Induction, Inductive Sets, and Matrices

Dr. Isaac Griffith

Idaho State University

Induction

CS 1187

Mathematical Induction

- Many theorems state that $P(n)$ is true for all positive integers n, where $P(n)$ is a propositional function, such as the statement that $1 + 2 + \ldots + n = \frac{n(n+1)}{2}$ or the statement that $n \leq 2^n$

- **Mathematical Induction** is a technique for proving theorems of this kind

- In other words, mathematical induction is used to prove propositions of the form: $\forall_x P(x)$, where the universe of discourse is the set of positive integers.

Mathematical Induction

Principle of Mathematical Induction

- A proof by mathematical induction that $P(x)$ is true for every positive integer n consists of two steps:
  1. **Basis Step:** The proposition $P(1)$ is shown to be true.
  2. **Inductive Step:** The implication $P(k) \to P(k+1)$ is shown to be true for every positive integer k
- Here, the statement $P(k)$ for a fixed positive integer k is called the **inductive hypothesis**
- When we complete both steps of a proof by mathematical induction, we have proved that $P(n)$ is true for all positive integers n; that is we have shown that $\forall_n P(n)$ is true.
- Expressed as a rule of inference, this proof technique can be stated as:

$$[P(1) \wedge \forall_k (P(k)) \to P(k+1))] \to \forall_n P(n)$$

Mathematical Induction

Idaho State University | Computer Science

$$[P(1) \wedge \forall_k (P(k)) \rightarrow P(k+1))] \rightarrow \forall_n P(n)$$

- To prove $\forall n.\ P(n)$ is true $\forall n \in \mathbb{Z}^+$:
  1. Show that $P(1)$ is true.
     - This amounts to showing that the particular statement obtained when n is replaced by 1 in $P(n)$ is true.
  2. Show that $P(k) \rightarrow P(k+1)$ is true for every positive integer k.

     2.1 To prove that this implication is true for every positive integer k we need to show that $P(k+1)$ cannot be false when $P(k)$ is true.

     2.2 Assume that $P(k)$ is true.

     2.3 Show that **under this hypothesis** $P(k+1)$ must also be true.

ROAR

Proof Examples

Example: $P(n) : 2^n < n!$ for $n \geq 4$
Proof:

Basis Step: $P(4) : 2^4 = 16 < 24 = 4!$ **true**
Inductive Step: Assume $P(k)$ is true $(k \geq 4)$

Multiply both sides by 2

$$\begin{aligned} 2 \cdot 2^k &< 2 \cdot k! \\ &< (k+1) \cdot k! \\ &= (k+1)! \end{aligned}$$

Example: $P(n) : 4n < (n^2 - 7)$ for $n \geq 6$

Proof:

Basis Step: $P(6) : 24 < 29$ **true**
Inductive Step: Assume $P(k)$ is true. $(k \geq 6)$

We want to show that $4(k+1) < (k+1)^2 - 7$

$$\begin{array}{rclcl} 4k & < & (k^2 - 7) & & \\ 4k + 4 & < & (k^2 - 7) + 4 & < & (k^2 - 7) + (2k + 1) \\ & & & = & k^2 + 2k + 1 - 7 \\ 4(k+1) & & & = & (k+1)^2 - 7 \end{array}$$

Proof Examples

Idaho State University | Computer Science

Example: $P(n) : 1 + 2 + 2^2 + \ldots + 2^n = 2^{n+1} - 1$ and $n \geq 0$

Proof:

Basis Step: $P(0)$: $2^0 = 1 = 1 = 2^{0+1} - 1$ **True**

Inductive Step: Assume $P(k)$ is true. $(k \geq 0)$

We want to show that

$$1 + 2 + 2^2 + \ldots + 2^k + 2^{k+1} = 2^{(k+1)+1} - 1 = 2^{k+2} - 1$$

$$\begin{aligned} 1 + 2 + 2^2 + \ldots + 2^k + 2^{k+1} &= (1 + 2 + 2^2 + \ldots + 2^k) + 2^{k+1} \\ &= 2^{k+1} - 1 + 2^{k+1} \\ &= 2 \cdot 2^{k+1} - 1 \\ &= 2^{k+2} - 1 \end{aligned}$$

