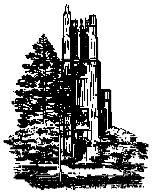
Wednesday May 25, 2016
Lecture 07

Proofs, Set Theory



Notables

- Homework#4
  - Page 78, Problem 4
  - Page 80, Problem 20, and 24
  - Page 91, Problem 6, and 8
  - Due Thursday May 26, 2016
- Read Chapter 2
- Tentative Schedule for the week

Week	M	T	W	R	Topic	Section
1	5-23				Nested Quantifiers, Rules of Inference	1.5, 1.6
		5-23			Proof	1.7
			5-25		Proofs, Sets	1.8
				5-26	Sets	2.1

Prove that $\forall x\,(A(x) \vee B(x)) \Rightarrow \forall x\, A(x) \vee \exists x\, B(x)$

1.	$\neg(\forall x\, A(x) \vee \exists x\, B(x))$	Contrary Assumption
2.	$\neg\forall x\, A(x) \wedge \neg\exists x\, B(x)$	1 & De Morgan's
3.	$\neg\forall x\, A(x)$	2
4.	$\exists x\,\neg A(x)$	3 & De Morgan's
5.	$\neg\exists x\, B(x)$	2
6.	$\forall x\,\neg B(x)$	5 & De Morgan's
7.	$\neg A(y)$	4, ES, fixed *y*
8.	$\neg B(y)$	6, US, free *y*, choose as in 7
9.	$\neg A(y) \wedge \neg B(y)$	7 & 8
10.	$\neg(A(y) \vee B(y))$	9, De Morgan's
11.	$\forall x\,(A(x) \vee B(x))$	Premise
12.	$A(y) \vee B(y)$	11, US, any *y*, same as in 9
13.	Contradiction	10 & 12

Proofs, and Proof Methods, Summary & Recap

- What is a *logical argument*?
  - *Logical Implication* (⇒)
- When is a mathematical argument correct?
  - Need *rules of inference*

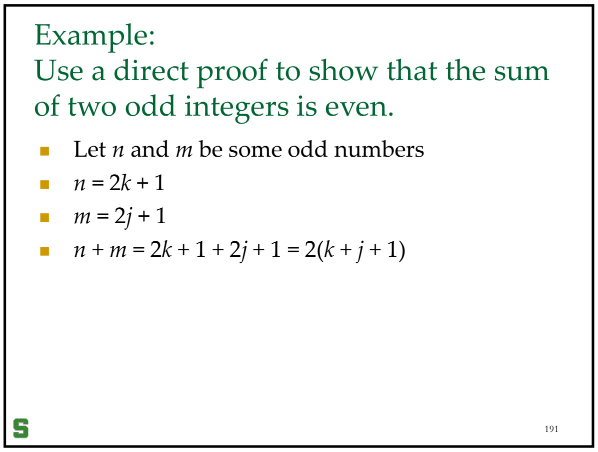
Proof Methods (Does $p \Rightarrow c$?)

		Direct	*Contrapositive (indirect)*	*Contradiction*
p	*c*	$p \Rightarrow c$	$\neg c \Rightarrow \neg p$	$p \wedge \neg c \Rightarrow$ F
T	T	T	T	F
T	F	F	F	T
F	T	T	T	F
F	F	T	T	F

Let's prove 2 = 1 ☺

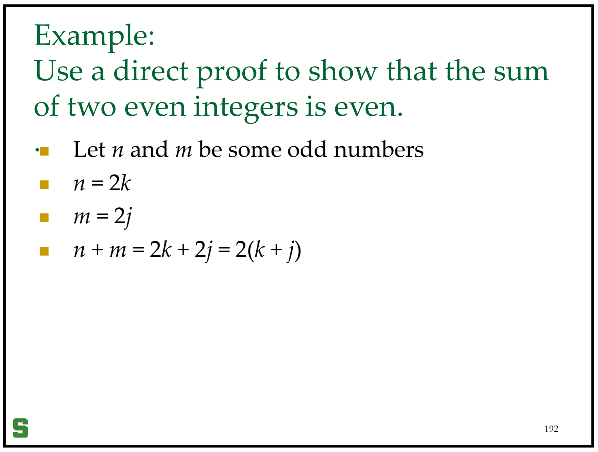
Proof: We use these steps, where *a* and *b* are two equal positive integers.

	Step	Reason
1.	$a = b$	Given
2.	$a^2 = ab$	Multiply both sides of (1) by *a*
3.	$a^2 - b^2 = ab - b^2$	Subtract b^2 from both sides of (2)
4.	$(a - b)(a + b) = b(a - b)$	Factor both sides of (3)
5.	$a + b = b$	Divide both sides of (4) by $a - b$
6.	$2b = b$	Replace *a* by *b* in (5) because $a = b$ and simplify
7.	$2 = 1$	Divide both sides of (6) by *b*



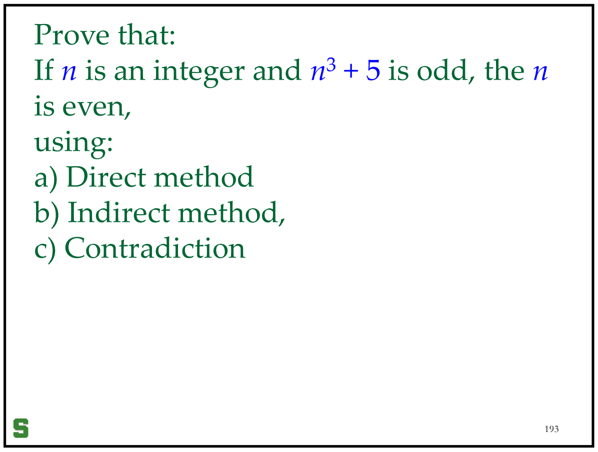
Example:
Use a direct proof to show that the sum of two odd integers is even.

