

Logic

- Crucial for mathematical reasoning
- Used for designing electronic circuitry
- Logic is a system based on propositions.
- A proposition is a statement that is either true or false (not both).
- We say that the truth value of a proposition is either true (T) or false (F).
- Corresponds to 1 and 0 in digital circuits

Fall 2002

CMSC 203 - Discrete Structures

The Sta	atement/Pro	position Game
"Elepha	ants are bigge	r than mice."
Is this a st	tatement?	yes
Is this a p	roposition?	yes
What is th of the prop	e truth value position?	true
Fall 2002	CMSC 203 - Discrete Str	uctures 3

The Sta	tement/Pro	position Game	;
	"520 < 11	1″	
Is this a st	atement?	yes	
Is this a pr	oposition?	yes	
What is the of the prop	e truth value osition?	false	
Fall 2002	CMSC 203 - Discrete S	ructures	4



The Sta	itement/Pro	position Game
	"y > 5"	
Is this a st	atement?	yes
Is this a pr	oposition?	no
l ts truth v but this val We call thi proposition	alue depends lue is not spec s type of stat al function or	on the value of y, cified. tement a r open sentence.
Fall 2002	CMSC 203 - Discrete St	ructures 5

The Sta	atement/Pro	position Game
"Today	is January 1	and 99 < 5."
Is this a st	atement?	yes
Is this a pr	oposition?	yes
What is the of the prop	e truth value position?	false
Fall 2002	CMSC 203 - Discrete Stru	uctures 6



The Stat	ement/Prop	osition Game
"l f they co	elephants wer uld hide in che	re red, erry trees."
ls this a sta	tement?	yes
ls this a pro	position?	yes
What is the of the prope	truth value sition?	probably false
Fall 2002	CMSC 203 - Discrete Structur	res 8



Combining Propositions

As we have seen in the previous examples, one or more propositions can be combined to form a single compound proposition.

We formalize this by denoting propositions with letters such as p, q, r, s, and introducing several logical operators.

CMSC 203 - Discrete Structures

10

Fall 2002

Logical Operators (Connectives) We will examine the following logical operators: Negation (NOT) Conjunction (AND) Disjunction (OR) (XOR) Exclusive or Implication (if - then) • Biconditional (if and only if) Truth tables can be used to show how these operators can combine propositions to compound propositions. Fall 2002 CMSC 203 - Discrete Structures 11





	Conjuna	ction (A	AND)	
Binary Operator, Symbol: 🔥				٨
	Р	Q	P∧Q	
	Т	Т	Т	
	Т	F	F	
	F	Т	F	
	F	F	F	
Fall 2002	CMSC :	203 - Discrete Structures		13



















S	Stat Statemer	emei nts and o way t	nts a operator o form r	ind C rs can be new state	Derator: e combined in an ements.	S y
	Р	Q	¬Ρ	¬Q	(¬P)∨(¬Q)	
	Т	Т	F	F	F	
	Т	F	F	Т	Т	
	F	Т	Т	F	Т	
	F	F	Т	Т	Т	
Fall	2002		CMSC 203 - Di	screte Structure	s	18



Statem	ents and way	l operato to form	ors can be co new statem	ombined in any ents.
Р	Q	P∧Q	¬ (P∧Q)	(¬P)∨(¬Q)
Т	Т	Т	F	F
Т	F	F	Т	Т
F	Т	F	Т	Т
F	F	F	Т	Т



Equivalent Statements					
	Р	Q	¬(P∧Q)	(¬P)∨(¬Q)	$\neg(P \land Q) \leftrightarrow (\neg P) \lor (\neg Q)$
	Т	Т	F	F	Т
	Т	F	Т	Т	Т
	F	Т	Т	Т	Т
	F	F	Т	Т	Т
The statements $\neg(P \land Q)$ and $(\neg P) \lor (\neg Q)$ are logically equivalent, since $\neg(P \land Q) \leftrightarrow (\neg P) \lor (\neg Q)$ is always true.					
Fall 2002 CMSC 203 - Discrete Structures 20			cs 20		





Tautologies and Contradictions

A contradiction is a statement that is always false.

Examples:

• R∧(¬R)

Fall 2002

• $\neg(\neg(P \land Q) \leftrightarrow (\neg P) \lor (\neg Q))$

The negation of any tautology is a contradiction, and the negation of any contradiction is a tautology.

CMSC 203 - Discrete Structures

22

ExercisesWe already know the following tautology: $\neg(P \land Q) \Leftrightarrow (\neg P) \lor (\neg Q)$ $\neg(P \land Q) \Leftrightarrow (\neg P) \land (\neg Q)$ Nice home exercise:Show that $\neg(P \lor Q) \Leftrightarrow (\neg P) \land (\neg Q)$ These two tautologies are known as De
Morgan's laws.Table 5 in Section 1.2 shows many useful laws.Exercises 1 and 7 in Section 1.2 may help you
get used to propositions and operators.

Let's Talk About Logic

- Logic is a system based on propositions.
- A proposition is a statement that is either true or false (not both).
- We say that the truth value of a proposition is either true (T) or false (F).

CMSC 203 - Discrete Structures

Corresponds to 1 and 0 in digital circuits

Logical Op	perators (Connectives	s)
 Negation Conjunction Disjunction Exclusive or Implication Biconditional 	(NOT) (AND) (OR) (XOR) (if - then) (if and only if)	
Truth tables ca operators can c compound propo	n be used to show how these ombine propositions to ositions.	
Fall 2002	CMSC 203 - Discrete Structures	25





Prop	ositional Func ⁻	tions	
Propositional f statement invo	unction (open sente living one or more va	nce): ariables,	
e.g.: x-3 > 5.			
Let us call this where P is the	propositional funct predicate and x is t	ion P(x), he variabl	e.
What is the tr	uth value of P(2) ?	false	
What is the tr	uth value of P(8) ?	false	
What is the tr	uth value of P(9) ?	true	
Fall 2002	CMSC 203 - Discrete Structures		28

Propositional Functions

Let us consider the propositional function Q(x, y, z) defined as:

x + y = z.

Here, \boldsymbol{Q} is the predicate and $\boldsymbol{x},\,\boldsymbol{y},$ and \boldsymbol{z} are the variables.

What is the truth value of Q(2, 3, 5)?trueWhat is the truth value of Q(0, 1, 2)?falseWhat is the truth value of Q(9, -9, 0)?trueFall 2002CMSC 203 - Discrete Structures29

Universal Quantification

Let P(x) be a propositional function.

Universally quantified sentence:

For all x in the universe of discourse P(x) is true.

Using the universal quantifier \forall : $\forall x P(x)$ "for all x P(x)" or "for every x P(x)"

(Note: $\forall x P(x)$ is either true or false, so it is a proposition, not a propositional function.)

CMSC 203 - Discrete Structures

Fall 2002





Existential Quantification

Example:

P(x): x is a UMBC professor. G(x): x is a genius.

What does $\exists x (P(x) \land G(x)) \text{ mean }$?

"There is an x such that x is a UMBC professor and x is a genius." or

CMSC 203 - Discrete Structures

"At least one UMBC professor is a genius."

Fall 2002

Quantification
Another example: Let the universe of discourse be the real numbers.
What does $\forall x \exists y (x + y = 320)$ mean ?
"For every x there exists a y so that $x + y = 320$."
Is it true? yes
Is it true for the natural numbers? no
Fall 2002 CMSC 203 - Discrete Structures 34

Γ





Negation

 $\neg(\forall x P(x))$ is logically equivalent to $\exists x (\neg P(x))$.

 $\neg(\exists x P(x))$ is logically equivalent to $\forall x (\neg P(x))$.

See Table 3 in Section 1.3.

I recommend exercises 5 and 9 in Section 1.3.

CMSC 203 - Discrete Structures



Set Theory

Actually, you will see that logic and set theory are very closely related.

CMSC 203 - Discrete Structures

37

Fall 2002



	Set Equality		
Sets A and E contain exac	3 are equal if and only it tly the same elements.	f they	
Examples:			
• A = {9, 2, 7	′, -3}, B = {7, 9, -3, 2} :	A = B	
• A = {dog, ca B = {cat, h	at, horse}, orse, squirrel, dog} :	A ≠ B	
• A = {dog, ca B = {cat, h	at, horse}, orse, dog, dog} :	A = B	
Fall 2002	CMSC 203 - Discrete Structures		39

Examples for Sets

"Standard" Sets:

Fall 2002

- Natural numbers **N** = {0, 1, 2, 3, ...}
- Integers **Z** = {..., -2, -1, 0, 1, 2, ...}
- Positive Integers **Z**⁺ = {1, 2, 3, 4, ...}
- Real Numbers $\mathbf{R} = \{47.3, -12, \pi, ...\}$
- Rational Numbers **Q** = {1.5, 2.6, -3.8, 15, ...} (correct definition will follow)

CMSC 203 - Discrete Structures

40





 $\mathbf{Q} = \{a/b \mid a \in \mathbf{Z} \land b \in \mathbf{Z} \land b \neq 0\}$

or

Fall 2002

And how about the set of real numbers R?

R = {r | r is a real number} That is the best we can do.

CMSC 203 - Discrete Structures

Subsets							
$\begin{array}{ll} A \subseteq B & "A \text{ is a subset of } B" \\ A \subseteq B \text{ if and only if every element of } A \text{ is also} \\ an element of B. \\ We can completely formalize this: \\ A \subseteq B \Leftrightarrow \forall x \ (x \in A \to x \in B) \end{array}$							
Examples:							
$A=\{3,9\},B=\{5,9,1,3\},$	$A \subseteq B$?	true					
$A=\{3,3,3,9\},B=\{5,9,1,3\},$	A⊆B?	true					
A = $\{1, 2, 3\}, B = \{2, 3, 4\},\$	$A \subseteq B$?	false					
Fall 2002 CMSC 203 - Discrete Structures		43					







Subsets						
Useful rules: • $\emptyset \subseteq A$ for any set A • $A \subseteq A$ for any set A						
Proper subsets: $A \subset B$ "A is a proper subset of B" $A \subset B \Leftrightarrow \forall x \ (x \in A \to x \in B) \land \exists x \ (x \in B \land x \notin A)$ or $A \subset B \Leftrightarrow \forall x \ (x \in A \to x \in B) \land \neg \forall x \ (x \in B \to x \in A)$						
Fall 2002 CMSC 203 - Discrete Structures 45						



Cardinality of Sets					
If a set S contains n distinct elements, $n \in \mathbf{N}$, we call S a finite set with cardinality n.					
Examples:					
A = {Mercedes,	BMW, Porsche},	A = 3			
B = {1, {2, 3}, {4,	B = 4				
C = Ø		C = 0			
$D = \{ x \in \mathbf{N} \mid x \leq$	7000 }	D = 7001			
E = { x∈ N x ≥	7000 }	E is infinite	ļ		
Fall 2002	CMSC 203 - Discrete Structures		46		



The Power Set	
$\begin{array}{ll} P(A) & "power set of A" \\ P(A) = \{B \mid B \subseteq A\} & (contains all subsets of A) \end{array}$	
Examples:	
	ł
$A = \emptyset$ P(A) = { \emptyset } Note: A = 0, P(A) = 1	
Fall 2002 CMSC 203 - Discrete Structures 47	

	The Power Set									
Cardinality of power sets:										
F	$ P(A) = 2^{ A }$									
•	l ma	gine	eac	h el	eme	nt i	n A	has	an "	on/off" switch
 Each possible switch configuration in A corresponds to one element in 2^A 										
	А	1	2	3	4	5	6	7	8	
	х	х	х	х	х	х	х	х	х	
	у	у	у	у	у	у	у	у	у	
	Z	Z	Ζ	Ζ	Ζ	Ζ	Z	Z	Ζ	
For 3 elements in A, there are										
	2x2	×2 =	8 e	lem	ents	in F	P(A)			
	Fall 200	02			CMSC	203 - Di	screte Sti	uctures		48

Cartesian Product

The ordered n-tuple $(a_1, a_2, a_3, ..., a_n)$ is an ordered collection of objects.

Two ordered n-tuples $(a_1, a_2, a_3, ..., a_n)$ and $(b_1, b_2, b_3, ..., b_n)$ are equal if and only if they contain exactly the same elements in the same order, i.e. $a_i = b_i$ for $1 \le i \le n$.