Induction on Lists

- **Principle of List Induction:** supose $P(xs)$ is a predicate on lists of type `[a]`, for some type `a`.
  - The Base Case is to Suppose that $P([])$ is true
  - Further, suppose that if $P(xs)$ holds for arbitrary `xs :: [a]`, then $P(x : xs)$ also holds for arbitrary `x :: a`.
  - Then, $P(xs)$ holds for every list xs that has finite length
- **Example:** `length (map f xs) = length xs`
  - **Proof:** Induction over `xs`
    - **Base Case:**
      $$\begin{aligned} & \mathit{length}\ (\mathit{map}\ f\ []) \\ &= \mathit{length}\ [] && \{\ \mathit{map}.1\ \} \end{aligned}$$
    - **Inductive Case:** assume `length (map f xs) = length xs`. Then
      $$\begin{aligned} & \mathit{length}\ (\mathit{map}\ f\ (x : xs)) \\ &= \mathit{length}\ (f\ x\ :\ \mathit{map}\ f\ xs) && \{\ \mathit{map}.2\ \} \\ &= 1 + \mathit{length}\ (\mathit{map}\ f\ xs) && \{\ \mathit{length}.2\ \} \\ &= 1 + \mathit{length}\ xs && \{\ \mathit{hypothesis}\ \} \\ &= \mathit{length}\ (x\ :\ xs) && \{\ \mathit{length}.2\ \} \end{aligned}$$

Functional Equality

- If two algorithms, defined as functions, are applied to the same arguments they will produce the same result
  - If true, it may seem that we could state $f = g$, when f and g are functions.
  - But, what does $f = g$ mean?

- **Intensional Equality:** Two functions f and g are *intensionally equal* if their definitions are identical.
  - For programs this means that the source code is identical

- **Extensional Equality:** Two functions f and g are *extensionally equal* if the have the same type $a \rightarrow b$ and $f(x) = g(x)$ for all well typed arguments $x : a$. That is, $f = g$ iff

$$\forall x : a.\ f(x) = g(x)$$

  - Proof of this simply requires that we prove the proposition $\forall x : a.\ f(x) = g(x)$, by selecting an arbitrary $x : a$ and proving the equation $f(x) = g(x)$

Strong Induction

- **Strong Induction:** To prove that $P(n)$ is true for all positive integers n, where $P(n)$ is a propositional function, we complete two steps:
  - **Basis Step:** We verify that the proposition $P(1)$ is true
  - **Inductive Step:** We show that the conditional statement $[P(1) \wedge P(2) \wedge \ldots \wedge P(k)] \rightarrow P(k+1)$ is true for all positive integers k
    - That is, here we show that fall all positive integers j not exceeding k, then $P(k+1)$ is true
- For our *inductive hypothesis*, we assume $P(j)$ is true for $j = 1, 2, \ldots, k$
- **Well-Ordering Property:** Every nonempty set of nonnegative integers has at least one element.

Example

- Prove that every amount of postage of 12 cents or more can be formed using just 4-cent and 5-cent stamps
  - Let $P(n)$ be the statement that postage of n cents can be formed using 4-cent and 5-cent stamps
- **Basis Step:** We can form postage of 12, 13, 14, and 15 cents as follows:
  - P(12) - three 4-cent stamps
  - P(13) - two 4-cent stamps
  - P(14) - one 4-cent stamps and two 5-cent stamps
  - P(15) - three 5-cent stamps
- **Inductive Step:** Assume we can form postage of j cents, where $12 \leq j \leq k$
  - We need to show that under the assumption $P(k+1)$ is true, we can also form postage of $k+1$ cents.
  - We can assume that $P(k-3)$ is true because $k-3 \geq 12$
  - To form postage of k + 1 cents, we need only add another 4-cent stamp to the stamps used for $k-3$ cents.
  - Thus, we've shown the *inductive hypothesis* is true, then $P(K+1)$ is also true