- Let n and m be some odd numbers
- $n = 2k + 1$
- $m = 2j + 1$
- $n + m = 2k + 1 + 2j + 1 = 2(k + j + 1)$

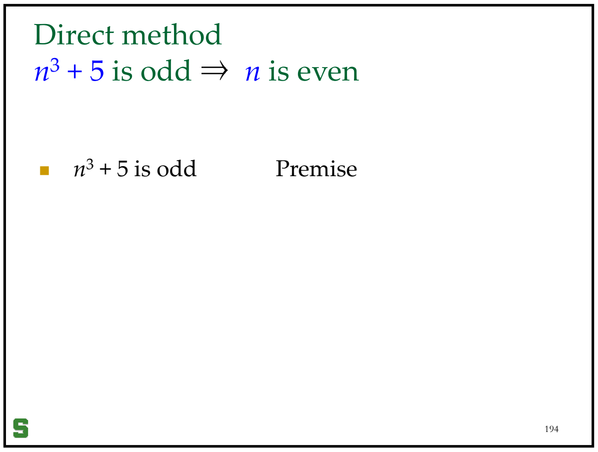


Example:
Use a direct proof to show that the sum of two even integers is even.

- Let n and m be some odd numbers
- $n = 2k$
- $m = 2j$
- $n + m = 2k + 2j = 2(k + j)$

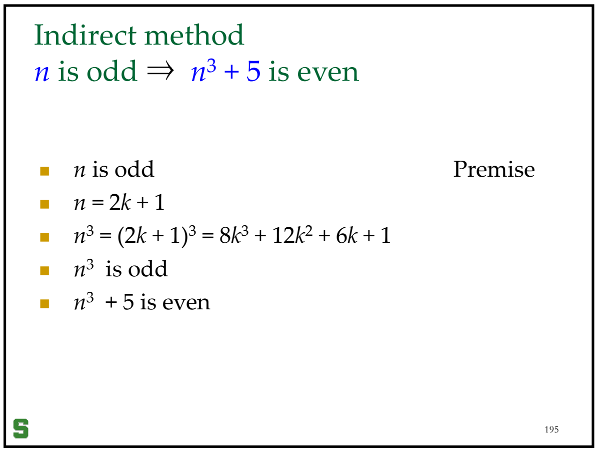


Prove that:
If n is an integer and $n^3 + 5$ is odd, the n is even,
using:
a) Direct method
b) Indirect method,
c) Contradiction



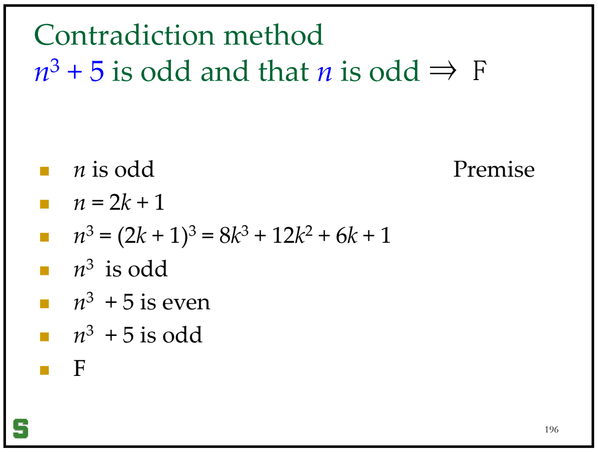
Direct method
$n^3 + 5$ is odd $\Rightarrow$ n is even

- $n^3 + 5$ is odd Premise



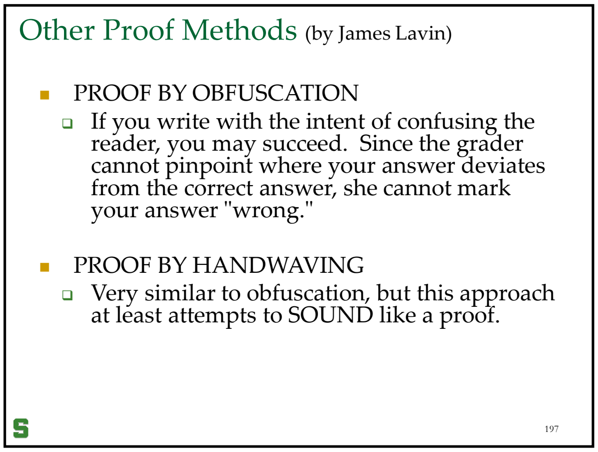
Indirect method
n is odd $\Rightarrow$ $n^3 + 5$ is even

- n is odd Premise
- $n = 2k + 1$
- $n^3 = (2k + 1)^3 = 8k^3 + 12k^2 + 6k + 1$
- n^3 is odd
- $n^3 + 5$ is even



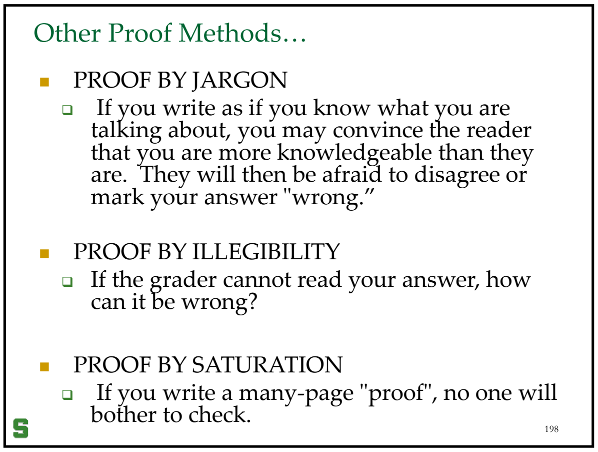
Contradiction method
$n^3 + 5$ is odd and that n is odd $\Rightarrow$ F

- n is odd Premise
- $n = 2k + 1$
- $n^3 = (2k + 1)^3 = 8k^3 + 12k^2 + 6k + 1$
- n^3 is odd
- $n^3 + 5$ is even
- $n^3 + 5$ is odd
- F