The Cartesian product of two sets is defined as: $A \times B = \{(a, b) \mid a \in A \land b \in B\}$ Example: A = {x, y}, B = {a, b, c} $A \times B = \{(x, a), (x, b), (x, c), (y, a), (y, b), (y, c)\}$ Full 2002 CMSC 203- Discrete Structures 49





Note that:

- $A \times \emptyset = \emptyset$
- $\emptyset \times A = \emptyset$

Fall 2002

- For non-empty sets A and B: $A \neq B \Leftrightarrow A \times B \neq B \times A$
- $|A \times B| = |A| \cdot |B|$

The Cartesian product of two or more sets is defined as:

 $A_1 \times A_2 \times ... \times A_n = \{(a_1, a_2, ..., a_n) \mid a_i \in A \text{ for } 1 \le i \le n\}$

51

CMSC 203 - Discrete Structures

Set Operations

CMSC 203 - Discrete Structures

52

Union: $A \cup B = \{x \mid x \in A \lor x \in B\}$

Example: A = {a, b}, B = {b, c, d} A \cup B = {a, b, c, d}

Intersection: $A \cap B = \{x \mid x \in A \land x \in B\}$

Example: A = {a, b}, B = {b, c, d} A \cap B = {b}

Fall 2002

Set OperationsTwo sets are called disjoint if their intersectionis empty, that is, they share no elements: $A \cap B = \emptyset$ The difference between two sets A and Bcontains exactly those elements of A that arenot in B: $A - B = \{x \mid x \in A \land x \notin B\}$ Example: $A = \{a, b\}, B = \{b, c, d\}, A - B = \{a\}$

Set Operations

The complement of a set A contains exactly those elements under consideration that are not in A:

CMSC 203 - Discrete Structures

 $A^c = U - A$

Example: U = N, B = {250, 251, 252, ...} B^c = {0, 1, 2, ..., 248, 249}

Fall 2002

Set Operations

Table 1 in Section How can we prov	n 1.5 shows many useful equat e A∪(B∩C) = (A∪B)∩(A∪C)?	tions
Method I: $x \in A \cup (B \cap C)$ $\Leftrightarrow x \in A \lor x \in (B \cap C)$ $\Leftrightarrow x \in A \lor (x \in B \land C)$ $\Leftrightarrow (x \in A \lor x \in B)$ (distributive $\Leftrightarrow x \in (A \cup B) \land x \in C$ $\Leftrightarrow x \in (A \cup B) \cap (A \cup C)$	C) $x \in C$) $(x \in A \lor x \in C)$ law for logical expressions) $(A \cup C)$ $\cup C$)	
Fall 2002	CMSC 203 - Discrete Structures	55

Set Operations

Method II: Membership table 1 means "x is an element of this set" 0 means "x is not an element of this set"

А	В	С	B∩C	A∪(B∩C)	A∪B	A∪C	$(A {\cup} B) \cap (A {\cup} C)$
0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0
0	1	0	0	0	1	0	0
0	1	1	1	1	1	1	1
1	0	0	0	1	1	1	1
1	0	1	0	1	1	1	1
1	1	0	0	1	1	1	1
1	1	1	1	1	1	1	1
Fall 2002 CMSC 203 - Discrete Structures							

Set Operations

Every logical expression can be transformed into an equivalent expression in set theory and vice versa.

You could work on Exercises 9 and 19 in Section 1.5 to get some practice.

CMSC 203 - Discrete Structures

Fall 2002



FunctionsA function f from a set A to a set B is an
assignment of exactly one element of B to each
element of A.We writef(a) = bif b is the unique element of B assigned by the
function f to the element a of A.If f is a function from A to B, we write
f: A \rightarrow B
(note: Here, " \rightarrow " has nothing to do with if... then)

Functions

If f:A \rightarrow B, we say that A is the domain of f and B is the codomain of f.

If f(a) = b, we say that b is the image of a and a is the pre-image of b.

The range of f:A \rightarrow B is the set of all images of elements of A.

CMSC 203 - Discrete Structures

We say that $f:A \rightarrow B$ maps A to B.

Fall 2002

Functions

Let us take a look at the function f:P→C with P = {Linda, Max, Kathy, Peter} C = {Boston, New York, Hong Kong, Moscow}

f(Linda) = Moscow f(Max) = Boston f(Kathy) = Hong Kong f(Peter) = New York

Here, the range of f is C.

Fall 2002

 Functions

 Let us re-specify f as follows:

 f(Linda) = Moscow

 f(Max) = Boston

 f(Kathy) = Hong Kong

 f(Peter) = Boston

 Is f still a function? yes

 What is its range? {Moscow, Boston, Hong Kong}

CMSC 203 - Discrete Structures





	Functions				
If the domain of our function f is large, it is convenient to specify f with a formula, e.g.:					
f: R→R f(x) = 2x					
This leads to: f(1) = 2 f(3) = 6 f(-3) = -6					
Fall 2002	CMSC 203 - Discrete Structures	64			

Functions

Let f_1 and f_2 be functions from A to **R**. Then the sum and the product of f_1 and f_2 are also functions from A to **R** defined by: $(f_1 + f_2)(x) = f_1(x) + f_2(x)$ $(f_1f_2)(x) = f_1(x) f_2(x)$ Example: $f_1(x) = 3x, f_2(x) = x + 5$ $(f_1 + f_2)(x) = f_1(x) + f_2(x) = 3x + x + 5 = 4x + 5$ $(f_1f_2)(x) = f_1(x) f_2(x) = 3x (x + 5) = 3x^2 + 15x$

Fall 2002

Functions

CMSC 203 - Discrete Structures

We already know that the range of a function $f:A \rightarrow B$ is the set of all images of elements $a \in A$.

If we only regard a subset $S \subseteq A$, the set of all images of elements $s \in S$ is called the image of S.

CMSC 203 - Discrete Structures

We denote the image of S by f(S):

 $f(S) = \{f(s) \mid s \in S\}$

Fall 2002

66

Functions

Let us look at the following well-known function: f(Linda) = Moscow f(Max) = Boston f(Kathy) = Hong Kong f(Peter) = Boston What is the image of $S = \{Linda, Max\}$? f(S) = {Moscow, Boston} What is the image of S = {Max, Peter} ? $f(S) = \{Boston\}$ Fall 2002

CMSC 203 - Discrete Structures

67

Properties of Functions A function $f:A \rightarrow B$ is said to be one-to-one (or injective), if and only if $\forall x, y \in A \ (f(x) = f(y) \rightarrow x = y)$ In other words: f is one-to-one if and only if it does not map two distinct elements of A onto the same element of B. Fall 2002 CMSC 203 - Discrete Structures 68

Properties of Functions						
And again						
f(Linda) = Moscow	1	g(Linda) = Moscow				
f(Max) = Boston		g(Max) = Boston				
f(Kathy) = Hong K	ong	g(Kathy) = Hong Kong	g			
f(Peter) = Boston		g(Peter) = New York				
lsfone-to-one?		Isgone-to-one?				
No, Max and Peter mapped onto the s element of the ima	r are same age.	Yes, each element is assigned a unique element of the image	e.			
Fall 2002	CMSC 203 - Disc	rete Structures	69			



Properties of Functions

How can we prove that a function f is one-to-one? Whenever you want to prove something, first take a look at the relevant definition(s): $\forall x, y \in A (f(x) = f(y) \rightarrow x = y)$ Example:

f:**R→R**

Fall 2002

 $f(x) = x^2$

Disproof by counterexample:

f(3) = f(-3), but $3 \neq -3$, so f is not one-to-one.

CMSC 203 - Discrete Structures

70

Properties of Functions... and yet another example: $f: \mathbf{R} \rightarrow \mathbf{R}$ f(x) = 3xOne-to-one: $\forall x, y \in A$ $(f(x) = f(y) \rightarrow x = y)$ To show: $f(x) \neq f(y)$ whenever $x \neq y$ $x \neq y$ $\Leftrightarrow 3x \neq 3y$ $\Leftrightarrow f(x) \neq f(y)$,so if $x \neq y$, then $f(x) \neq f(y)$, that is, f is one-to-one.Full 2002CMSC 203- Discrete Structures71

Properties of Functions

A function f:A \rightarrow B with A,B \subseteq R is called strictly increasing, if $\forall x, y \in A \ (x < y \rightarrow f(x) < f(y))$, and strictly decreasing, if $\forall x, y \in A \ (x < y \rightarrow f(x) > f(y))$.

Obviously, a function that is either strictly increasing or strictly decreasing is one-to-one.

CMSC 203 - Discrete Structures

Fall 2002

Properties of Functions

A function $f:A \rightarrow B$ is called onto, or surjective, if and only if for every element $b \in B$ there is an element $a \in A$ with f(a) = b.

In other words, ${\rm f}$ is onto if and only if its range is its entire codomain.

A function f: $A \rightarrow B$ is a one-to-one correspondence, or a bijection, if and only if it is both one-to-one and onto.

Obviously, if f is a bijection and A and B are finite sets, then |A| = |B|.

CMSC 203 - Discrete Structures

73

Fall 2002

Properties of FunctionsExamples:In the following examples, we use the arrow
representation to illustrate functions f:A→B.In each example, the complete sets A and B are
shown.























Inver	Inversion						
Example:	The inverse function f ⁻¹ is given by:						
f(Linda) = Moscow f(Max) = Boston f(Kathy) = Hong Kong f(Peter) = Lübeck f(Helena) = New York	f ⁻¹ (Moscow) = Linda f ⁻¹ (Boston) = Max f ⁻¹ (Hong Kong) = Kathy f ⁻¹ (Lübeck) = Peter f ⁻¹ (New York) = Helena						
Clearly, f is bijective.	I nversion is only possible for bijections (= invertible functions) crete Structures 81						











Composition

Composition of a function and its inverse:

 $(f^{-1\circ}f)(x) = f^{-1}(f(x)) = x$

Fall 2002

The composition of a function and its inverse is the **identity function** i(x) = x.

CMSC 203 - Discrete Structures

85

GraphsThe graph of a function f:A→B is the set of
ordered pairs {(a, b) | a∈A and f(a) = b}.The graph is a subset of A×B that can be used
to visualize f in a two-dimensional coordinate
system.

Floor and Ceiling Functions

The **floor** and **ceiling** functions map the real numbers onto the integers $(\mathbf{R} \rightarrow \mathbf{Z})$.

The **floor** function assigns to $r \in \mathbf{R}$ the largest $z \in \mathbf{Z}$ with $z \le r$, denoted by $\lfloor r \rfloor$.

Examples: $\lfloor 2.3 \rfloor = 2, \lfloor 2 \rfloor = 2, \lfloor 0.5 \rfloor = 0, \lfloor -3.5 \rfloor = -4$

The **ceiling** function assigns to $r \in \mathbf{R}$ the smallest $z \in \mathbf{Z}$ with $z \ge r$, denoted by $\lceil r \rceil$.

Examples: [2.3] = 3, [2] = 2, [0.5] = 1, [-3.5] = -3

CMSC 203 - Discrete Structures





Sequences							
Sequences represent ordered lists of elements.							
A sequence is defined as a function from a subset of N to a set S. We use the notation a_n to denote the image of the integer n. We call a_n a term of the sequence.							
Example:							
subset of N:	1	2	3	4	5		
	ļ	ļ	ļ	ļ	ļ		
S:	2	4	6	8	10		
Fall 2002	CMSC 203 - Discrete Structures 90					90	



Sequences

We use the notation $\{a_n\}$ to describe a sequence.

I mportant: Do not confuse this with the {} used in set notation.

It is convenient to describe a sequence with a **formula**.

For example, the sequence on the previous slide can be specified as $\{a_n\}$, where $a_n = 2n$.

CMSC 203 - Discrete Structures

91

92

Fall 2002

The Formula Game

What are the formulas that describe the following sequences $a_{11}, a_{22}, a_{33}, ...$?

1, 3, 5, 7, 9, ... $a_n = 2n - 1$ -1, 1, -1, 1, -1, ... $a_n = (-1)^n$ 2, 5, 10, 17, 26, ... $a_n = n^2 + 1$ 0.25, 0.5, 0.75, 1, 1.25 ... $a_n = 0.25n$ 3, 9, 27, 81, 243, ... $a_n = 3^n$ Fall 2002CMSC 203 - Discrete Structures

Strings

Finite sequences are also called $\mbox{strings},$ denoted by $a_1a_2a_3...a_n.$

The $\ensuremath{\textit{length}}$ of a string S is the number of terms that it consists of.

The **empty string** contains no terms at all. It has length zero.

CMSC 203 - Discrete Structures



Summations How can we express the sum of the first 1000 terms of the sequence $\{a_n\}$ with $a_n=n^2$ for n = 1, 2, 3, ?					
We write it as $\sum_{j=1}^{\infty} j^2$.					
What is the value of $\sum_{j=1}^{6} j$?					
It is 1 + 2 + 3 + 4 + 5 + 6 = 21.					
What is the value of $\sum_{j=1}^{100} j$?					
It is so much work to calculate this					
Fall 2002 CMSC 203 - Discrete Structures	95				



summation can be calculated much more easily, for example:

96

 $=\frac{100(100+1)}{2}=\frac{10100}{2}=5050$ $\sum_{j=1}^{100}$ Fall 2002 CMSC 203 - Discrete Structures

 $\sum_{j=1}^{n}$





















Algorithms

What is an algorithm?