Defining Sets Inductively

CS 1187

Defining Sets Using Induction

- Beyond the base and inductive cases, inductive set definition needs one more component: the *extremal clause*

- **Extremal Clause:** A statement which excludes anything from the set that are not introduced by the base case, or are instantiations of the induction case, it reads something like the following:

  "Nothing is an element of the set unless it can be constructed by a finite number of uses of the first two clauses"

- Thus all inductive set definitions include 3 parts:
  - **Base Case:** a simple statement of some mathematical fact: i.e., $1 \in S$
  - **Induction Case:** an implication in a general form: $\forall\, x \in U,\ x \in S \rightarrow x + 1 \in S$
  - **Extremal Clause:** Nothing is in the set being defined unless it got there by a finite number of uses of the first two cases

The Natural Numbers

- The set of natural numbers, $\mathbb{N}$, is defined as follows
  - **Base Case:** $0 \in \mathbb{N}$
  - **Induction case:** $x \in \mathbb{N} \rightarrow x + 1 \in \mathbb{N}$
  - **Extremal clause:** nothing is an element of the set $\mathbb{N}$ unless it can be constructed with a finite number of uses of the base and induction cases.

- We can show that an arbitrary number above and including 0 are in $\mathbb{N}$

$$
\begin{array}{llr}
1. & 0 \in \mathbb{N} & \text{Base Case} \\
2. & 0 \in \mathbb{N} \rightarrow 1 \in \mathbb{N} & instantiationrule, inductioncase \\
3. & 1 \in \mathbb{N} & \text{1, 2, Modus Ponens} \\
4. & 1 \in \mathbb{N} \rightarrow 2 \in \mathbb{N} & \text{instantiation rule, induction case} \\
5. & 2 \in \mathbb{N} & \text{3, 4, Modus Ponens}
\end{array}
$$

Binary Machine Words

- Let *BinDigit* be the set $\{0, 1\}$. The set *BinWords* of machine words in binary is defined as follows:
  - **Base Case:** $x \in \texttt{BinDigit} \rightarrow x \in \texttt{BinWords}$
  - **Induction Case:** if x is a binary digit and y is a binary word, then their concatenation xy is also a binary word

$$(x \in \texttt{BinDigit} \land y \in \texttt{BinWords}) \rightarrow xy \in \texttt{BinWords}$$

  - **Extremal Clause:** Nothing is an element of `BinWords` unless it can be constructed with a finite number of uses of the base and induction cases

- A set based on another set S in this way is given the name S^+
  - it is the set of all possible non-empty strings over S
  - S^* is similar to S^+ except S^* includes the empty string
  - `BinWords` could have also been written as $\texttt{BinDigit}^+$

Haskell Implementation

Idaho State University | Computer Science

- We can define a function to create two new `BinWords` based on one that has been provided
  - i.e., given `[1, 0]` it will return `[0, 1, 0]` and `[1, 1, 0]`

```
newBinaryWords :: [Integer] -> [[Integer]]
newBinaryWords ys = [0:ys, 1:ys]
```

- We then define the set of `BinWords` as:

```
mappend :: (a -> [b]) -> [a] -> [b]
mappend f []     = []
mappend f (x:xs) = f x ++ mappend f xs

binWords = [0] : [1] : (mappend newBinaryWords binWords)
```

The Set of Integers

- Both of the prior sets are **well-founded**, meaning they are infinite in only one direction, and they have a *least* element
- **Countable Set:** a set which can be counted using the natural numbers
  - Are the integers countable?
    - Doesn't have a least element
    - Infinite in two directions
  - However we can count hem using natural numbers as follows:
    - Start at 0
    - For every number $n \in \mathbb{N}$, we count both n and $-n$ in $\mathbb{Z}$
  - That is, we can consider the set of integers as an infinitely long tape folded in half at 0, and then count the overlapping numbers $(i, -i)$ for each $i \in \mathbb{N}$
- Yet, this does not specify $\mathbb{Z}$