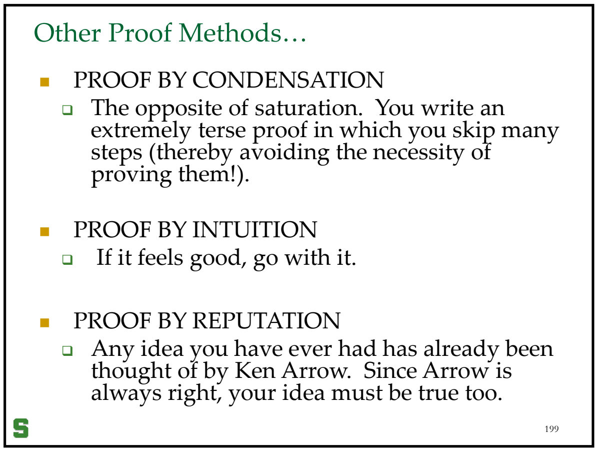
Other Proof Methods (by James Lavin)

- PROOF BY OBFUSCATION
  - If you write with the intent of confusing the reader, you may succeed. Since the grader cannot pinpoint where your answer deviates from the correct answer, she cannot mark your answer "wrong."
- PROOF BY HANDWAVING
  - Very similar to obfuscation, but this approach at least attempts to SOUND like a proof.



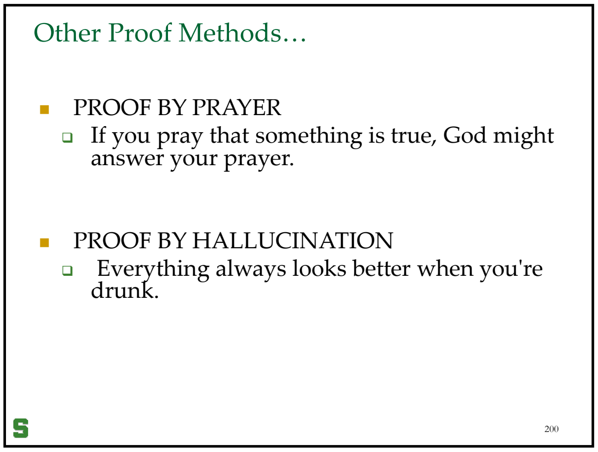
Other Proof Methods…

- PROOF BY JARGON
  - If you write as if you know what you are talking about, you may convince the reader that you are more knowledgeable than they are. They will then be afraid to disagree or mark your answer "wrong."
- PROOF BY ILLEGIBILITY
  - If the grader cannot read your answer, how can it be wrong?
- PROOF BY SATURATION
  - If you write a many-page "proof", no one will bother to check.



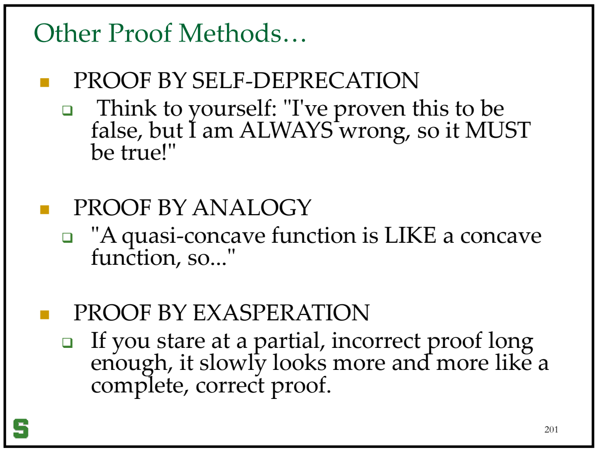
Other Proof Methods…

- PROOF BY CONDENSATION
  - The opposite of saturation. You write an extremely terse proof in which you skip many steps (thereby avoiding the necessity of proving them!).
- PROOF BY INTUITION
  - If it feels good, go with it.
- PROOF BY REPUTATION
  - Any idea you have ever had has already been thought of by Ken Arrow. Since Arrow is always right, your idea must be true too.



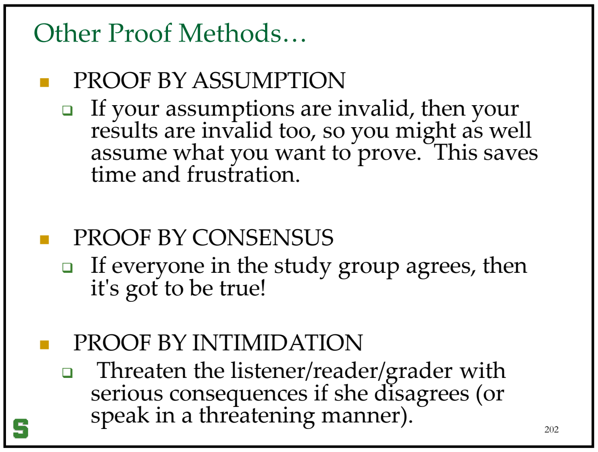
Other Proof Methods…

- PROOF BY PRAYER
  - If you pray that something is true, God might answer your prayer.
- PROOF BY HALLUCINATION
  - Everything always looks better when you're drunk.



Other Proof Methods…

- PROOF BY SELF-DEPRECATION
  - Think to yourself: "I've proven this to be false, but I am ALWAYS wrong, so it MUST be true!"
- PROOF BY ANALOGY
  - "A quasi-concave function is LIKE a concave function, so..."
- PROOF BY EXASPERATION
  - If you stare at a partial, incorrect proof long enough, it slowly looks more and more like a complete, correct proof.



Other Proof Methods…

- PROOF BY ASSUMPTION
  - If your assumptions are invalid, then your results are invalid too, so you might as well assume what you want to prove. This saves time and frustration.
- PROOF BY CONSENSUS
  - If everyone in the study group agrees, then it's got to be true!
- PROOF BY INTIMIDATION
  - Threaten the listener/reader/grader with serious consequences if she disagrees (or speak in a threatening manner).