Fall 2002

An algorithm is a finite set of precise instructions for performing a computation or for solving a problem.

This is a rather vague definition. You will get to know a more precise and mathematically useful definition when you attend CS420.

CMSC 203 - Discrete Structures

But this one is good enough for now...

103

104

Algorithms Properties of algorithms: • Input from a specified set, • Output from a specified set (solution), • Definiteness of every step in the computation, • Correctness of output for every possible input,

+ $\ensuremath{\textit{Finiteness}}$ of the number of calculation steps,

CMSC 203 - Discrete Structures

- Effectiveness of each calculation step and
- Generality for a class of problems.

Fall 2002

Algorithm Examples

We will use a pseudocode to specify algorithms, which slightly reminds us of Basic and Pascal. Example: an algorithm that finds the maximum element in a finite sequence

CMSC 203 - Discrete Structures

 $\begin{array}{l} \textbf{procedure } \max(a_1, a_2, ..., a_n: \text{ integers}) \\ \max := a_1 \\ \textbf{for } i := 2 \textbf{ to } n \\ \quad \textbf{if } \max < a_i \textbf{ then } \max := a_i \\ \{ \text{max is the largest element} \} \end{array}$

Fall 2002

Algorithm Examples						
Another example: a linear search algorithm, that is, an algorithm that linearly searches a sequence for a particular element.						
<pre>procedure linear_search(x: integer; a1, a2,, an: integers)</pre>						
i := 1 while (i ≤ n and x ≠ a _i) i := i + 1 if i ≤ n then location := i						
else location := 0 {location is the subscript of the term that equals x, or is zero if x is not found}						
Fall 2002 CMSC 203 - Discrete Structures 106						






















Algorithm Examples procedure binary_search(x: integer; a₁, a₂, ..., a_n: integers) i := 1 {i is left endpoint of search interval} j := n {j is right endpoint of search interval} while (i < j) begin $m := \lfloor (i + j)/2 \rfloor$ if x > a_m then i := m + 1 else j := m end if x = a_i then location := i else location := 0 {location is the subscript of the term that equals x, or is zero if x is not found} Full 202 CMSC 203-Discrete Structures 113

Complexity
In general, we are not so much interested in the time and space complexity for small inputs.
For example, while the difference in time complexity between linear and binary search is meaningless for a sequence with $n = 10$, it is gigantic for $n = 2^{30}$.

CMSC 203 - Discrete Structures

Complexity

For example, let us assume two algorithms A and B that solve the same class of problems.

The time complexity of A is 5,000n, the one for B is $[1.1^n]$ for an input with n elements.

For n = 10, A requires 50,000 steps, but B only 3, so B seems to be superior to A.

CMSC 203 - Discrete Structures

For n = 1000, however, A requires 5,000,000 steps, while B requires $2.5 \cdot 10^{41}$ steps.

Fall 2002



Complexity			
Comparison: time complexity of algorithms A and B			
	Input Size	Algorithm A	Algorithm B
	n	5,000n	[1.1 ⁿ]
	10	50,000	3
	100	500,000	13,781
	1,000	5,000,000	2.5·10 ⁴¹
	1,000,000	5·10 ⁹	4.8·10 ⁴¹³⁹²
	Fall 2002	CMSC 203 - Discrete Structures	117



Complexity

This means that algorithm B cannot be used for large inputs, while running algorithm A is still feasible.

So what is important is the **growth** of the complexity functions.

The growth of time and space complexity with increasing input size n is a suitable measure for the comparison of algorithms.

Fall 2002

CMSC 203 - Discrete Structures

118

The Growth of Functions The growth of functions is usually described using the big-O notation .
The growth of functions is usually described using the big-O notation .
Definition: Let f and g be functions from the integers or the real numbers to the real numbers. We say that $f(x)$ is $O(g(x))$ if there are constants C and k such that
$ f(x) \le C g(x) $
whenever x > k.
Fall 2002 CMSC 203 - Discrete Structures 119

The Growth of Functions

When we analyze the growth of **complexity functions**, f(x) and g(x) are always positive.

Therefore, we can simplify the big-O requirement to

 $f(x) \leq C \cdot g(x)$ whenever x > k.

If we want to show that f(x) is O(g(x)), we only need to find **one** pair (C, k) (which is never unique).

CMSC 203 - Discrete Structures

The Growth of Functions

The idea behind the big-O notation is to establish an **upper boundary** for the growth of a function f(x) for large x.

This boundary is specified by a function g(x) that is usually much **simpler** than f(x).

We accept the constant C in the requirement

 $f(x) \le C \cdot g(x)$ whenever x > k,

because C does not grow with $\boldsymbol{x}.$

We are only interested in large x, so it is OK if $f(x) > C \cdot g(x)$ for $x \le k$.

CMSC 203 - Discrete Structures

Fall 2002

121



The Growth of Functions

Question: If f(x) is $O(x^2)$, is it also $O(x^3)$?

Yes. x^3 grows faster than x^2 , so x^3 grows also faster than f(x).

Therefore, we always have to find the **smallest** simple function g(x) for which f(x) is O(g(x)).

CMSC 203 - Discrete Structures

The Growth of Functions "Popular" functions g(n) are n log n, 1, 2 ⁿ , n ² , n!, n, n ³ , log n	i
Listed from slowest to fastest growth: • 1 • log n • n • n log n • n ² • n ³ • 2 ⁿ • n!	
Fall 2002 CMSC 203 - Discrete Structures	124





Useful Rules for Big-O

For any **polynomial** $f(x) = a_n x^n + a_{n-1} x^{n-1} + ... + a_0$, where $a_0, a_1, ..., a_n$ are real numbers, f(x) is $O(x^n)$.

If $f_1(x)$ is $O(g_1(x))$ and $f_2(x)$ is $O(g_2(x))$, then $(f_1 + f_2)(x)$ is $O(max(g_1(x), g_2(x)))$

If $f_1(x)$ is O(g(x)) and $f_2(x)$ is O(g(x)), then $(f_1 + f_2)(x)$ is O(g(x)).

If $f_1(x)$ is $O(g_1(x))$ and $f_2(x)$ is $O(g_2(x))$, then $(f_1f_2)(x)$ is $O(g_1(x) g_2(x))$.

Fall 2002 CMSC 203 - Discrete Structures





Let us get into... Number Theory

CMSC 203 - Discrete Structures







Primes

A positive integer p greater than 1 is called prime if the only positive factors of p are 1 and p.

A positive integer that is greater than 1 and is not prime is called composite.

The fundamental theorem of arithmetic:

Fall 2002

Every positive integer can be written **uniquely** as the **product of primes**, where the prime factors are written in order of increasing size.

CMSC 203 - Discrete Structures

133

Primes			
Examples	S:		
15 =	3.5		
48 =	$2 \cdot 2 \cdot 2 \cdot 2 \cdot 3 = 2^4 \cdot 3$		
17 =	17		
100 =	$2 \cdot 2 \cdot 5 \cdot 5 = 2^2 \cdot 5^2$		
512 =	$2 \cdot 2 = 2^9$		
515 =	5.103		
28 =	2.2.7		
Fall 2002	CMSC 203 - Discrete Structures	134	

Primes

If n is a composite integer, then n has a prime divisor less than or equal \sqrt{n} .

This is easy to see: if n is a composite integer, it must have two prime divisors p_1 and p_2 such that $p_1 \cdot p_2 = n$.

CMSC 203 - Discrete Structures

 $\mathbf{p_1}$ and $\mathbf{p_2}$ cannot both be greater than \sqrt{n} , because then $p_1 \cdot p_2 > n.$

The Division Algorithm

CMSC 203 - Discrete Structures

Let **a** be an integer and **d** a positive integer. Then there are unique integers **q** and **r**, with $0 \le r < d$, such that a = dq + r.

In the above equation,

- d is called the divisor,
- a is called the dividend,
- q is called the quotient, and
 r is called the remainder.
- I is called the remainder

Fall 2002





Greatest Common Divisors Let a and b be integers, not both zero. The largest integer d such that d | a and d | b is called the **greatest common divisor** of a and b. The greatest common divisor of a and b is denoted by gcd(a, b). **Example 1:** What is gcd(48, 72) ? The positive common divisors of 48 and 72 are

1, 2, 3, 4, 6, 8, 12, 16, and 24, so gcd(48, 72) = 24. **Example 2:** What is gcd(19, 72) ? The only positive common divisor of 19 and 72 is 1, so gcd(19, 72) = 1.

CMSC 203 - Discrete Structures

139



Relatively Prime Integers

Definition:

Fall 2002

Two integers a and b are **relatively prime** if gcd(a, b) = 1.

Examples:

Are 15 and 28 relatively prime? Yes, gcd(15, 28) = 1. Are 55 and 28 relatively prime? Yes, gcd(55, 28) = 1. Are 35 and 28 relatively prime? No, gcd(35, 28) = 7.

Relatively Prime Integers

Definition:

The integers $a_1, a_2, ..., a_n$ are **pairwise relatively prime** if $gcd(a_i, a_i) = 1$ whenever $1 \le i < j \le n$.

Examples:

Fall 2002

Are 15, 17, and 27 pairwise relatively prime? No, because gcd(15, 27) = 3.

Are 15, 17, and 28 pairwise relatively prime? Yes, because gcd(15, 17) = 1, gcd(15, 28) = 1 and gcd(17, 28) = 1.

CMSC 203 - Discrete Structures

142



143

Least Common Multiples

Using prime factorizations:

 $a = p_1{}^{a_1} \ p_2{}^{a_2} \dots \ p_n{}^{a_n} \,, \ b = p_1{}^{b_1} \ p_2{}^{b_2} \dots \ p_n{}^{b_n} \,,$ where $p_1 < p_2 < ... < p_n$ and $a_{i'}$ $b_i \in ~\boldsymbol{N}$ for $1 \leq i \leq n$

Icm(a, b) = $p_1^{\max(a_1, b_1)} p_2^{\max(a_2, b_2)} \dots p_n^{\max(a_n, b_n)}$

Example:

 $a = 60 = 2^2 3^1 5^1$

 $b = 54 = 2^1 3^3 5^0$

 $Icm(a, b) = 2^2 3^3 5^1 = 4 \cdot 27 \cdot 5 = 540$ Fall 2002 CMSC 203 - Discrete Structures







		_
Cong	ruences	
Let a and b be integers We say that a is congr m divides a - b.	and m be a positive integer uent to b modulo m if	
We use the notation $\mathbf{a} = \mathbf{b} \pmod{\mathbf{m}}$ to indicate that a is congruent to b modulo m.		
In other words: $a \equiv b \pmod{m}$ if and only if a mod m = b mod m .		
Fall 2002 CMSC 202	- Discrete Structures 147	

Congruences

Examples:

Is it true that $46 \equiv 68 \pmod{11}$? Yes, because 11 | (46 - 68). Is it true that $46 \equiv 68 \pmod{22}$? Yes, because 22 | (46 - 68). For which integers z is it true that $z \equiv 12 \pmod{10}$? It is true for any $z \in \{\dots, -28, -18, -8, 2, 12, 22, 32, \dots\}$

Theorem: Let m be a positive integer. The integers a and b are congruent modulo m if and only if there is an integer k such that a = b + km.

CMSC 203 - Discrete Structures

148

Fall 2002

CongruencesTheorem: Let m be a positive integer.If $a \equiv b \pmod{m}$ and $c \equiv d \pmod{m}$, then $a + c \equiv b + d \pmod{m}$ and $ac \equiv bd \pmod{m}$.Proof:We know that $a \equiv b \pmod{m}$ and $c \equiv d \pmod{m}$.Therefore: $b + d \pmod{m} + (c + tm) = (a + c) + m(s + t)$ andbd = (a + sm) + (c + tm) = (a + c) + m(s + t) andbd = (a + sm)(c + tm) = ac + m(at + cs + stm).Hence, $a + c \equiv b + d \pmod{m}$ and $ac \equiv bd \pmod{m}$.

The Euclidean Algorithm

The Euclidean Algorithm finds the greatest common divisor of two integers a and b. For example, if we want to find gcd(287, 91), we divide 287 by 91: $287 = 91 \cdot 3 + 14$ We know that for integers a, b and c, if a | b and a | c, then a | (b + c). Therefore, any divisor of 287 and 91 must also be a divisor of 287 - 91 \cdot 3 = 14.

CMSC 203 - Discrete Structures

Consequently, gcd(287, 91) = gcd(14, 91).

Fall 2002

The Euclidean AlgorithmIn the next step, we divide 91 by 14: $91 = 14 \cdot 6 + 7$ $91 = 14 \cdot 6 + 7$ This means that gcd(14, 91) = gcd(14, 7).So we divide 14 by 7: $14 = 7 \cdot 2 + 0$ We find that 7 | 14, and thus gcd(14, 7) = 7.Therefore, gcd(287, 91) = 7.