The Set of Integers

- The set $\mathbb{Z}$ of integer is defined as follows:
  - **Base Case:** $0 \in \mathbb{Z}$
  - **Induction Case:** $(x \in \mathbb{Z} \wedge x \geq 0) \rightarrow x + 1 \in \mathbb{Z} \wedge -(x+1) \in \mathbb{Z}$
  - **Extremal Clause:** nothing is in $\mathbb{Z}$ unless its presence is justified by a finite number of uses of the base and induction cases

Thus, we can define integers using Haskell, as follows

```haskell
build :: a -> (a -> a) -> Set a
build a f = set
    where set = a : map f set

builds :: a -> (a -> [a]) -> Set a
builds a f = set
    where set = a : mappend f set

nextInteger :: Integer -> [Integer]
nextInteger x
  = if x > 0 \/ x == 0
      then [x + 1, -(x + 1)]
      else []

integer :: [Integer]
integers = builds 0 next Integers
```

Matrices

CS 1187

ROAR

Matrices

Idaho State University | Computer Science

- **Matrix:** a rectangular array of numbers.
  - Matrix with *m* rows and *n* columns is called an $m \times n$ matrix
  - Matrix with the same number of rows and columns is called *square*.
  - Two matrices are equal if they have the same number of rows and the same number of columns and the corresponding entries in every position are equal
- **Example:** a 3×2 matrix

$$\begin{bmatrix} 1 & 1 \\ 0 & 2 \\ 1 & 3 \end{bmatrix}$$

Matrices

Let m and n be positive integers and let

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}$$

The ith *row* of $\mathbf{A}$ is the $1 \times n$ matrix $[a_{i1}, a_{i2}, \dots, a_{in}]$. The jth *column* of $\mathbf{A}$ is the $m \times 1$ matrix

$$\begin{bmatrix} a1j \\ a_{2j} \\ \cdot \\ \cdot \\ \cdot \\ a_{mj} \end{bmatrix}$$

- The (i, j)th *element* or *entry* of $\mathbf{A}$ is the element a_{ij}, that is, the number in the ith row and jth column of $\mathbf{A}$. A convenient shorthand notation for expressing the matrix $\mathbf{A}$ is to write $\mathbf{A} = [a_{ij}]$, which indicates that $\mathbf{A}$ is the matrix with its (i, j)th element equal to a_{ij}

Matrix Sums

Let $\mathbf{A} = [a_{ij}]$ and $\mathbf{B} = [b_{ij}]$ be $m \times n$ matrices:

- **Sum:** the *sum* of $\mathbf{A}$ and $\mathbf{B}$, denoted $\mathbf{A} + \mathbf{B}$, is the $m \times n$ matrix that has $a_{ij} + b_{ij}$ as its (i, j)th element.
  - That is, $\mathbf{A} + \mathbf{B} = [a_{ij} + b_{ij}]$

$$\begin{bmatrix} 1 & 0 & -1 \\ 2 & 2 & -3 \\ 3 & 4 & 0 \end{bmatrix} + \begin{bmatrix} 3 & 4 & -1 \\ 1 & -3 & 0 \\ - & 1 & 2 \end{bmatrix} = \begin{bmatrix} 4 & 4 & -2 \\ 3 & -1 & -3 \\ 2 & 5 & 2 \end{bmatrix}$$