Other Proof Methods…

- PROOF BY TRIVIALIZATION
  - How many times have you read in a textbook, "It is obvious that..." or "It can easily be shown that..." or, "The attentive reader will understand that..." It is no coincidence that these propositions are always made about the most difficult concepts in the book. The author could not prove them, so they trivialized them. It's a very effective technique with widespread application and makes you look very, very smart in the process!

Other Proof Methods…

- PROOF BY SUPPOSITION
  - Suppose that the proposition you are trying to prove is true. Try to show a contradiction. If you cannot, then the proposition must be true!
- PROOF BY UPPER-YEAR STUDENTS
  - Consult your favorite upper-year student who has been through the course already.

Other Proof Methods…

- PROOF BY LAST YEAR'S ANSWER SET
  - Superior to consulting upper-year students, but useful only if they have bothered to save their notes. The more confusing the class, the more likely they are to have burned their notes.
- PROOF BY PROOF
  - Just find the result in a book somewhere and copy it.

Other Proof Methods…Finally

- PROOF BY PARENTHESES:
  - Let's say that you have reduced the proof to solving the equation

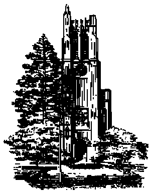
    $$3 + 14 + 1 = 20.$$

    It may seem that you are stuck, but if you are observant, you will notice that adding parentheses will solve the problem:

    $$(3 + 1)(4+1) = 20$$

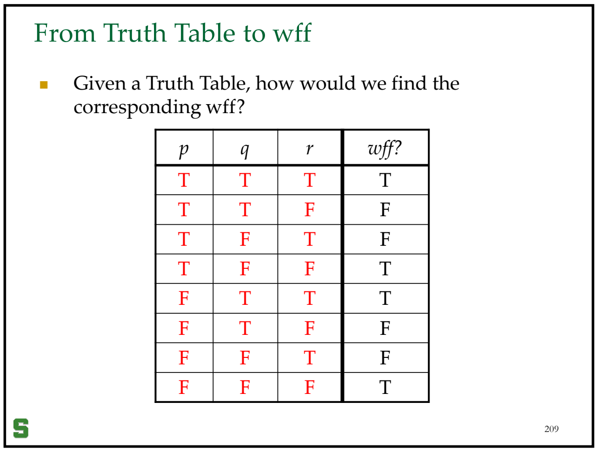
Applications of Logic and Formal Proofs

Supplemental Material



Applications of Logic

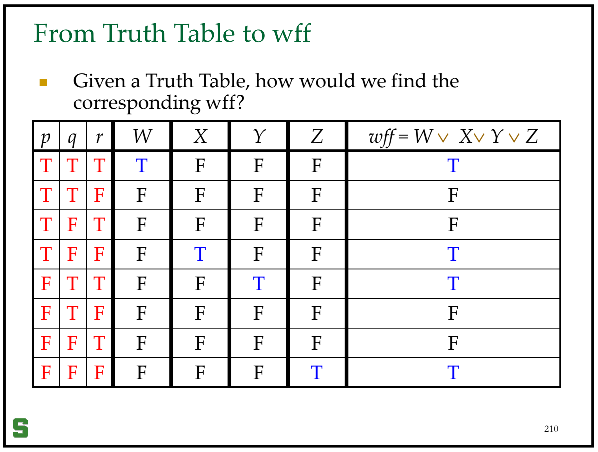
- Day-to-day conversation
- The equivalent Algebra (Boolean) for circuit design
- Program Correctness
- Complexity Theory



From Truth Table to wff

- Given a Truth Table, how would we find the corresponding wff?

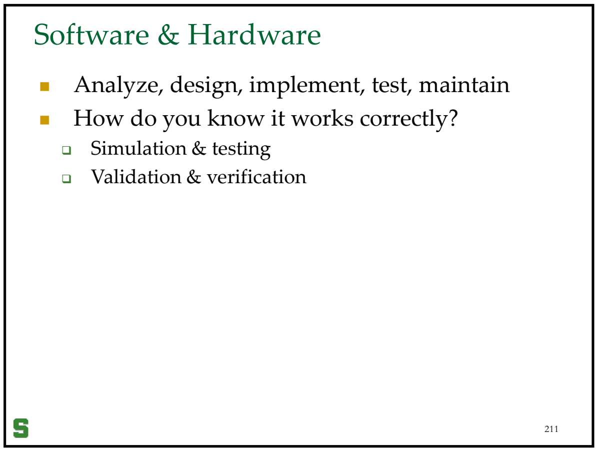
p	*q*	*r*	*wff?*
T	T	T	T
T	T	F	F
T	F	T	F
T	F	F	T
F	T	T	T
F	T	F	F
F	F	T	F
F	F	F	T



From Truth Table to wff

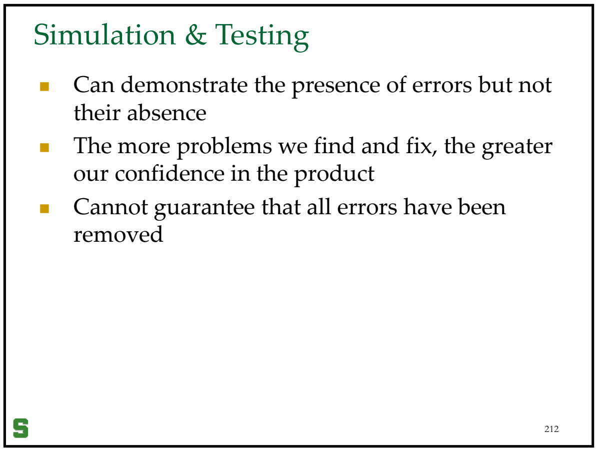
- Given a Truth Table, how would we find the corresponding wff?