Representations of Integers		
Let b be a positive integer greater than 1. Then if n is a positive integer, it can be expressed uniquely in the form:		
$n = a_k b^k + a_{k-1} b^{k-1} + \dots + a_1 b + a_0,$		
where k is a nonnegative integer, $a_0, a_1,, a_k$ are nonnegative integers less than b, and $a_k \neq 0$.		
Example for b=10: $859 = 8 \cdot 10^2 + 5 \cdot 10^1 + 9 \cdot 10^0$		
Fall 2002 CMSC 203 - Discrete Structures 153		

Representations of Integers

Example for b=2 (binary expansion): $(10110)_2 = 1.2^4 + 1.2^2 + 1.2^1 = (22)_{10}$

Example for b=16 (hexadecimal expansion): (we use letters A to F to indicate numbers 10 to 15) $(3AOF)_{16} = 3.16^3 + 10.16^2 + 15.16^0 = (14863)_{10}$

CMSC 203 - Discrete Structures

Fall 2002

154



Representations of Integers

Example: What is the base 8 expansion of $(12345)_{10}$?

First, divide 12345 by 8: 12345 = 8·1543 + 1

1543 = 8.192 + 7 192 = 8.24 + 0 24 = 8.3 + 03 = 8.0 + 3

Fall 2002

The result is: $(12345)_{10} = (30071)_8$.

CMSC 203 - Discrete Structures

Representations of Integers procedure base_b_expansion(n, b: positive integers) q := n k := 0 while $q \neq 0$ begin $a_k := q \mod b$ $q := \lfloor q/b \rfloor$ k := k + 1 end {the base b expansion of n is $(a_{k-1} \dots a_1 a_0)_b$ } Fall 2002 CMSC 203 - Discrete Structures 157



Addition of Integers

CMSC 203 - Discrete Structures

Continue this process until you obtain c_{n-1} .

The leading bit of the sum is $s_n = c_{n-1}$.

The result is:

 $a + b = (s_n s_{n-1} \dots s_1 s_0)_2$

 Addition of Integers

 Example:

 Add $a = (1110)_2$ and $b = (1011)_2$.

 $a_0 + b_0 = 0 + 1 = 0.2 + 1$, so that $c_0 = 0$ and $s_0 = 1$.

 $a_1 + b_1 + c_0 = 1 + 1 + 0 = 1.2 + 0$, so $c_1 = 1$ and $s_1 = 0$.

 $a_2 + b_2 + c_1 = 1 + 0 + 1 = 1.2 + 0$, so $c_2 = 1$ and $s_2 = 0$.

 $a_3 + b_3 + c_2 = 1 + 1 + 1 = 1.2 + 1$, so $c_3 = 1$ and $s_3 = 1$.

 $s_4 = c_3 = 1$.

 Therefore, $s = a + b = (11001)_2$.







Addition of Integers

Let $a = (a_{n-1}a_{n-2}...a_1a_0)_2$, $b = (b_{n-1}b_{n-2}...b_1b_0)_2$. How can we **algorithmically** add these two binary numbers? First, add their rightmost bits: $a_0 + b_0 = c_0.2 + s_0$, where s_0 is the **rightmost bit** in the binary expansion of a + b, and c_0 is the **carry**. Then, add the next pair of bits and the carry:

$$a_1 + b_1 + c_0 = c_1 \cdot 2 + s_1,$$
where s_1 is the **next bit** in the binary expansion of
 a + b, and c_1 is the carry.
 Fall 2002 CMSC 203- Discrete Structures 162
 162









Mathematical Reasoning We need **mathematical reasoning** to • determine whether a mathematical argument is

correct or incorrect andconstruct mathematical arguments.

Mathematical reasoning is not only important for conducting **proofs** and **program verification**, but also for **artificial intelligence** systems (drawing inferences).

CMSC 203 - Discrete Structures

167

168

Fall 2002

Fall 2002

Terminology

An $\ensuremath{\textit{axiom}}$ is a basic assumption about mathematical structured that needs no proof.

We can use a **proof** to demonstrate that a particular statement is true. A proof consists of a sequence of statements that form an argument.

The steps that connect the statements in such a sequence are the **rules of inference**.

Cases of incorrect reasoning are called **fallacies**.

A **theorem** is a statement that can be shown to be true.

CMSC 203 - Discrete Structures

Terminology

A **lemma** is a simple theorem used as an intermediate result in the proof of another theorem.

Fall 2002

A **corollary** is a proposition that follows directly from a theorem that has been proved.

A **conjecture** is a statement whose truth value is unknown. Once it is proven, it becomes a theorem.

CMSC 203 - Discrete Structures

169

Rules of InferenceRules of inference provide the justification of
the steps used in a proof.One important rule is called modus ponens or the
law of detachment. It is based on the tautology
 $(p \land (p \rightarrow q)) \rightarrow q$. We write it in the following way:p
 $p \rightarrow q$
 $\overrightarrow{\cdot q}$ The two hypotheses p and $p \rightarrow q$ are
written in a column, and the conclusion
below a bar, where \therefore means "therefore".

Rules of Inference			
The general form of a rule of inference is:			
p ₁ p ₂	The rule states that if p_1 and p_2 and and p_n are all true, then q is true as we	II.	
p _n ∴ q	These rules of inference can be used in any mathematical argument and do not require any proof.	1	
Fall 2002	CMSC 203 - Discrete Structures 17	1	





ArgumentsJust like a rule of inference, an argument consists
of one or more hypotheses and a conclusion.We say that an argument is valid, if whenever all
its hypotheses are true, its conclusion is also true.However, if any hypothesis is false, even a valid
argument can lead to an incorrect conclusion.

Arguments

Example:

"If 101 is divisible by 3, then 101^2 is divisible by 9. 101 is divisible by 3. Consequently, 101^2 is divisible by 9."

Although the argument is **valid**, its conclusion is **incorrect**, because one of the hypotheses is false ("101 is divisible by 3.").

I f in the above argument we replace 101 with 102, we could correctly conclude that 102^2 is divisible by 9.

CMSC 203 - Discrete Structures

Fall 2002

174

Arguments Which rule of inference was used in the last argument?	
p: "101 is divisible by 3." q: "101 ² is divisible by 9."	
$\frac{p}{p \rightarrow q} \qquad Modus \\ \frac{p}{\therefore q} \qquad ponens$	
Unfortunately, one of the hypotheses (p) is fal Therefore, the conclusion q is incorrect.	se.
Fall 2002 CMSC 203 - Discrete Structures 17	15

	Arguments	
Another exa	mple:	
"If it rains to barbeque too today, then v Therefore, if barbeque ton	oday, then we will not have a lay. If we do not have a barb ve will have a barbeque tomo f it rains today, then we will norrow."	oeque rrow. have a
This is a valic true, then its	d argument: If its hypotheses s conclusion is also true.	es are
Fall 2002	CMSC 203 - Discrete Structures	176

	Arguments	
Let us formalize the previous argument: p: "It is raining today." g: "We will not have a barbeque today."		
r: "We will have a barbecue tomorrow."		
So the arg	ument is of the following form:	
$p \rightarrow q$ $q \rightarrow r$ $\overrightarrow{:} p \rightarrow r$	Hypothetical syllogism	
Fall 2002	CMSC 203 - Discrete Structures	177

Arguments

Another example:

Gary is either intelligent or a good actor. If Gary is intelligent, then he can count from 1 to 10. Gary can only count from 1 to 2. Therefore, Gary is a good actor.

i: "Gary is intelligent."

Fall 2002

a: "Gary is a good actor."

c: "Gary can count from 1 to 10."

178

Arguments

CMSC 203 - Discrete Structures

i: "Gary is intelligent." a: "Gary is a good actor." c: "Gary can count from 1 to 10."			
Step 1: $\neg c$ Step 2: $i \rightarrow c$ Step 3: $\neg i$ Step 4: $a \lor i$ Step 5: a	Hypothesis Hypothesis Modus tollens Steps 1 & 2 Hypothesis Disjunctive Syllogism Steps 3 & 4		
Conclusion: a ("Gary is a good actor.")			
Fall 2002 Cl	ASC 203 - Discrete Structures 179		

Arguments

Yet another example:

I f you listen to me, you will pass CS 320. You passed CS 320. Therefore, you have listened to me.

Is this argument valid?

No, it assumes $((p{\rightarrow} q) \wedge q) \rightarrow p.$ This statement is not a tautology. It is false if p is false and q is true.

CMSC 203 - Discrete Structures

Fall 2002

Rules of Inference for Quantified Statements		
∀x P(x)	Universal	
∴ P(c) if c∈ U	instantiation	
P(c) for an arbitrary	itrary c∈U Universal	
∴ ∀x P(x)	generalization	
∃x P(x)	Existential	
∴ P(c) for some eler	ment $c \in U$ instantiation	
P(c) for some eleme	nt c∈U Existential	
∴ ∃x P(x)	generalization	
Fall 2002 CMSC 203	- Discrete Structures 181	



Rules of Inference for Quantified Statements		
Example:		
Every UMB st George is a U Therefore, G	tudent is a genius. MB student. eorge is a genius.	
U(x): "x is a UMB student." G(x): "x is a genius."		
Fall 2002	CMSC 203 - Discrete Structures	182

Rules of Inference for Quantified Statements			
The following steps are used in the argument:			
Step 1: $\forall x (U(x) \rightarrow G(x))$ Step 2: U(George) \rightarrow G(George) Step 3: U(George) Step 4: G(George)		Hypothesis Univ. instantiation using Step 1 Hypothesis Modus ponens	
	$\frac{\forall x P(x)}{\therefore P(c) \text{ if } c \in U}$	Univer instan	rsal tiation
Fall 2002	CMSC 203 - Discrete Structures 183		



Proving Theorems

Direct proof:

Fall 2002

An implication $p \rightarrow q$ can be proved by showing that if p is true, then q is also true.

Example: Give a direct proof of the theorem "If n is odd, then n^2 is odd."

I dea: Assume that the hypothesis of this implication is true (n is odd). Then use rules of inference and known theorems to show that q must also be true (n^2 is odd).

CMSC 203 - Discrete Structures

184

Proving Theorems n is odd. Then n = 2k + 1, where k is an integer. Consequently, $n^2 = (2k + 1)^2$. $= 4k^2 + 4k + 1$ $= 2(2k^2 + 2k) + 1$ Since n² can be written in this form, it is odd.

Proving Theorems

Indirect proof:

Fall 2002

An implication $p \rightarrow q$ is equivalent to its **contrapositive** $\neg q \rightarrow \neg p$. Therefore, we can prove $p \rightarrow q$ by showing that whenever q is false, then p is also false.

Example: Give an indirect proof of the theorem "If 3n + 2 is odd, then n is odd."

I dea: Assume that the conclusion of this implication is false (n is even). Then use rules of inference and known theorems to show that p must also be false (3n + 2 is even).

CMSC 203 - Discrete Structures

Proving Theoremsn is even.Then n = 2k, where k is an integer.I t follows that
$$3n + 2 = 3(2k) + 2$$
 $= 6k + 2$ $= 2(3k + 1)$ Therefore, $3n + 2$ is even.We have shown that the contrapositive of the
implication is true, so the implication itself is also
true (I f $2n + 3$ is odd, then n is odd).Full 2002CMSC 203 - Discrete StructuresFull 2002CMSC 203 - Discrete Structures





Induction

I f we have a propositional function P(n), and we want to prove that P(n) is true for any natural number n, we do the following:

 Show that P(0) is true. (basis step)

Fall 2002

• Show that if P(n) then P(n + 1) for any $n \in \mathbb{N}$. (inductive step)

CMSC 203 - Discrete Structures

190

191

 Then P(n) must be true for any n∈ N. (conclusion)

InductionExample:Show that n < 2ⁿ for all positive integers n.Let P(n) be the proposition "n < 2ⁿ."1. Show that P(1) is true.
(basis step)P(1) is true, because 1 < 2¹ = 2.

Induction

 Show that if P(n) is true, then P(n + 1) is true. (inductive step)

Assume that $n < 2^n$ is true. We need to show that P(n + 1) is true, i.e. $n+1 < 2^{n+1}$

CMSC 203 - Discrete Structures

We start from n < 2ⁿ: $n + 1 < 2^n + 1 \le 2^n + 2^n = 2^{n+1}$ Therefore, if n < 2ⁿ then n + 1 < 2ⁿ⁺¹

Fall 2002











Induction

The second principle of mathematical induction:

- Show that P(0) is true. (basis step)
- Show that if P(0) and P(1) and ... and P(n), then P(n + 1) for any n∈ N. (inductive step)

CMSC 203 - Discrete Structures

• Then P(n) must be true for any n∈ N. (conclusion)





Induction

 Then P(n) must be true for any n∈ N. (conclusion)

End of proof.

We have shown that **every integer greater than 1** can be written as the product of primes.