Matrix Multiplication

Idaho State University | Computer Science

- Let **A** be an $m \times k$ matrix and **B** be a $k \times n$ matrix. The *product* of **A** and **B**, denoted by **AB**, is the $m \times n$ matrix with its (i, j)th entry equalt to the sum of the products of the corresponding elements from the ith row of **A** and the jth column of **B**. That is, if $\mathbf{AB} = [c_{ij}]$, then

$$c_{ij} = a_{i1}b_{1j} + a_{i2}b_{2j} + \ldots + a_{ik}b_{kj}$$

$$\begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1k} \\ a_{21} & a_{22} & \ldots & a_{2k} \\ \vdots & \vdots & & \vdots \\ \mathbf{a_{i1}} & \mathbf{a_{i2}} & \ldots & \mathbf{a_{ik}} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \ldots & a_{mk} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} & \ldots & \mathbf{b_{1j}} & \ldots & b_{1n} \\ b_{21} & b_{22} & \ldots & \mathbf{b_{2j}} & \ldots & b_{2n} \\ \vdots & \vdots & & \vdots & & \vdots \\ b_{k1} & b_{k2} & \ldots & \mathbf{b_{kj}} & \ldots & b_{kn} \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & \ldots & c_{1n} \\ c_{11} & c_{12} & \ldots & c_{1n} \\ \vdots & \vdots & \mathbf{c_{ij}} & \vdots \\ c_{m1} & c_{m2} & \ldots & c_{mn} \end{bmatrix}$$

Example

Let,

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 4 \\ 2 & 1 & 1 \\ 3 & 1 & 0 \\ 0 & 2 & 2 \end{bmatrix} \quad \text{and} \quad \mathbf{B} = \begin{bmatrix} 2 & 4 \\ 1 & 1 \\ 3 & 0 \end{bmatrix}$$

$$\mathbf{AB} = \begin{bmatrix} (1\cdot 2)+(0\cdot 1)+(4\cdot 3) & (1\cdot 4)+(0\cdot 1)+(4\cdot 0) \\ (2\cdot 2)+(1\cdot 1)+(1\cdot 3) & (2\cdot 4)+(1\cdot 1)+(1\cdot 0) \\ (2\cdot 3)+(1\cdot 1)+(0\cdot 3) & (3\cdot 4)+(1\cdot 1)+(0\cdot 0) \\ (0\cdot 2)+(2\cdot 1)+(2\cdot 3) & (0\cdot 4)+(2\cdot 1)+(2\cdot 0) \end{bmatrix}$$

$$\mathbf{AB} = \begin{bmatrix} 14 & 4 \\ 8 & 9 \\ 7 & 13 \\ 8 & 2 \end{bmatrix}$$

Matrix Identity and Powers

- **Identity Matrix of Order n ($\mathbf{I}_n$):** is the $n \times n$ matrix $\mathbf{I}_n = [\delta_{ij}]$, (the *Kronecker delta*) where $\delta_{ij} = 1$ if $i = j$ and $\delta_{ij} = 0$ if $i \neq j$. Hence

$$\mathbf{I}_n = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ 0 & 0 & \dots & 1 \end{bmatrix}$$

- Multiplying a matrix by its identity matrix does not change the matrix: $\mathbf{A}\mathbf{I}_n = \mathbf{I}_m\mathbf{A} = \mathbf{A}$

- Powers of a square matrix can be defined because matrix multiplication is associative:
  - $\mathbf{A}^0 = \mathbf{I}_n$
  - $\mathbf{A}^r = \mathbf{A}\mathbf{A}\mathbf{A}\dots\mathbf{A}$

Transpose and Symmetry

- **Transpose:** Let $\mathbf{A} = [a_{ij}]$ be an $m \times n$ matrix. The *transpose* of $\mathbf{A}$, denoted by $\mathbf{A}^T$, is the $n \times m$ matrix obtained by interchanging the rows and columns of $\mathbf{A}$.
  - That is, if $\mathbf{A}^T = [b_{ij}]$, then $b_{ij} = a_{ji}$ for $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, m$

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \quad \mathbf{A}^T = \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}$$