p	*q*	*r*	*W*	*X*	*Y*	*Z*	*wff* = *W* ∨ *X*∨ *Y* ∨ *Z*
T	T	T	T	F	F	F	T
T	T	F	F	F	F	F	F
T	F	T	F	F	F	F	F
T	F	F	F	T	F	F	T
F	T	T	F	F	T	F	T
F	T	F	F	F	F	F	F
F	F	T	F	F	F	F	F
F	F	F	F	F	F	T	T



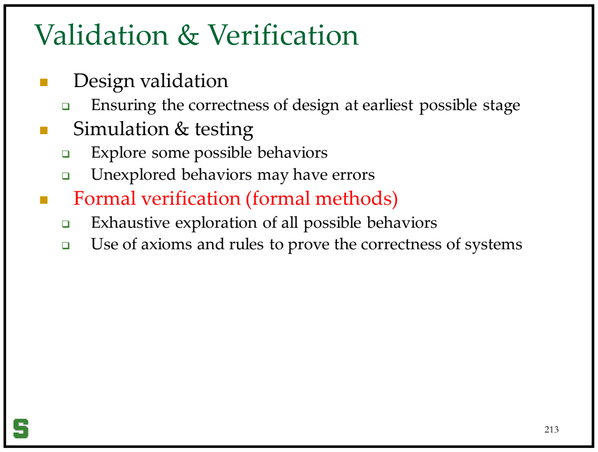
Software & Hardware

- Analyze, design, implement, test, maintain
- How do you know it works correctly?
  - Simulation & testing
  - Validation & verification



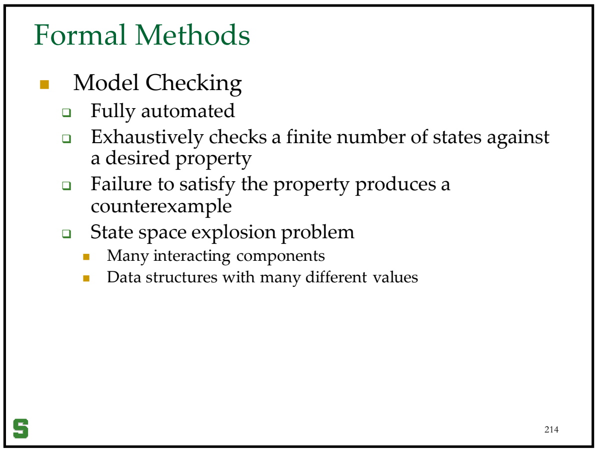
Simulation & Testing

- Can demonstrate the presence of errors but not their absence
- The more problems we find and fix, the greater our confidence in the product
- Cannot guarantee that all errors have been removed



Validation & Verification

- Design validation
  - Ensuring the correctness of design at earliest possible stage
- Simulation & testing
  - Explore some possible behaviors
  - Unexplored behaviors may have errors
- Formal verification (formal methods)
  - Exhaustive exploration of all possible behaviors
  - Use of axioms and rules to prove the correctness of systems



Formal Methods

- Model Checking
  - Fully automated
  - Exhaustively checks a finite number of states against a desired property
  - Failure to satisfy the property produces a counterexample
  - State space explosion problem
    - Many interacting components
    - Data structures with many different values

Model Checking Usage

- Hardware
  - Regularity in state space
  - Hardware Controllers
  - Digital Circuits
  - Communication Protocols
- Software
  - Less structured than hardware, leading to very large state explosion

Formal Methods

- Theorem Proving / Deductive Verification
  - Can reason about infinite state systems
  - Tools can enforce correct use of axioms and rules, and perform searches to suggest ways to make progress
  - In general, process cannot be fully automated

Critical Systems

- E-commerce
  - Is it secure?
- Communications networks
  - Is it reliable?
  - Telecommunications generally at the forefront of FM research
- Air traffic control systems
  - Is it safe?

Cost of Failure

- Loss of Money / Reputation
  - Intel Pentium floating point unit (1994)
  - AT&T 9-hr outage (Jan 15, 1990)
  - Mars Climate Orbiter crash (1999)
- Loss of Life
  - Therac-25 (1985-87)

Intel

- **TOP TEN NEW INTEL SLOGANS FOR THE PENTIUM**

  9.9999973251 It's a FLAW, not a Bug
  8.9999163362 It's Close Enough, We Say So
  7.9999414610 Nearly 300 Correct Opcodes
  6.9999831538 You Don't Need to Know What's Inside
  5.9999835137 Redefining the PC--and Mathematics As Well
  4.9999999021 We Fixed It, Really
  3.9998245917 Division Considered Harmful
  2.9991523619 Why Do You Think They Call It *Floating* Point?
  1.9999103517 We're Looking for a Few Good Flaws
  0.9999999998 The Errata Inside

Intel

- Error in FPU: High-precision calculations in some cases were only accurate to the 8th decimal place; in some cases only to the 1st or 4th decimal place
- Rare condition, not caught with testing prior to deployment
- *December 21, 1994: The Wall Street Journal, pg B1* **INTEL ANNOUNCED IT WILL REPLACE ALL OF ITS FLAWED PENTIUM CHIPS - IT IS THE FIRST CONSUMER RECALL OF A COMPUTER CHIP.**
- Source: http://davefaq.com/Opinions/Stupid/Pentium.html

AT&T

- 114 switching nodes went down for 9 hours
  - An estimated 60 thousand people were left without telephone service, and 70 million phone calls went uncompleted. AT&T estimates at least $60 million in lost revenue and damage to its reputation.
- Bug in the failure recovery code of the switches:
  - When a node crashed, it sent an "out of service" message to the neighboring nodes, which are supposed to re-route traffic around it. However, the bug (a misplaced "break" proposition in C code) caused the neighboring nodes to crash themselves upon receiving the message, and further propagate the fault by passing the message to nodes further out in the network.
- Source: http://www.cs.berkeley.edu/~nikitab/courses/cs294-8/hw1.html
- Esterel, SDL, Z, Promela, temporal logic