CMSC 203 - Discrete Structures





Recursively Defined Sequences Example:

The sequence $\{a_n\}$ of powers of 2 is given by a_n = 2^n for n = 0, 1, 2, \dots .

The same sequence can also be defined **recursively**:

 $a_0 = 1$ $a_{n+1} = 2a_n$ for n = 0, 1, 2, ...

Obviously, induction and recursion are similar principles.

```
CMSC 203 - Discrete Structures
```

Fall 2002





Recursively Defined Functions		
How can we recursively define the factorial function $f(n) = n!$?		
f(0) = 1 f(n + 1) = (n + 1)f(n + 1)	-(n)	
f(0) = 1 $f(1) = 1f(0) = 1 \cdot 1 = 1$ $f(2) = 2f(1) = 2 \cdot 1 = 2$ $f(3) = 3f(2) = 3 \cdot 2 = 6$ $f(4) = 4f(3) = 4 \cdot 6 = 24$		
Fall 2002	CMSC 203 - Discrete Structures	207

Recursively Defined Functions

A famous example: The Fibonacci numbers f(0) = 0, f(1) = 1 f(n) = f(n - 1) + f(n - 2) f(0) = 0 f(1) = 1 f(2) = f(1) + f(0) = 1 + 0 = 1 f(3) = f(2) + f(1) = 1 + 1 = 2 f(4) = f(3) + f(2) = 2 + 1 = 3 f(5) = f(4) + f(3) = 3 + 2 = 5 f(6) = f(5) + f(4) = 5 + 3 = 8Pall 2002 CMSC 203- Discrete Structures 2008



Recursively Defined Sets

Proof:

Fall 2002

Let A be the set of all positive integers divisible by 3.

To show that A = S, we must show that $A \subseteq S$ and $S \subseteq A$.

Part I: To prove that $A \subseteq S$, we must show that every positive integer divisible by 3 is in S.

We will use mathematical induction to show this.

CMSC 203 - Discrete Structures

Recursively Defined Sets

Let P(n) be the statement "3n belongs to S".

Basis step: P(1) is true, because 3 is in S.

Inductive step: To show: If P(n) is true, then P(n + 1) is true.

Assume 3n is in S. Since 3n is in S and 3 is in S, it follows from the recursive definition of S that 3n + 3 = 3(n + 1) is also in S.

Conclusion of Part I: $A \subseteq S$.

Fall 2002

CMSC 203 - Discrete Structures

Recursively Defined Sets

Part II: To show: $S \subseteq A$.

Basis step: To show: All initial elements of S are in A. 3 is in A. True.

Inductive step: To show: (x + y) is in A whenever x and y are in S.

If x and y are both in A, it follows that 3 | x and 3 | y. From Theorem I, Section 2.3, it follows that 3 | (x + y).

Conclusion of Part II: $S \subseteq A$. Overall conclusion: A = S. CMSC 203 - Discrete Structures

Fall 2002

212

211

Recursively Defined Sets

Another example:

The well-formed formulae of variables, numerals and operators from $\{+, -, *, /, ^\}$ are defined by:

x is a well-formed formula if x is a numeral or variable.

(f + g), (f - g), (f * g), (f / g), $(f ^ g)$ are wellformed formulae if f and g are.

CMSC 203 - Discrete Structures

Recursively Defined Sets

With this definition, we can construct formulae such as:

(x - y) ((z / 3) - y) ((z / 3) - (6 + 5)) ((z / (2 * 4)) - (6 + 5))

Fall 2002

Fall 2002

Recursive	Algorithms
NCCUI SIVC	rugoritinns

CMSC 203 - Discrete Structures

214

215

An algorithm is called **recursive** if it solves a problem by reducing it to an instance of the same problem with smaller input.

Example I: Recursive Euclidean Algorithm

procedure gcd(a, b: nonnegative integers with a < b)
if a = 0 then gcd(a, b) := b
else gcd(a, b) := gcd(b mod a, a)</pre>

CMSC 203 - Discrete Structures

Recursive Algorithms

CMSC 203 - Discrete Structures

Example II: Recursive Fibonacci Algorithm

procedure fibo(n: nonnegative integer) **if** n = 0 **then** fibo(0) := 0 **else if** n = 1 **then** fibo(1) := 1 **else** fibo(n) := fibo(n - 1) + fibo(n - 2)

Fall 2002




Recursive Algorithms procedure iterative_fibo(n: nonnegative integer)
if n = 0 then y := 0 else begin x := 0 y := 1 for i := 1 to n-1 begin z := x + y x : = y y := z end end {y is the n-th Fibonacci number} Fall 2002 CMSC 203 - Discrete Structures 218



CMSC 203 - Discrete Structures





Basic Counting Principles

The sum rule:

If a task can be done in n_1 ways and a second task in n_2 ways, and if these two tasks cannot be done at the same time, then there are $n_1 + n_2$ ways to do either task.

Example:

The department will award a free computer to either a CS student or a CS professor. How many different choices are there, if there are 530 students and 15 professors?

CMSC 203 - Discrete Structures

There are 530 + 15 = 545 choices.

Fall 2002

Basic Counting Principles

Generalized sum rule:

If we have tasks T_1 , T_2 , ..., T_m that can be done in n_1 , n_2 , ..., n_m ways, respectively, and no two of these tasks can be done at the same time, then there are $n_1 + n_2 + ... + n_m$ ways to do one of these tasks.

CMSC 203 - Discrete Structures

223

Fall 2002

Basic Counting Principles**The product rule:**Suppose that a procedure can be broken down
into two successive tasks. If there are n, ways to
do the first task and n2 ways to do the second
task after the first task has been done, then
there are n1n2 ways to do the procedure.

Basic Counting Principles

Example:

How many different license plates are there that containing exactly three English letters ?

Solution:

There are 26 possibilities to pick the first letter, then 26 possibilities for the second one, and 26 for the last one.

So there are $26 \cdot 26 \cdot 26 = 17576$ different license plates.

CMSC 203 - Discrete Structures

Basic Counting Principles

Generalized product rule:

If we have a procedure consisting of sequential tasks T₁, T₂, ..., T_m that can be done in n₁, n₂, ..., n_m ways, respectively, then there are $n_1 \cdot n_2 \cdot ... \cdot n_m$ ways to carry out the procedure.

Basic Counting Principles

CMSC 203 - Discrete Structures

The sum and product rules can also be phrased in terms of set theory.

Sum rule: Let $A_1, A_2, ..., A_m$ be disjoint sets. Then the number of ways to choose any element from one of these sets is $|A_1 \cup A_2 \cup ... \cup A_m| =$ $|A_1| + |A_2| + ... + |A_m|.$

Product rule: Let $A_1, A_2, ..., A_m$ be finite sets. Then the number of ways to choose one element from each set in the order $A_1, A_2, ..., A_m$ is $|\mathsf{A}_1 \times \mathsf{A}_2 \times ... \times \mathsf{A}_m| = |\mathsf{A}_1| \cdot |\mathsf{A}_2| \cdot ... \cdot |\mathsf{A}_m|.$

CMSC 203 - Discrete Structures

Fall 2002

Fall 2002

226

227

Inclusion-Exclusion

How many bit strings of length 8 either start with a 1 or end with 00?

Task 1: Construct a string of length 8 that starts with a 1.

There is one way to pick the first bit (1), two ways to pick the second bit (0 or 1), two ways to pick the third bit (0 or 1),

two ways to pick the eighth bit (0 or 1).

Product rule: Task 1 can be done in $1.2^7 = 128$ ways. CMSC 203 - Discrete Structures

Fall 2002





Inclusion-Exclusion

I f we want to use the sum rule in such a case, we have to subtract the cases when Tasks 1 and 2 are done at the same time.

How many cases are there, that is, how many strings start with 1 **and** end with 00?

There is one way to pick the first bit (1), two ways for the second, ..., sixth bit (0 or 1), one way for the seventh, eighth bit (0).

Product rule: In 2^5 = 32 cases, Tasks 1 and 2 are carried out at the same time.

CMSC 203 - Discrete Structures

Fall 2002

Inclusion-Exclusion

Since there are 128 ways to complete Task 1 and 64 ways to complete Task 2, and in 32 of these cases Tasks 1 and 2 are completed at the same time, there are

128 + 64 - 32 = 160 ways to do either task.

In set theory, this corresponds to sets A_1 and A_2 that are **not** disjoint. Then we have:

 $|\mathsf{A}_1\cup\mathsf{A}_2|=|\mathsf{A}_1|+|\mathsf{A}_2|-|\mathsf{A}_1\cap\mathsf{A}_2|$

Fall 2002

This is called the principle of inclusion-exclusion.

CMSC 203 - Discrete Structures

232





The Pigeonhole Principle

The pigeonhole principle: If (k + 1) or more objects are placed into k boxes, then there is at least one box containing two or more of the objects.

Example 1: If there are 11 players in a soccer team that wins 12-0, there must be at least one player in the team who scored at least twice.

Example 2: If you have 6 classes from Monday to Friday, there must be at least one day on which you have at least two classes.

CMSC 203 - Discrete Structures

The Pigeonhole Principle

The generalized pigeonhole principle: If N objects are placed into k boxes, then there is at least one box containing at least [N/k] of the objects.

Example 1: In our 60-student class, at least 12 students will get the same letter grade (A, B, C, D, or F).

CMSC 203 - Discrete Structures

235

The Pigeonhole Principle

Example 2: Assume you have a drawer containing a random distribution of a dozen brown socks and a dozen black socks. It is dark, so how many socks do you have to pick to be sure that among them there is a matching pair?

There are two types of socks, so if you pick at least 3 socks, there must be either at least two brown socks or at least two black socks.

CMSC 203 - Discrete Structures

Generalized pigeonhole principle: $\lceil 3/2 \rceil = 2$.

Fall 2002

Fall 2002

Fall 2002

236

Permutations and Combinations

How many ways are there to pick a set of 3 people from a group of 6?

There are 6 choices for the first person, 5 for the second one, and 4 for the third one, so there are $6.5 \cdot 4 = 120$ ways to do this.

This is not the correct result!

For example, picking person C, then person A, and then person E leads to the **same group** as first picking E, then C, and then A.

However, these cases are counted **separately** in the above equation.

CMSC 203 - Discrete Structures

Permutations and Combinations

So how can we compute how many different subsets of people can be picked (that is, we want to disregard the order of picking) ?

To find out about this, we need to look at **permutations**.

A **permutation** of a set of distinct objects is an ordered arrangement of these objects.

An ordered arrangement of r elements of a set is called an **r-permutation**.

Fall 2002

CMSC 203 - Discrete Structures

238

Permutations and Combinations

Example: Let $S = \{1, 2, 3\}$. The arrangement 3, 1, 2 is a permutation of S. The arrangement 3, 2 is a 2-permutation of S. The number of r-permutations of a set with n distinct elements is denoted by P(n, r).

We can calculate P(n, r) with the product rule: P(n, r) = $n \cdot (n - 1) \cdot (n - 2) \cdot ... \cdot (n - r + 1)$. (n choices for the first element, (n - 1) for the second one, (n - 2) for the third one...)

CMSC 203 - Discrete Structures

Fall 2002

239

Permutations and Combinations

Example:

 $P(8, 3) = 8 \cdot 7 \cdot 6 = 336$ $= (8 \cdot 7 \cdot 6 \cdot 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1) / (5 \cdot 4 \cdot 3 \cdot 2 \cdot 1)$

General formula:

P(n, r) = n!/(n - r)!

Knowing this, we can return to our initial question: How many ways are there to pick a set of 3 people from a group of 6 (disregarding the order of picking)?

CMSC 203 - Discrete Structures

Fall 2002

Permutations and Combinations

An **r-combination** of elements of a set is an unordered selection of r elements from the set. Thus, an r-combination is simply a subset of the set with r elements.

Example: Let S = $\{1, 2, 3, 4\}$. Then $\{1, 3, 4\}$ is a 3-combination from S.

The number of r-combinations of a set with n distinct elements is denoted by C(n, r).

Example: C(4, 2) = 6, since, for example, the 2-combinations of a set $\{1, 2, 3, 4\}$ are $\{1, 2\}$, $\{1, 3\}$, $\{1, 4\}$, $\{2, 3\}$, $\{2, 4\}$, $\{3, 4\}$.

CMSC 203 - Discrete Structures

241

Permutations and Combinations

How can we calculate C(n, r)?

Consider that we can obtain the r-permutation of a set in the following way:

First, we form all the r-combinations of the set (there are C(n, r) such r-combinations). Then, we generate all possible orderings in each of

these r-combinations (there are P(r, r) such orderings in each case).

CMSC 203 - Discrete Structures

Therefore, we have:

 $P(n, r) = C(n, r) \cdot P(r, r)$

Fall 2002

Fall 2002

Fall 2002

242

Permutations and Combinations C(n, r) = P(n, r)/P(r, r)= n!/(n - r)!/(r!/(r - r)!)

= n!/(r!(n - r)!)

