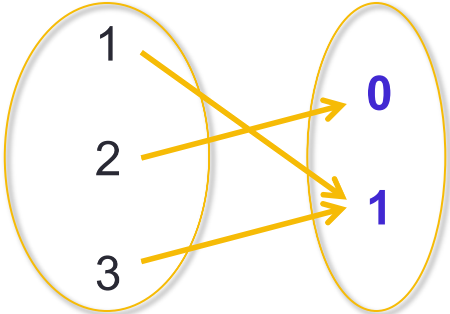
E. None of the above

Properties of functions

Rosen p. 143

possible

- A function f is **onto** (surjective) means **at least one input for every output**

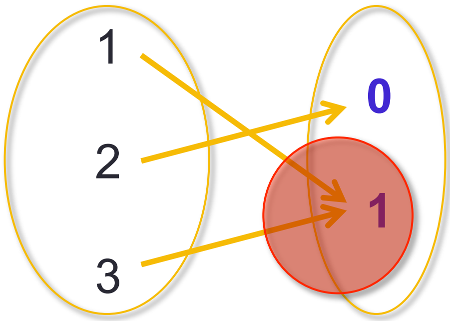


$$\forall b(b \in C \rightarrow \exists a\,(a \in D \wedge f(a) = b))$$

Properties of functions

Rosen p. 141

- A function f is **one-to-one** means **no duplicate images** (injective)



How can we formalize this?

A. $\forall a \forall b((a \in D \wedge b \in D) \rightarrow f(a) \neq f(b))$

B. $\forall a \forall b((a \in D \wedge b \in D) \rightarrow (f(a) = f(b) \rightarrow a = b))$

C. $\forall a \forall b((a \in C \wedge b \in C) \rightarrow a \neq b)$

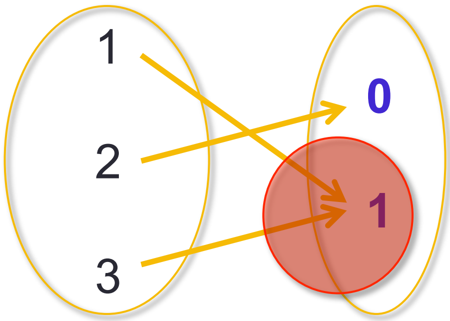
D. $\forall a \forall b((a \in C \wedge b \in C) \rightarrow f(a) \neq f(b))$

E. None of the above

Properties of functions

Rosen p. 141

- A function f is **one-to-one** means **no duplicate images** (injective)

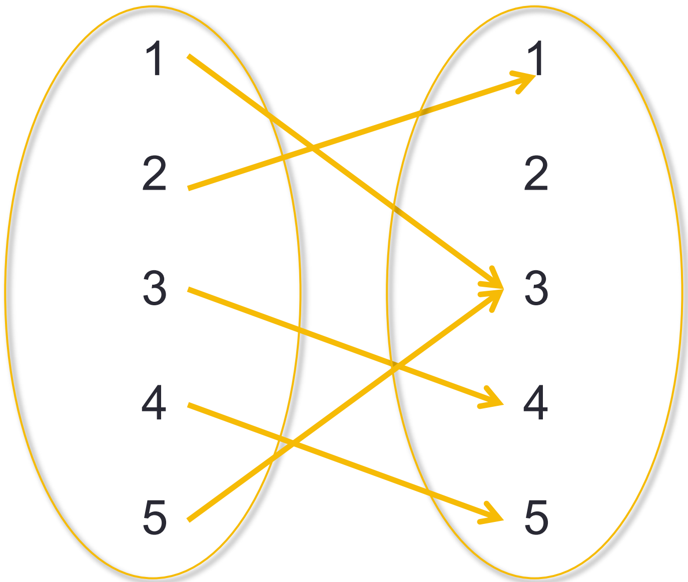


$$\forall a \forall b((a \in D \land b \in D) \rightarrow (f(a) = f(b) \rightarrow a = b))$$

$$\forall a \forall b((a \in D \land b \in D) \rightarrow (a \neq b \rightarrow f(a) \neq f(b)))$$

Onto? One-to-one?

Consider the function over domain and codomain {1,2,3,4,5} defined by



This function is

A. Well defined, onto, and one-to-one.
B. Well defined, but neither onto nor one-to-one.
C. Well defined, onto, but not one-to-one.
D. Not well-defined, not onto, not one-to-one.
E. None of the above.

Onto? One-to-one?

Consider the function over domain and codomain $\mathbf{R}^{>=0}$ defined by

$f(x) = x^2$

This function is

A. Well defined, onto, and one-to-one.
B. Well defined, but neither onto nor one-to-one.
C. Well defined, onto, but not one-to-one.
D. Not well-defined, not onto, not one-to-one.
E. None of the above.

Proving a function is …

Define f:{0,1}* → **N** by f(w) = |w|, or formally, f is defined recursively by

$$\begin{cases} f(\lambda) = 0 \\ f(w0) = f(w) + 1 \\ f(w1) = f(w) + 1 \end{cases}$$

Fact: This function is onto.

Proving a function is …

Define f:{0,1}* → **N** by f(w) = |w|, or formally, f is defined recursively by

$$\begin{cases} f(\lambda) = 0 \\ f(w0) = f(w) + 1 \\ f(w1) = f(w) + 1 \end{cases}$$

Fact: This function is not one-to-one.

Proving a function is …

Let A = {1,2,3} and B = {2,4,6}.

Define a function from the power set of A to the power set of B by:

$$f : \mathcal{P}(A) \to \mathcal{P}(B)$$
$$f(X) = X \cap B$$

Well-defined?

Onto?

One-to-one?

Reminder

- Exam 2 is next class: **Tuesday Feb 23**
  - Review sessions this weekend.
  - Extra office hours available.
  - Practice exam available on class website.
  - One note sheet allowed.
  - Seat map will be posted on class website.