- **Symmetric:** A square matrix $\mathbf{A}$ is called *symmetric* if $\mathbf{A} = \mathbf{A}^T$. Thus, $\mathbf{A} = [a_{ij}]$ if $a_{ij} = a_{ji}$ for all i and j with $1 \leq i \leq n$ and $1 \leq j \leq n$.
  - **Example:**

$$\begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Zero-One Matrices

- **Zero-One Matrix:** a matrix all of whose entries are either 0 or 1
- Arithmetic on these matrices is base on the Boolean operations $\wedge$ and $\vee$

$$b_1 \wedge b_2 = \begin{cases} 1 & \text{if } b_1 = b_2 = 1 \\ 0 & \text{otherwise} \end{cases} \qquad b_1 \vee b_2 = \begin{cases} 1 & \text{if } b_1 = 1 \text{ or } b_2 = 1 \\ 0 & \text{otherwise} \end{cases}$$

Zero-One Matrix Arithmetic

Let $\mathbf{A} = [a_{ij}]$ and $\mathbf{B} = [b_ij]$ be $m \times n$ zero-one matrices

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \qquad \mathbf{B} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 0 \end{bmatrix}$$

Join: the *join* of $\mathbf{A}$ and $\mathbf{B}$, denoted $\mathbf{A} \vee \mathbf{B}$, is the zero-one matrix with (i, j)th entry $a_{ij} \vee b_{ij}$

Merge: the *merge* of $\mathbf{A}$ and $\mathbf{B}$, denoted $\mathbf{A} \wedge \mathbf{B}$, is the zero-one matrix with (i, j)th entry $a_{ij} \wedge b_{ij}$

$$\mathbf{A} \vee \mathbf{B} = \begin{bmatrix} 1 \vee 0 & 0 \vee 1 & 1 \vee 0 \\ 0 \vee 1 & 1 \vee 1 & 0 \vee 0 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

$$\mathbf{A} \wedge \mathbf{B} = \begin{bmatrix} 1 \wedge 0 & 0 \wedge 1 & 1 \wedge 0 \\ 0 \wedge 1 & 1 \wedge 1 & 0 \wedge 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

Zero-One Matrix Product

- Let $\mathbf{A} = [a_{ij}]$ be an $m \times n$ zero-one matrix and $\mathbf{B} = [b_{ij}]$ be a $k \times n$ zero-one matrix. Then the **Boolean product** of $\mathbf{A}$ and $\mathbf{B}$, denoted by $\mathbf{A} \odot \mathbf{B}$, is the $m \times n$ matrix with (i, j)th entry c_{ij} where

$$c_{ij} = (a_{i1} \wedge b_{1j}) \vee (a_{i2} \wedge b_{2j}) \vee \ldots \vee (a_{ik} \wedge b_{kj})$$

Example: Find the Boolean product of $\mathbf{A} \odot \mathbf{B}$

$$\mathbf{A} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

$$\begin{aligned} \mathbf{A} \odot \mathbf{B} &= \begin{bmatrix} (1 \wedge 1) \vee (0 \wedge 0) & (1 \wedge 1) \vee (0 \wedge 1) & (1 \wedge 0) \vee (0 \wedge 1) \\ (0 \wedge 1) \vee (1 \wedge 0) & (0 \wedge 1) \vee (1 \wedge 1) & (0 \wedge 0) \vee (1 \wedge 1) \\ (1 \wedge 1) \vee (0 \wedge 0) & (1 \wedge 1) \vee (0 \wedge 1) & (1 \wedge 0) \vee (0 \wedge 1) \end{bmatrix} \\ &= \begin{bmatrix} 1 \vee 0 & 1 \vee 0 & 0 \vee 0 \\ 0 \vee 0 & 0 \vee 1 & 0 \vee 1 \\ 1 \vee 0 & 1 \vee 0 & 0 \vee 0 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix} \end{aligned}$$

ROAR

For Next Time

- Review DMUC Chapters 4, 9 and 11
- Review DMA Chapters 2.3 - 2.5 and 5.1 - 5.2
- Review this Lecture
- Read DMUC Chapter 11.2.3, 11.2.4, 11.3 - 11.4
- Read DMA Chapters 2.6, 4, 5.5

Are there any questions?

ROAR