Now we can answer our initial question: How many ways are there to pick a set of 3 people from a group of 6 (disregarding the order of picking)?

 $C(6, 3) = 6!/(3!\cdot3!) = 720/(6\cdot6) = 720/36 = 20$

There are 20 different ways, that is, 20 different groups to be picked.

CMSC 203 - Discrete Structures

Permutations and Combinations

Corollary:

Fall 2002

Fall 2002

Let n and r be nonnegative integers with $r \le n$. Then C(n, r) = C(n, n - r).

Note that "picking a group of r people from a group of n people" is the same as "splitting a group of n people into a group of r people and another group of (n - r) people".

244

245

Please also look at proof on page 252.

CMSC 203 - Discrete Structures

Permutations and Combinations Example: A soccer club has 8 female and 7 male members. For today's match, the coach wants to have 6 female and 5 male players on the grass. How many possible configurations are there? $C(8, 6) \cdot C(7, 5) = 8!/(6! \cdot 2!) \cdot 7!/(5! \cdot 2!)$ $= 28 \cdot 21$ = 588

CMSC 203 - Discrete Structures

CombinationsWe also saw the following: $C(n,n-r) = \frac{n!}{(n-r)![n-(n-r)]!} = \frac{n!}{(n-r)!r!} = C(n,r)$ This symmetry is intuitively plausible. For example,
let us consider a set containing six elements (n = 6).Picking two elements and leaving four is essentially
the same as picking four elements and leaving two.In either case, our number of choices is the
number of possibilities to divide the set into one
set containing two elements.Full 202CMSC 203 - District Structure

Combinations

CMSC 203 - Discrete Structures

Pascal's Identity:

Let n and k be positive integers with $n \ge k$. Then C(n + 1, k) = C(n, k - 1) + C(n, k).

How can this be explained?

What is it good for?

Fall 2002













Binomial Coefficients

For $(a + b)^3 = (a + b)(a + b)(a + b)$ we have

 $(a + b)^3 = aaa + aab + aba + abb + baa + bab + bba + bbb$ $(a + b)^3 = a^3 + 3a^2b + 3ab^2 + b^3$

There is only one term a^3 , because there is only one possibility to form it: Choose **a** from all three factors: C(3, 3) = 1.

There is three times the term a^2b , because there are three possibilities to choose **a** from two out of the three factors: C(3, 2) = 3.

Similarly, there is three times the term ab^2 (C(3, 1) = 3) and once the term b^3 (C(3, 0) = 1).

CMSC 203 - Discrete Structures

Fall 2002

Binomial Coefficients This leads us to the following formula: $(a+b)^n = \sum^n C(n,j) \cdot a^{n-j} b^j$ (Binomial Theorem) With the help of Pascal's triangle, this formula can considerably simplify the process of expanding powers of binomial expressions. For example, the fifth row of Pascal's triangle (1 - 4 - 6 - 4 - 1) helps us to compute $(a + b)^4$: $(a + b)^4 = a^4 + 4a^3b + 6a^2b^2 + 4ab^3 + b^4$ Fall 2002 CMSC 203 - Discrete Structures 253





Recurrence Relations

A **recurrence relation** for the sequence $\{a_n\}$ is an equation that expresses a_n is terms of one or more of the previous terms of the sequence, namely, a_0 , a_1 , ..., a_{n-1} , for all integers n with $n \ge n_0$, where n_0 is a nonnegative integer.

A sequence is called a **solution** of a recurrence relation if it terms satisfy the recurrence relation.

CMSC 203 - Discrete Structures

Fall 2002

Recurrence Relations

In other words, a recurrence relation is like a recursively defined sequence, but without specifying any initial values (initial conditions).

Therefore, the same recurrence relation can have (and usually has) **multiple solutions**.

If **both** the initial conditions and the recurrence relation are specified, then the sequence is **uniquely** determined.

CMSC 203 - Discrete Structures

256

Example:Consider the recurrence relation $a_n = 2a_{n-1} - a_{n-2}$ for n = 2, 3, 4, ...

Is the sequence $\{a_n\}$ with $a_n=3n$ a solution of this recurrence relation? For $n\geq 2$ we see that $2a_{n-1}-a_{n-2}=2(3(n-1))-3(n-2)=3n=a_n.$ Therefore, $\{a_n\}$ with $a_n=3n$ is a solution of the recurrence relation.

CMSC 203 - Discrete Structures

Fall 2002

Fall 2002

257

Recurrence Relations

Is the sequence $\{a_n\}$ with $a_n\mbox{=}5$ a solution of the same recurrence relation?

For $n \ge 2$ we see that

 $2a_{n-1} - a_{n-2} = 2.5 - 5 = 5 = a_n.$

Therefore, $\{a_n\}$ with $a_n\text{=}5$ is also a solution of the recurrence relation.

CMSC 203 - Discrete Structures

Fall 2002

Modeling with Recurrence Relations **Example:** Someone deposits \$10,000 in a savings account at a bank yielding 5% per year with interest compounded annually. How much money will be in the account after 30 years? **Solution:** Het P_n denote the amount in the account after n years. How can we determine P_n on the basis of P_{n-1} ?

 $\label{eq:spherical states} \begin{array}{l} \mbox{Modeling with Recurrence Relations} \\ \mbox{We can derive the following recurrence relation:} \\ P_n = P_{n-1} + 0.05P_{n-1} = 1.05P_{n-1}. \\ \mbox{The initial condition is } P_0 = 10,000. \\ \mbox{Then we have:} \\ P_1 = 1.05P_0 \\ P_2 = 1.05P_1 = (1.05)^2P_0 \\ P_3 = 1.05P_2 = (1.05)^3P_0 \\ \mbox{...} \\ P_n = 1.05P_{n-1} = (1.05)^nP_0 \\ \mbox{We now have a formula to calculate } P_n \mbox{ for any natural number n and can avoid the iteration.} \\ \mbox{Full 202} \end{array}$

Modeling with Recurrence Relations Let us use this formula to find P_{30} under the initial condition $P_0 = 10,000$: $P_{30} = (1.05)^{30}.10,000 = 43,219.42$ After 30 years, the account contains \$43,219.42.

CMSC 203 - Discrete Structures

Fall 2002

Modeling with Recurrence Relations

Another example:

Let a_n denote the number of bit strings of length n that do not have two consecutive Os ("valid strings"). Find a recurrence relation and give initial conditions for the sequence $\{a_n\}$.

Solution:

Fall 2002

I dea: The number of valid strings equals the number of valid strings ending with a 0 plus the number of valid strings ending with a 1.

CMSC 203 - Discrete Structures

262



Modeling with Recurrence Relations Now we need to know: How many valid strings of length n are there, if the string ends with a **0**? Valid strings of length n ending with a 0 **must have a 1 as their (n - 1)st bit** (otherwise they would end with 00 and would not be valid). And what is the number of valid strings of length (n - 1) that end with a 1? We already know that there are a_{n-1} strings of length n that end with a 1. Therefore, there are a_{n-2} strings of length (n - 1)

that end with a 1.

Fall 2002 CMSC 203 - Discrete Structures

Modeling with Recurrence Relations

So there are a_{n-2} valid strings of length n that end with a 0 (all valid strings of length (n – 2) with 10 appended to them).

As we said before, the number of valid strings is the number of valid strings ending with a 0 plus the number of valid strings ending with a 1.

That gives us the following **recurrence relation**: $a_n = a_{n-1} + a_{n-2}$

CMSC 203 - Discrete Structures

Fall 2002

265

Modeling with Recurrence Relations What are the initial conditions? $a_1 = 2 (0 \text{ and } 1)$ $a_2 = 3 (01, 10, \text{ and } 11)$ $a_3 = a_2 + a_1 = 3 + 2 = 5$ $a_4 = a_3 + a_2 = 5 + 3 = 8$ $a_5 = a_4 + a_3 = 8 + 5 = 13$... This sequence satisfies the same recurrence relation as the Fibonacci sequence. Since $a_1 = f_3$ and $a_2 = f_4$, we have $a_n = f_{n+2}$. Fall 202 CMSC 203-Discret Structure



In general, we would prefer to have an **explicit** formula to compute the value of a_n rather than conducting n iterations.

For one class of recurrence relations, we can obtain such formulas in a systematic way.

Those are the recurrence relations that express the terms of a sequence as **linear combinations** of previous terms.

CMSC 203 - Discrete Structures

Solving Recurrence Relations

Definition: A linear homogeneous recurrence relation of degree k with constant coefficients is a recurrence relation of the form:

 $\begin{array}{l} a_n=c_1a_{n-1}+c_2a_{n-2}+...+c_ka_{n-k'}\\ Where \ c_1,\ c_2,\ ...,\ c_k \ are \ real \ numbers, \ and \ c_k\neq 0. \end{array}$

A sequence satisfying such a recurrence relation is uniquely determined by the recurrence relation and the k initial conditions

 $a_0 = C_{0'} \ a_1 = C_{1'} \ a_2 = C_{2'} \ \dots, \ a_{k-1} = C_{k-1}.$

CMSC 203 - Discrete Structures

268

Solving Recurrence Relations

Examples:

Fall 2002

The recurrence relation $P_n = (1.05)P_{n-1}$ is a linear homogeneous recurrence relation of **degree one**.

The recurrence relation $f_n = f_{n-1} + f_{n-2}$ is a linear homogeneous recurrence relation of **degree two**.

The recurrence relation a_n = $a_{n\mbox{-}5}$ is a linear homogeneous recurrence relation of degree five.

CMSC 203 - Discrete Structures

Fall 2002



Solving Recurrence Relations The solutions of this equation are called the **characteristic roots** of the recurrence relation. Let us consider linear homogeneous recurrence relations of degree two. **Theorem:** Let c_1 and c_2 be real numbers. Suppose that $r^2 - c_1 r - c_2 = 0$ has two distinct roots r_1 and r_2 . Then the sequence $\{a_n\}$ is a solution of the recurrence relation $a_n = c_1 a_{n-1} + c_2 a_{n-2}$ if and only if $a_n = \alpha_1 r_1^n + \alpha_2 r_2^n$ for n = 0, 1, 2, ..., where α_1 and α_2 are constants. See pp. 321 and 322 for the proof.

Fall 2002

CMSC 203 - Discrete Structures

271





Solving Recurrence Relations

 $\begin{array}{l} a_n=r^n \mbox{ is a solution of the linear homogeneous} \\ recurrence relation \\ a_n=c_1a_{n-1}+c_2a_{n-2}+\ldots+c_ka_{n-k} \\ \mbox{ if and only if} \\ r^n=c_1r^{n-1}+c_2r^{n-2}+\ldots+c_kr^{n-k}. \\ \mbox{ Divide this equation by } r^{n-k} \mbox{ and subtract the } \\ \mbox{ right-hand side from the left:} \\ r^k-c_1r^{k-1}-c_2r^{k-2}-\ldots-c_{k-1}r-c_k=0 \\ \mbox{ This is called the$ **characteristic equation** $of the recurrence relation. \\ \end{array}$





CMSC 203 - Discrete Structures

Fall 2002

Solving Recurrence Relations Therefore, the Fibonacci numbers are given by $+a_{3}$ for some constants α_1 and α_2 . We can determine values for these constants so that the sequence meets the conditions $f_0 = 0$ and $f_1 = 1$: $f_0 = \boldsymbol{a}_1 + \boldsymbol{a}_2 = 0$

CMSC 203 - Discrete Structures

277

Fall 2002







Solving Recurrence Relations Example: What is the solution of the recurrence relation $a_n = 6a_{n-1} - 9a_{n-2}$ with $a_0 = 1$ and $a_1 = 6$? Solution: The only root of $r^2 - 6r + 9 = 0$ is $r_0 = 3$. Hence, the solution to the recurrence relation is $a_n = \alpha_1 3^n + \alpha_2 n 3^n$ for some constants α_1 and α_2 . To match the initial condition, we need $a_0 = 1 = \alpha_1$ $a_1 = 6 = \alpha_1 \cdot 3 + \alpha_2 \cdot 3$ Solving these equations yields $\alpha_1 = 1$ and $\alpha_2 = 1$. Consequently, the overall solution is given by $a_n = 3^n + n 3^n$. Ful 202 CMSC 2014 Constants



Relations

I f we want to describe a relationship between elements of two sets A and B, we can use **ordered pairs** with their first element taken from A and their second element taken from B.

Since this is a relation between **two sets**, it is called a **binary relation**.

Definition: Let A and B be sets. A binary relation from A to B is a subset of $A \times B$.

In other words, for a binary relation R we have $R \subseteq A \times B$. We use the notation aRb to denote that $(a, b) \in R$ and aRb to denote that $(a, b) \notin R$.

CMSC 203 - Discrete Structures

Fall 2002

Relations

When (a, b) belongs to R, a is said to be **related** to b by R.

Example: Let P be a set of people, C be a set of cars, and D be the relation describing which person drives which car(s).