NASA

- Navigation and stabilizing thrusters
  - Spacecraft returning metric data
  - Earthside engineering sending data using English units
  - Difference 1.3 meters / second
- $125 million spacecraft crashed into atmosphere of Mars and was lost
- Source: http://clive.canoe.ca/CNEWSHeyMartha9911/10_metric.html
- SCR

THERAC-25

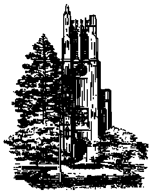
- Medical linear accelerator
  - Accelerated electrons for shallow tissue
  - X-ray photons for deep tissue
- Prior versions of the machine
  - Capable of standing alone
  - Industry-standard hardware safety features and interlocks
- Therac-25 to take advantage of computer control from the outset
  - Interactions between different components
  - Edits performed quickly are not registered
- (1985-1987) 6 massive (100x) overdoses, at least two deaths
- Source: http://courses.cs.vt.edu/~cs3604/lib/Therac_25/Therac_1.html

Problem Complexity

- Types of Problems
  - Easy
  - Seemingly easy, but time consuming
  - Difficult problems
  - Unsolvable problems
- The Satisfiability Problem:
  - Is there a truth assignment that would make the following wff to be True

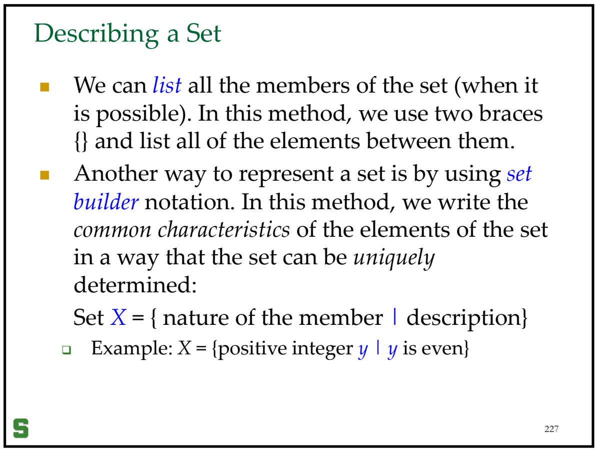
$$(p \vee q \vee \neg r) \wedge (p \vee \neg q \vee \neg s) \wedge (p \vee \neg r \vee \neg s) \wedge (\neg p \vee \neg q \vee \neg s) \wedge (p \vee q \vee \neg s)$$

Set Theory



What is a *Set*?

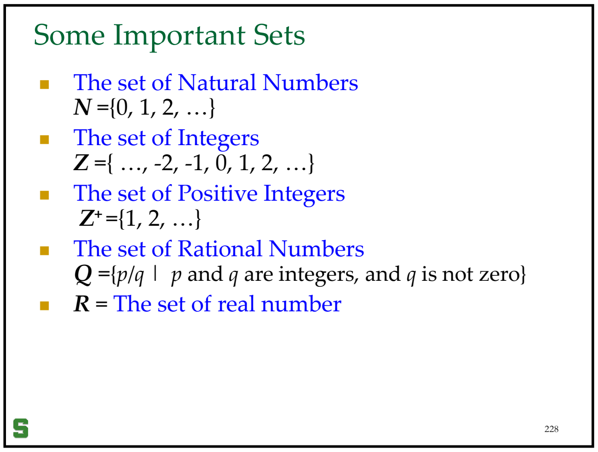
- Definition: A *set* is an *unordered* collection of objects, called the *elements* or *members* of the set. A set is called to *contain* its elements.
- Note that the definition of a set does not require any relationship among the members of a set.
- In a set, repeated elements are ignored.
- To indicate the fact that:
  - x is an element of the set S, we write: $x \in S$
  - x is not an element of the set S, we write: $x \notin S$



Describing a Set

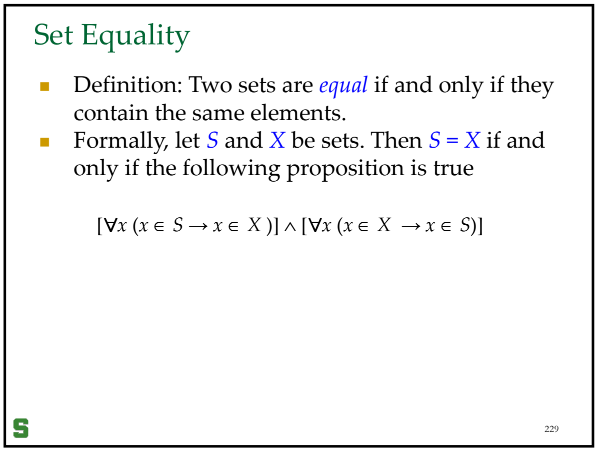
- We can *list* all the members of the set (when it is possible). In this method, we use two braces {} and list all of the elements between them.
- Another way to represent a set is by using *set builder* notation. In this method, we write the *common characteristics* of the elements of the set in a way that the set can be *uniquely* determined:

  Set X = { nature of the member | description}
  - Example: X = {positive integer y | y is even}



Some Important Sets

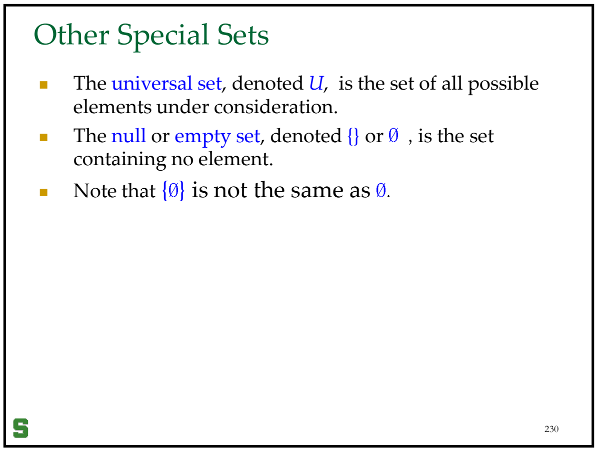
- The set of Natural Numbers
  $\boldsymbol{N}$ ={0, 1, 2, …}
- The set of Integers
  $\boldsymbol{Z}$ ={ …, -2, -1, 0, 1, 2, …}
- The set of Positive Integers
  $\boldsymbol{Z}^{+}$={1, 2, …}
- The set of Rational Numbers
  $\boldsymbol{Q}$ ={p/q | p and q are integers, and q is not zero}
- $\boldsymbol{R}$ = The set of real number



