# EE360T/382C-16 Software Testing

khurshid@ece.utexas.edu

Lecture 2

# Today and next time

Discrete math basics

- Review some material from MIT's 6.042 text
  - https://courses.csail.mit.edu/6.042/spring17/mcs.pdf
- Focus on Section I Proofs in the text
  - **Propositions, predicates**
  - **Logical formulas**
  - Mathematical data types
  - Induction
  - State machines

# Propositions

A proposition is a statement that is either true or false

[Prop. 1.1.1] $2 + 3 = 5$

[Prop. 1.1.2] $1 + 1 = 3$

Can you think of a sentence that is not a proposition?

# Propositions

[Claim 1.1.3] For every non-negative integer $n$ the value of $n^2 + n + 41$ is prime

Let $p(n) ::= n^2 + n + 41$

Example values: $p(0) = 41$, $p(1) = 43$; $p(2) = 47$; $p(3) = 53$, …, $p(20) = 461$ are all prime

But p(40) = 40.40 + 40 + 41 = 41.41 is not a prime

# Propositions

[Euler's conjecture, 1769] $a^4 + b^4 + c^4 = d^4$ has no solution when a, b, c, d are positive integers

Proved false 218 years later [Elkies]

- A = 95800, b = 217519, c = 414560, d = 422481

# Propositions

[Fermat's last theorem, 1630] There are no positive integers x, y, and z such that $x^n + y^n = z^n$ for some integer $n > 2$

Fermat claimed to have a proof but not enough space to fit it in a margin

Over the years, shown to hold for all $n \le 4{,}000{,}000$

In 1994, Andrew Wiles gave a proof after working on it for 7 years

# Propositions

[Goldbach conjecture, 1742] Every even integer greater than 2 is a sum of 2 primes

Known to hold for all numbers up to $10^{18}$

But we do not know if it true or false

# Predicates

A predicate is a proposition whose truth depends on the value of one or more variables

P(n) ::= “n is a perfect square”

- Truth depends on the value of n

P(4) is true but P(5) is false

A predicate is analogous to a function

- A predicate is a boolean function

# Logical formulas

Natural language sentences can have ambiguity

- “You may have cake, or you may have ice cream.”
- “If you can solve any problem we come up with, then you get an A for the course.”

# Boolean operators

| P | NOT(P) |
|---|---|
| True | False |
| False | True |

| P | Q | P AND Q | P OR Q |
|---|---|---|---|
| T | T | T | T |
| T | F | F | T |
| F | T | F | T |
| F | F | F | F |

# Boolean operators

| P | Q | P if and only iff Q |
|---|---|---|
| T | T | T |
| T | F | F |
| F | T | F |
| F | F | T |

P and Q have the same truth value

Ex: for any real number x, $x^2 - 4 >= 0$ IFF $|x| >= 2$

# Boolean operators

| P | Q | P IMPLIES Q |
|---|---|---|
| T | T | T |
| T | F | F |
| F | T | T |
| F | F | T |

Is the following propostion true or false:

- If Goldbach's conjecture is true, then $x^2$ >= 0 for every real number x

# Propositional logic in code

Example Java statement:

```
if (x > 0 || (x <= 0 && y > 100)) …
```

Let A be the expression “x > 0” and B be “y > 100”

The condition is “A OR (NOT(A) AND B)”

# Propositional logic in code

| A | B | A OR (NOT(A) AND B) | | | A OR B |
|---|---|---|---|---|---|
| T | T | **T** | F | F | **T** |
| T | F | **T** | F | F | **T** |
| F | T | **T** | T | T | **T** |
| F | F | **F** | T | F | **F** |

"A OR (NOT(A) AND B)" is equivalent to "A OR B"

- Simpler, easier to comprehend
- Can be used to simplify the original program and possibly make it run faster

# Notation

| **English** | **Symbolic Notation** |
|---|---|
| $\text{NOT}(P)$ | $\neg P$ (alternatively, $\overline{P}$) |
| $P$ AND $Q$ | $P \wedge Q$ |
| $P$ OR $Q$ | $P \vee Q$ |
| $P$ IMPLIES $Q$ | $P \longrightarrow Q$ |
| if $P$ then $Q$ | $P \longrightarrow Q$ |
| $P$ IFF $Q$ | $P \longleftrightarrow Q$ |
| $P$ XOR $Q$ | $P \oplus Q$ |

source: page 54 of https://courses.csail.mit.edu/6.042/spring17/mcs.pdf

# Equivalence

Do the following two sentences say the same thing?

- If I am hungry, then I am grumpy (S1)
- If I am not grumpy, then I am not hungry (S2)

Let P be "I am hungry" and Q be "I am grumpy"

- S1 is P IMPLIES Q
- S2 is NOT(Q) IMPLIES NOT(P)

# Equivalence

| P | Q | P IMPLIES Q | NOT(Q) IMPLIES NOT(P) | | |
|---|---|---|---|---|---|
| T | T | **T** | F | **T** | F |
| T | F | **F** | T | **F** | F |
| F | T | **T** | F | **T** | T |
| F | F | **T** | T | **T** | T |

NOT(Q) IMPLIES NOT(P) is called the contrapositive of P IMPLIES Q

- An implication and its contrpositive are always equivalent

# Validity

A formula is valid if it is always true regardless of the values of its variables

Ex: P or not(P)

Ex: (P implies Q and Q implies R) implies (P implies R)

# Satisfiability

A formula is satisfiable if there is some assignment of values to its variables such that the formula is true

Ex: P and Q is satisfiable because for P = T and Q = T, P and Q = T

P is satisfiable if and only if its negation not(P) is not valid

# The SAT problem

Is the given formula satisfiable?

- (p || q || r) && (!p || !q) && (!p || !r) && (!r || !q)

Can construct a truth table to check satisfiability

- Size of table grows exponentially

Unknown whether there a polynomial-time solution

- “P versus NP” problem

# Quantifiers

Universal – for all:

- Ex: $\forall x \in \mathbb{R} . x^2 \geqslant 0$

Existential – there exists:

- Ex: $\exists x \in \mathbb{R} . 5x^2 - 7 = 0$

$\forall x \in \mathbb{R} . 5x^2 - 7 = 0$ is false

# Mixing quantifiers

Recall Goldbach's conjecture: Every even integer greater than 2 is a sum of 2 primes

Let *Evens* be the set of all evens > 2 and *Primes* be the set of all primes

$$\forall n \in Evens \, \exists p \in Primes \, \exists q \in Primes \, . \, n = p + q$$

# Order of quantifiers

Swapping the order of different types of quantifiers usually changes the meaning of the formula

Ex: the following is false:

$$\exists p \in Primes\, \exists q \in Primes\, \forall n \in Evens \,.\, n = p + q$$

# Negating quantifiers

Ex: the following sentences mean the same thing

- Not everyone likes ice cream
- There is someone who does not like ice cream

In general

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

# ?/!