P = {Carl, Suzanne, Peter, Carla},

C = {Mercedes, BMW, tricycle}

Fall 2002

Fall 2002

D = {(Carl, Mercedes), (Suzanne, Mercedes), (Suzanne, BMW), (Peter, tricycle)}

This means that Carl drives a Mercedes, Suzanne drives a Mercedes and a BMW, Peter drives a tricycle, and Carla does not drive any of these vehicles.

CMSC 203 - Discrete Structures

283

Functions as Relations

You might remember that a **function** f from a set A to a set B assigns a unique element of B to each element of A.

The **graph** of f is the set of ordered pairs (a, b) such that b = f(a).

Since the graph of f is a subset of A×B, it is a **relation** from A to B.

Moreover, for each element **a** of A, there is exactly one ordered pair in the graph that has **a** as its first element.

CMSC 203 - Discrete Structures

284

Functions as Relations

Conversely, if R is a relation from A to B such that every element in A is the first element of exactly one ordered pair of R, then a function can be defined with R as its graph.

This is done by assigning to an element $a \in A$ the unique element $b \in B$ such that $(a, b) \in R$.

CMSC 203 - Discrete Structures

Fall 2002







Relations on a Set

How many different relations can we define on a set \boldsymbol{A} with \boldsymbol{n} elements?

A relation on a set A is a subset of A×A. How many elements are in A×A ?

There are n^2 elements in A×A, so how many subsets (= relations on A) does A×A have?

The number of subsets that we can form out of a set with m elements is 2^m . Therefore, 2^{n^2} subsets can be formed out of A×A.

Answer: We can define 2^{n^2} different relations on A.

CMSC 203 - Discrete Structures

Fall 2002

Pr	operties of Relations	
We will now I relations.	ook at some useful ways to cl	assify
Definition: A	relation R on a set A is calle	d
reflexive if (a, a)∈R for every element a∈	A.
Are the follo	wing relations on {1, 2, 3, 4} r	eflexive?
R = {(1, 1), (1,	2), (2, 3), (3, 3), (4, 4)}	No.
R = {(1, 1), (2	, 2), (2, 3), (3, 3), (4, 4)}	Yes.
R = {(1, 1), (2	, 2), (3, 3)}	No.
Definition: A relation on a set A is called irreflexive if (a, a)∉ R for every element a∈ A.		
Fall 2002	CMSC 203 - Discrete Structures	289



Properties of Relations	
Definitions:	
A relation R on a set A is called symmetric if $a \in R$ whenever $(a, b) \in R$ for all $a, b \in A$.	(b,
A relation R on a set A is called antisymmetric if $a = b$ whenever $(a, b) \in R$ and $(b, a) \in R$.	
A relation R on a set A is called asymmetric if $(a, b) \in R$ implies that $(b, a) \notin R$ for all $a, b \in A$.	
Fall 2002 CMSC 203 - Discrete Structures	290

Properties of Relations		
Are the following relations on {1, 2, 3, 4} symmetric, antisymmetric, or asymmetric?		
R = {(1, 1), (1, 2), (2, 1), (3, 3), (4, 4)} R = {(1, 1)}	symmetric sym. and antisym.	
R = {(1, 3), (3, 2), (2, 1)}	antisym. and asym.	
$R = \{(4, 4), (3, 3), (1, 4)\}$	antisym.	
Fall 2002 CMSC 203 - Discrete Structures	291	



Prope	erties of Relation	S
Definition: A relation R on a set A is called transitive if whenever $(a, b) \in R$ and $(b, c) \in R$, then $(a, c) \in R$ for a, b, $c \in A$.		
Are the following transitive?	relations on {1, 2, 3, 4	1}
R = {(1, 1), (1, 2), ((2, 2), (2, 1), (3, 3)}	Yes.
$R=\{(1,3),(3,2),$	(2, 1)}	No.
$R = \{(2, 4), (4, 3),$	(2, 3), (4, 1)}	No.
Fall 2002	CMSC 203 - Discrete Structures	292

Со	unting Relations	
Example: How ma can be defined or	any different reflexi n a set A containing r	ve relations elements?
Solution: Relatio contains n^2 element Therefore, differ generated by cho these n^2 element A reflexive relat elements (a, a) for Consequently, we n(n - 1) elements there are $2^{n(n - 1)}$	ns on R are subsets c ents. rent relations on A c bosing different subs s, so there are 2^{n^2} re ion, however, must co or every $a \in A$. can only choose amo to generate reflexiv of them.	of A×A, which an be sets out of elations. ontain the n ng n ² – n = e relations, so
Fall 2002	CMSC 203 - Discrete Structures	293

Combining Relations

Relations are sets, and therefore, we can apply the usual **set operations** to them.

If we have two relations R_1 and R_2 , and both of them are from a set A to a set B, then we can combine them to $R_1 \cup R_2$, $R_1 \cap R_2$, or $R_1 - R_2$.

In each case, the result will be another relation from A to B.

CMSC 203 - Discrete Structures

Fall 2002

Combining Relations

 \ldots and there is another important way to combine relations.

Definition: Let R be a relation from a set A to a set B and S a relation from B to a set C. The **composite** of R and S is the relation consisting of ordered pairs (a, c), where $a \in A$, $c \in C$, and for which there exists an element $b \in B$ such that (a, b) $\in R$ and (b, c) $\in S$. We denote the composite of R and S by $S \circ R$.

In other words, if relation R contains a pair (a, b) and relation S contains a pair (b, c), then S-R contains a pair (a, c).

CMSC 203 - Discrete Structures

Fall 2002

295

Combining Relations Example: Let D and S be relations on A = {1, 2, 3, 4}. D = {(a, b) | b = 5 - a} "b equals (5 - a)" S = {(a, b) | a < b}</td> "a is smaller than b" D = {(1, 4), (2, 3), (3, 2), (4, 1)} S = {(1, 2), (1, 3), (1, 4), (2, 3), (2, 4), (3, 4)} S • D = {(2, 4), (3, 3), (3, 4), (4, 2), (4, 3), (4, 4)} D maps an element a to the element (5 - a), and afterwards S maps (5 - a) to all elements larger than (5 - a), resulting in S°D = {(a,b) | b > 5 - a} or S°D = {(a,b) | a + b > 5}.

Combining Relations

We already know that **functions** are just **special cases** of **relations** (namely those that map each element in the domain onto exactly one element in the codomain).

If we formally convert two functions into relations, that is, write them down as sets of ordered pairs, the composite of these relations will be exactly the same as the composite of the functions (as defined earlier).

CMSC 203 - Discrete Structures

Combining Relations

Definition: Let R be a relation on the set A. The powers R^n , n = 1, 2, 3, ..., are defined inductively by $R^1 = R$ $\mathbb{R}^{n+1} = \mathbb{R}^{n} \circ \mathbb{R}$

In other words:

Fall 2002

Fall 2002

 R^n = $R^{\circ}R^{\circ} \dots {}^{\circ}R$ (n times the letter R)

298

299

Combining Relations

CMSC 203 - Discrete Structures

Theorem: The relation R on a set A is transitive if and only if $\mathbb{R}^n \subseteq \mathbb{R}$ for all positive integers n. Remember the definition of transitivity: Definition: A relation R on a set A is called transitive if whenever $(a, b) \in R$ and $(b, c) \in R$, then $(a, c) \in R$ for $a, b, c \in A$. The composite of R with itself contains exactly these pairs (a, c). Therefore, for a transitive relation R, R°R does not contain any pairs that are not in R, so $R \circ R \subseteq R$. Since R^oR does not introduce any pairs that are not already in R, it must also be true that $(R \circ R) \circ R \subseteq R$, and so on, so that $R^n \subseteq R$. CMSC 203 - Discrete Structures

n-ary Relations

In order to study an interesting application of relations, namely databases, we first need to generalize the concept of binary relations to **n-ary** relations.

Definition: Let $A_{1'}$, $A_{2'}$, ..., A_n be sets. An **n-ary relation** on these sets is a subset of $A_1 \times A_2 \times ... \times A_n$. The sets A_{11} , A_{22} , ..., A_n are called the **domains** of the relation, and n is called its degree.

CMSC 203 - Discrete Structures

Fall 2002

n-ary Relations
Example:
Let R = {(a, b, c) a = $2b \land b = 2c$ with a, b, $c \in \mathbf{N}$ }
What is the degree of R?

What is t The degree of R is 3, so its elements are triples. What are its domains? Its domains are all equal to the set of integers. Is (2, 4, 8) in R? No. Is (4, 2, 1) in R? Yes.

CMSC 203 - Discrete Structures

301

Fall 2002

Databases and Relations Let us take a look at a type of database representation that is based on relations, namely the relational data model. A database consists of n-tuples called **records**, which are made up of fields. These fields are the **entries** of the n-tuples. The relational data model represents a database as an n-ary relation, that is, a set of records. Fall 2002 CMSC 203 - Discrete Structures 302

Databases and Relations

Example: Consider a database of students, whose records are represented as 4-tuples with the fields Student Name, ID Number, Major, and GPA:

R = {(Ackermann, 231455, CS, 3.88), (Adams, 888323, Physics, 3.45), (Chou, 102147, CS, 3.79), (Goodfriend, 453876, Math, 3.45), (Rao, 678543, Math, 3.90), (Stevens, 786576, Psych, 2.99)}

Relations that represent databases are also called tables, since they are often displayed as tables. CMSC 203 - Discrete Structures

Fall 2002

Databases and Relations

A domain of an n-ary relation is called a **primary key** if the n-tuples are uniquely determined by their values from this domain.

This means that no two records have the same value from the same primary key.

In our example, which of the fields **Student Name**, **ID Number**, **Major**, and **GPA** are primary keys?

Student Name and **ID Number** are primary keys, because no two students have identical values in these fields.

In a real student database, only $\ensuremath{\text{ID}}$ Number would be a primary key.

CMSC 203 - Discrete Structures

Databases and Relations

In a database, a primary key should remain one even if new records are added.

Therefore, we should use a primary key of the **intension** of the database, containing all the n-tuples that can ever be included in our database.

Combinations of domains can also uniquely identify n-tuples in an n-ary relation.

When the values of a **set of domains** determine an n-tuple in a relation, the **Cartesian product** of these domains is called a **composite key**.

CMSC 203 - Discrete Structures

Fall 2002

Fall 2002

305

306

304

Databases and Relations

We can apply a variety of **operations** on n-ary relations to form new relations.

 $\begin{array}{l} \textbf{Definition: The projection } \mathsf{P}_{i_1,\,i_2,\,\ldots,\,i_m} \text{ maps the n-tuple } (a_1,\,a_2,\,\ldots,\,a_n) \text{ to the m-tuple } (a_{i_1}^{},\,a_{i_2}^{},\,\ldots,\,a_{i_m}^{}), \\ \text{where } m \leq n. \end{array}$

In other words, a projection $\mathsf{P}_{i_1,\,i_2,\,\cdots,\,i_m}$ keeps the m components $a_{i_1},\,a_{i_2},\,\cdots,\,a_{i_m}$ of an h-tuple and deletes its (n – m) other components.

Example: What is the result when we apply the projection $P_{2,4}$ to the student record (Stevens, 786576, Psych, 2.99) ?

Solution: It is the pair (786576, 2.99). Fall 2002 CMSC 203 - Discrete Structures

Databases and Relations

In some cases, applying a projection to an entire table may not only result in fewer columns, but also in **fewer rows**.

Why is that?

Some records may only have differed in those fields that were deleted, so they become **identical**, and there is no need to list identical records more than once.

Fall 2002

CMSC 203 - Discrete Structures

307

Databases and Relations

We can use the **join** operation to combine two tables into one if they share some identical fields.

 $\begin{array}{l} \mbox{Definition: Let R be a relation of degree m and S a relation of degree n. The join <math display="inline">J_p(R, S), \mbox{ where } p \leq m \\ \mbox{and } p \leq n, \mbox{ is a relation of degree } m + n - p \mbox{ that } \\ \mbox{consists of all } (m + n - p) \mbox{-tuples} \\ (a_1, a_2, ..., a_{m-p'}, c_1, c_2, ..., c_p, b_1, b_2, ..., b_{n-p}), \\ \mbox{where the m-tuple } (a_1, a_2, ..., a_{m-p'}, c_1, c_2, ..., c_p) \\ \mbox{belongs to R and the n-tuple } (c_1, c_2, ..., c_p, b_1, b_2, ..., c_p) \\ \end{array}$

belongs to R and the n-tuple $(c_1, c_2, ..., c_p, b_1, b_2, ..., b_{n-p})$ belongs to S.

CMSC 203 - Discrete Structures

Fall 2002

308

Databases and Relations

In other words, to generate Jp(R, S), we have to find all the elements in R whose p last components match the p first components of an element in S.

The new relation contains exactly these matches, which are combined to tuples that contain each matching field only once.