Set Equality

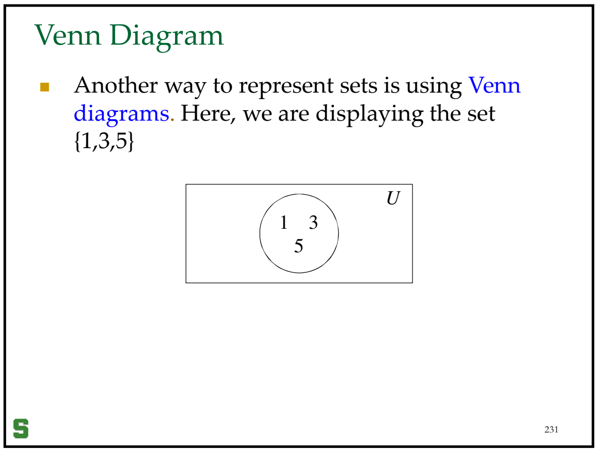
- Definition: Two sets are *equal* if and only if they contain the same elements.
- Formally, let S and X be sets. Then $S = X$ if and only if the following proposition is true

$$[\forall x\ (x \in S \rightarrow x \in X\ )] \wedge [\forall x\ (x \in X\ \rightarrow x \in S)]$$



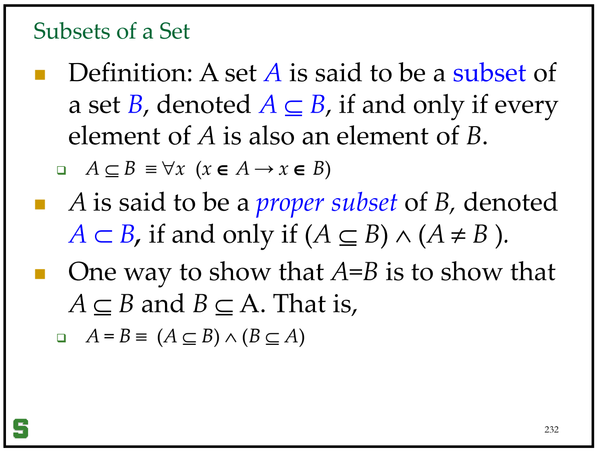
Other Special Sets

- The universal set, denoted U, is the set of all possible elements under consideration.
- The null or empty set, denoted {} or ∅ , is the set containing no element.
- Note that {∅} is not the same as ∅.



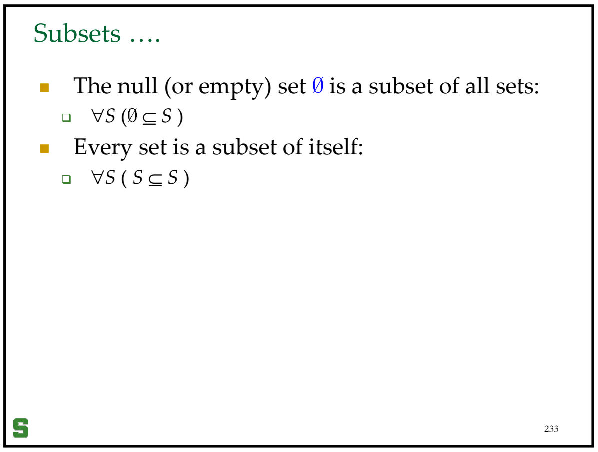
Venn Diagram

- Another way to represent sets is using Venn diagrams. Here, we are displaying the set {1,3,5}



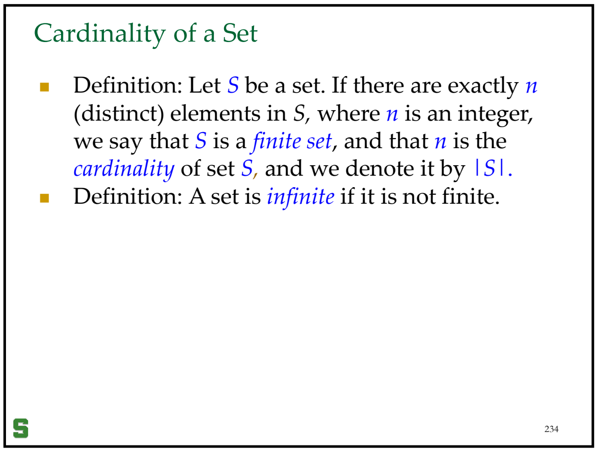
Subsets of a Set

- Definition: A set A is said to be a subset of a set B, denoted $A \subseteq B$, if and only if every element of A is also an element of B.
  - $A \subseteq B\ \equiv \forall x\ \ (x \in A \rightarrow x \in B)$
- A is said to be a *proper subset* of B, denoted $A \subset B$, if and only if $(A \subseteq B) \wedge (A \neq B\ )$.
- One way to show that A=B is to show that $A \subseteq B$ and $B \subseteq$ A. That is,
  - $A = B \equiv\ (A \subseteq B) \wedge (B \subseteq A)$



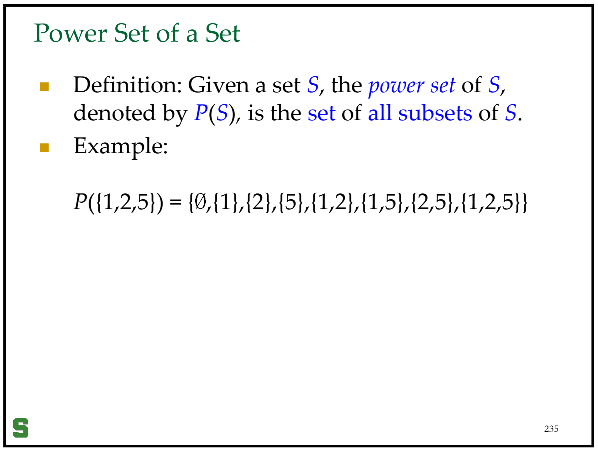
Subsets ….

- The null (or empty) set $\emptyset$ is a subset of all sets:
  - $\forall S \, (\emptyset \subseteq S)$
- Every set is a subset of itself:
  - $\forall S \, (S \subseteq S)$



Cardinality of a Set

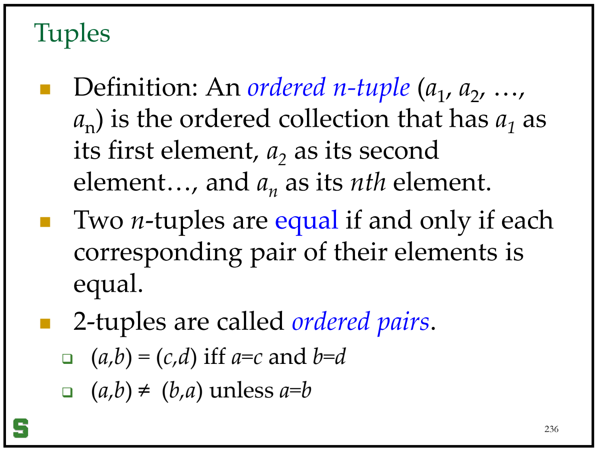
- Definition: Let *S* be a set. If there are exactly *n* (distinct) elements in *S*, where *n* is an integer, we say that *S* is a *finite set*, and that *n* is the *cardinality* of set *S*, and we denote it by $|S|$.
- Definition: A set is *infinite* if it is not finite.



Power Set of a Set

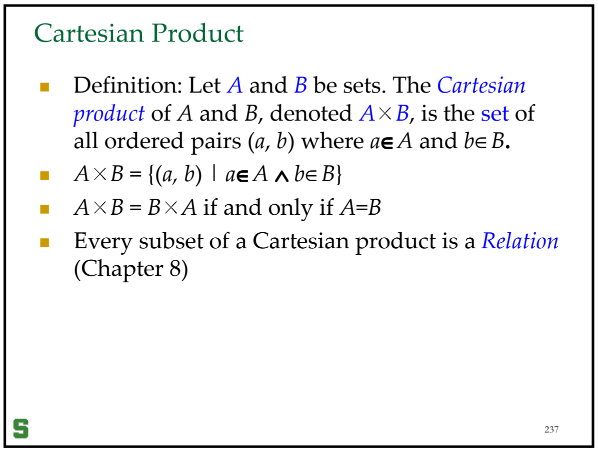
- Definition: Given a set *S*, the *power set* of *S*, denoted by *P*(*S*), is the set of all subsets of *S*.
- Example:

$$P(\{1,2,5\}) = \{\emptyset,\{1\},\{2\},\{5\},\{1,2\},\{1,5\},\{2,5\},\{1,2,5\}\}$$



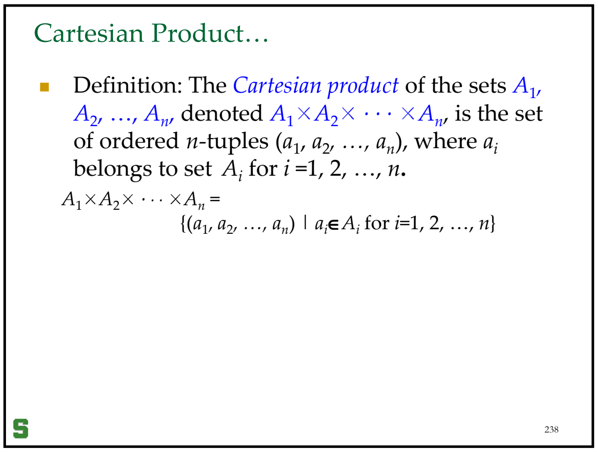
Tuples

- Definition: An *ordered n-tuple* $(a_1, a_2, \ldots, a_n)$ is the ordered collection that has a_1 as its first element, a_2 as its second element…, and a_n as its *nth* element.
- Two *n*-tuples are equal if and only if each corresponding pair of their elements is equal.
- 2-tuples are called *ordered pairs*.
  - $(a,b) = (c,d)$ iff $a=c$ and $b=d$
  - $(a,b) \neq (b,a)$ unless $a=b$



Cartesian Product

- Definition: Let *A* and *B* be sets. The *Cartesian product* of *A* and *B*, denoted $A \times B$, is the set of all ordered pairs (a, b) where $a \in A$ and $b \in B$.
- $A \times B = \{(a, b) \mid a \in A \wedge b \in B\}$
- $A \times B = B \times A$ if and only if $A=B$
- Every subset of a Cartesian product is a *Relation* (Chapter 8)



Cartesian Product…

- Definition: The *Cartesian product* of the sets $A_1, A_2, \ldots, A_n$, denoted $A_1 \times A_2 \times \cdots \times A_n$, is the set of ordered *n*-tuples $(a_1, a_2, \ldots, a_n)$, where a_i belongs to set A_i for $i = 1, 2, \ldots, n$.

$$A_1 \times A_2 \times \cdots \times A_n = \{(a_1, a_2, \ldots, a_n) \mid a_i \in A_i \text{ for } i=1, 2, \ldots, n\}$$