CMSC 203 - Discrete Structures

Fall 2002



 $\begin{array}{l} \mbox{Databases and Relations} \\ \mbox{Solution: The resulting relation is:} \\ & \{(1978, Ackermann, 231455, CS, 3.88), \\ & (1972, Adams, 888323, Physics, 3.45), \\ & (1972, Physics, 888323, Physics, 88323, Physics, 883$

Representing Relations

We already know different ways of representing relations. We will now take a closer look at two ways of representation: **Zero-one matrices** and **directed graphs**.

If R is a relation from A = {a₁, a₂, ..., a_m} to B = {b₁, b₂, ..., b_n}, then R can be represented by the zero-one matrix $M_R = [m_{ij}]$ with

 $m_{ij} = 1$, if $(a_i, b_j) \in R$, and

 $m_{ij} = 0$, if $(a_i, b_j) \notin R$.

Note that for creating this matrix we first need to list the elements in A and B in a **particular**, **but arbitrary order**.

Fall 2002 CMSC 203 - Discrete Structures

F	Representing Relations	
Example: How can we represent the relation R = {(2, 1), (3, 1), (3, 2)} as a zero-one matrix?		
Solution : The matrix M_R is given by		
$M_{R} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 1 \end{bmatrix}$		
Fall 2002	CMSC 203 - Discrete Structures	313











Representing Relations

The Boolean operations **join** and **meet** (you remember?) can be used to determine the matrices representing the **union** and the **intersection** of two relations, respectively.

To obtain the **join** of two zero-one matrices, we apply the Boolean "or" function to all corresponding elements in the matrices.

To obtain the **meet** of two zero-one matrices, we apply the Boolean "and" function to all corresponding elements in the matrices.

316

CMSC 203 - Discrete Structures

Fall 2002





Representing Relations Using MatricesExample: How can we represent the relation
 $R = \{(2, 1), (3, 1), (3, 2)\}$ as a zero-one matrix?Solution: The matrix M_R is given by $M_R = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 1 \end{bmatrix}$







Representing Relations Using Matrices Let us now assume that the zero-one matrices $M_A = [a_{ij}], M_B = [b_{ij}] \text{ and } M_C = [c_{ij}] \text{ represent}$ relations A, B, and C, respectively. **Remember:** For $M_C = M_A o M_B$ we have:

 $c_{ij} = 1$ if and only if at least one of the terms $(a_{in} \wedge b_{nj}) = 1$ for some n; otherwise $c_{ij} = 0$.

In terms of the **relations**, this means that C contains a pair (x_i, z_j) if and only if there is an element y_n such that (x_i, y_n) is in relation A and (y_n, z_j) is in relation B.

CMSC 203 - Discrete Structures

Therefore, $C = B \cdot A$ (composite of A and B).

Fall 2002







Representing Relations Using Digraphs Definition: A directed graph, or digraph, consists of a set V of vertices (or nodes) together with a set E of ordered pairs of elements of V called edges (or arcs). The vertex a is called the initial vertex of the edge (a, b), and the vertex b is called the terminal vertex of this edge. We can use arrows to display graphs.

CMSC 203 - Discrete Structures




Representing Relations Using DigraphsObviously, we can represent any relation R on a set
A by the digraph with A as its vertices and all pairs
(a, b) \in R as its edges.Vice versa, any digraph with vertices V and edges E
can be represented by a relation on V containing all
the pairs in E.This one-to-one correspondence between
relations and digraphs means that any statement
about relations also applies to digraphs, and vice
versa.

Equivalence Relations

Equivalence relations are used to relate objects that are similar in some way.

Definition: A relation on a set A is called an equivalence relation if it is reflexive, symmetric, and transitive.

Two elements that are related by an equivalence relation R are called **equivalent**.

CMSC 203 - Discrete Structures



Since R is $\ensuremath{\textbf{symmetric}}$, a is equivalent to b whenever b is equivalent to a.

Since R is $\ensuremath{\textbf{reflexive}}$, every element is equivalent to itself.

Since R is **transitive**, if a and b are equivalent and b and c are equivalent, then a and c are equivalent.

Obviously, these three properties are necessary for a reasonable definition of equivalence.

CMSC 203 - Discrete Structures

328

Equivalence Relations

Example: Suppose that R is the relation on the set of strings that consist of English letters such that aRb if and only if I(a) = I(b), where I(x) is the length of the string x. Is R an equivalence relation? **Solution:**

- R is reflexive, because I(a) = I(a) and therefore aRa for any string a.
- R is symmetric, because if I(a) = I(b) then I(b) = I(a), so if aRb then bRa.
- R is transitive, because if I(a) = I(b) and I(b) = I(c), then I(a) = I(c), so aRb and bRc implies aRc.
- R is an equivalence relation. Fall 2002 CMSC 203 - Discrete Structures

Fall 2002

329

Equivalence Classes

Definition: Let R be an equivalence relation on a set A. The set of all elements that are related to an element a of A is called the **equivalence class** of a.

The equivalence class of a with respect to R is denoted by $\left[a\right]_{R^{\text{.}}}$

When only one relation is under consideration, we will delete the subscript R and write **[a]** for this equivalence class.

If $b \in [a]_{R'}$ b is called a **representative** of this equivalence class.

CMSC 203 - Discrete Structures

Fall 2002

Equivalence Classes

Example: In the previous example (strings of identical length), what is the equivalence class of the word mouse, denoted by [mouse] ?

Solution: [mouse] is the set of all English words containing five letters.

For example, 'horse' would be a representative of this equivalence class.

CMSC 203 - Discrete Structures

Fall 2002



Equivalence Classes									
Examples: Let S be the set {u, m, b, r, o, c, k, s}. Do the following collections of sets partition S ?									
{{m, o, c, k}, {r, u, b, s}}	yes.								
{{c, o, m, b}, {u, s}, {r}}	no (k is missing).								
{{b, r, o, c, k}, {m, u, s, t}}	no (t is not in S).								
{{u, m, b, r, o, c, k, s}}	yes.								
{{b, o, o, k}, {r, u, m}, {c, s}}	yes ({ $b,0,0,k$ } = { $b,0,k$ }).								
	no (Ø not allowed). e Structures 333								



Equivalence Classes

Theorem: Let R be an equivalence relation on a set S. Then the **equivalence classes** of R form a **partition** of S. Conversely, given a partition $\{A_i \mid i \in I\}$ of the set S, there is an equivalence relation R that has the sets A_i , $i \in I$, as its equivalence classes.

Fall 2002

334

Equivalence Classes

CMSC 203 - Discrete Structures

Example: Let us assume that Frank, Suzanne and George live in Boston, Stephanie and Max live in Lübeck, and Jennifer lives in Sydney.

Let R be the **equivalence relation** {(a, b) | a and b live in the same city} on the set P = {Frank, Suzanne, George, Stephanie, Max, Jennifer}.

Then R = {(Frank, Frank), (Frank, Suzanne), (Frank, George), (Suzanne, Frank), (Suzanne, Suzanne), (Suzanne, George), (George, Frank), (George, Suzanne), (George, George), (Stephanie, Stephanie), (Stephanie, Max), (Max, Stephanie), (Max, Max), (Jennifer, Jennifer)}. Fall 2002 CMSC 2013- Discrete Structures 335

Equivalence Classes

Then the equivalence classes of R are:

{{Frank, Suzanne, George}, {Stephanie, Max}, {Jennifer}}.

This is a partition of P.

The equivalence classes of any equivalence relation R defined on a set S constitute a partition of S, because every element in S is assigned to **exactly one** of the equivalence classes.

CMSC 203 - Discrete Structures

Equivalence Classes			
Another example: Let R be the relation {(a, b) $a \equiv b \pmod{3}$ } on the set of integer Is R an equivalence relation? Yes, R is reflexive, symmetric, and transit	ers. ive.		
Another example: Let R be the relation {(a, b) $a \equiv b \pmod{3}$ } on the set of integers. Is R an equivalence relation? Yes, R is reflexive, symmetric, and transitive. What are the equivalence classes of R? {{, -6, -3, 0, 3, 6,}, {, -5, -2, 1, 4, 7,}, {, -4, -1, 2, 5, 8,}}			
E-II 2002 CMSC 202 Discrete Structures	337		





Boolean product

Fall 2002

CMSC 203 - Discrete Structures









CMSC 203 - Discrete Structures

Boolean Functions and Expressions

For example, we can create Boolean expression in the variables x, y, and z using the "building blocks" 0, 1, x, y, and z, and the construction rules:

Since x and y are Boolean expressions, so is xy.

Since z is a Boolean expression, so is (-z).

Since xy and (-z) are expressions, so is xy + (-z). ... and so on...

CMSC 203 - Discrete Structures

Fall 2002









Boolean Functions and Expressions

There is a simple method for deriving a Boolean expression for a function that is defined by a table. This method is based on **minterms**.

Definition: A **literal** is a Boolean variable or its complement. A **minterm** of the Boolean variables x_1 , x_2 , ..., x_n is a Boolean product $y_1y_2...y_n$, where $y_i = x_i$ or $y_i = -x_i$.

Hence, a minterm is a product of n literals, with one literal for each variable.

CMSC 203 - Discrete Structures

Fall 2002

346





Boolean Functions and Expressions

Definition: The Boolean functions F and G of n variables are **equal** if and only if $F(b_1, b_2, ..., b_n) = G(b_1, b_2, ..., b_n)$ whenever $b_1, b_2, ..., b_n$ belong to B. Two different Boolean expressions that represent the same function are called **equivalent**.

For example, the Boolean expressions xy, xy + 0, and $xy \cdot 1$ are equivalent.

CMSC 203 - Discrete Structures

Fall 2002

Boolean Functions and Expressions The complement of the Boolean function F is the function -F, where -F(b_1 , b_2 , ..., b_n) = -(F(b_1 , b_2 , ..., b_n)). Let F and G be Boolean functions of degree n. The Boolean sum F+G and Boolean product FG are then defined by (F + G)(b_1 , b_2 , ..., b_n) = F(b_1 , b_2 , ..., b_n) + G(b_1 , b_2 , ..., b_n) (FG)(b_1 , b_2 , ..., b_n) = F(b_1 , b_2 , ..., b_n) G(b_1 , b_2 , ..., b_n)



Boolean Functions and Expressions																	
Question: How many different Boolean functions of degree 2 are there?																	
Solution: There are 16 of them, $F_{1^{\prime}}F_{2^{\prime}},F_{16}\!:$																	
х	у	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F,	F ₁₀	F ₁₁	F_{12}	F ₁₃	F_{14}	F ₁₅	F ₁₆
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
Fall 2002 CMSC 203 - Discrete Structures									351								





Question: How many different Boolean functions of degree n are there?

Solution:

There are 2ⁿ different n-tuples of 0s and 1s.

A Boolean function is an assignment of 0 or 1 to each of these 2^n different n-tuples.

CMSC 203 - Discrete Structures

Therefore, there are 2^{2^n} different Boolean functions.

Fall 2002

352

Duality There are useful identities of Boolean expressions that can help us to transform an expression A into an equivalent expression B (see Table 5 on page 597 in the textbook). We can derive additional identities with the help of the **dual** of a Boolean expression. The dual of a Boolean expression is obtained by interchanging Boolean sums and Boolean products and interchanging 0s and 1s.

Fall 2002

353

Duality

CMSC 203 - Discrete Structures

Examples:

Fall 2002

The dual of x(y + z) is x + yz.

The dual of $-x \cdot 1 + (-y + z)$ is (-x + 0)((-y)z).

The **dual of a Boolean function F** represented by a Boolean expression is the function represented by the dual of this expression.

This dual function, denoted by F^d , **does not depend** on the particular Boolean expression used to represent F.

CMSC 203 - Discrete Structures

Duality

Therefore, an identity between functions represented by Boolean expressions remains valid when the duals of both sides of the identity are

We can use this fact, called the to derive new identities.

For example, consider the absorption law

By taking the duals of both sides of this identity, x + xy , which is also an identity (and also called an absorption law).

CMSC 203 - Discrete Structures

355

All the properties of Boolean functions and expressions that we have discovered also apply to other mathematical structures propositions and sets and the operations defined on them. Boolean algebra, then we know that all results

established about Boolean algebras apply to this

For this purpose, we need an abstract definition of a Boolean algebra. CMSC 203 - Discrete Structures

Fall 2002

A Boolean algebra is a set B with two , elements 0 and 1, and a binary operations v unary operation – such that the following properties hold for all x, y, and z in B: X∨ 1 = x (identity laws) X ∨ (-x) = 0 (domination laws) z) and (x ∨ Z = X ∨ z) and (associative laws) (X ∧ Z = X ^ x and $x \land$ x (commutative laws) $X \, \lor \,$ $X \, \lor \,$ Z) = (X ∨ (x ∨ $x \land (y \lor z) = (x \land y) \lor (x \land z)$ Fall 2002 357











