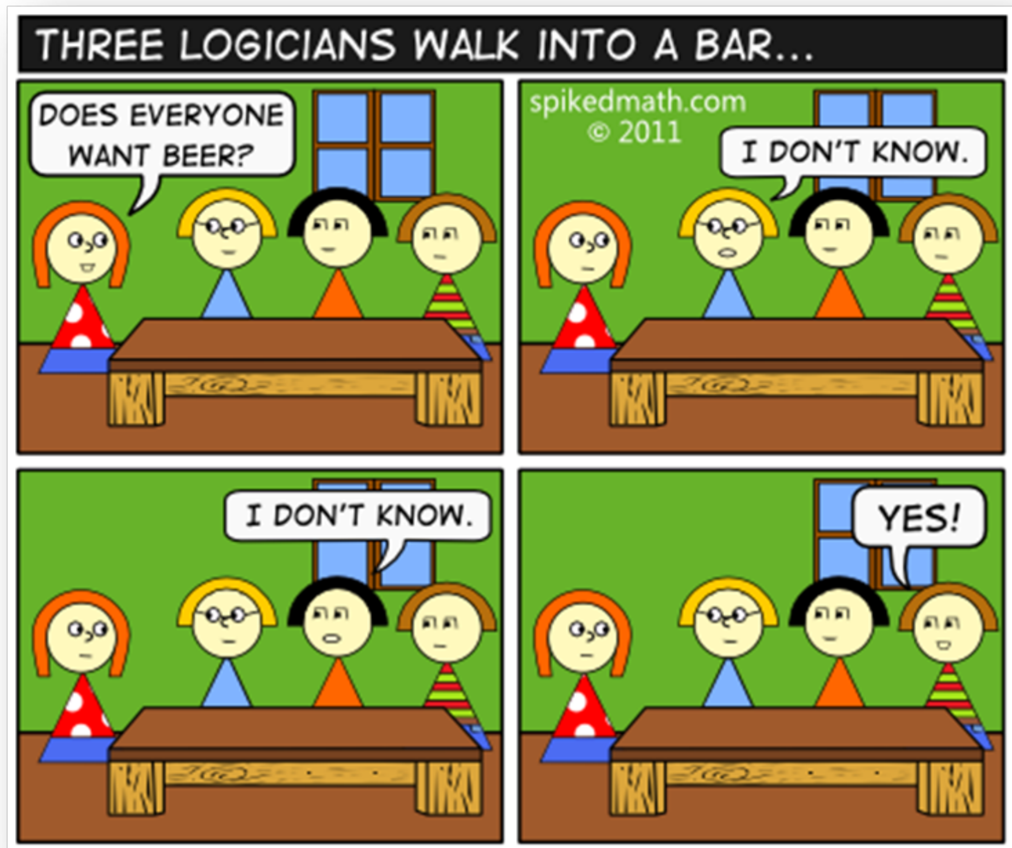
CSE 311: Foundations of Computing

Lecture 6: More Predicate Logic



Last class

Canonical Forms

- sum-of-products and product-of-sums
- both are useful

Corollaries of construction:

- any function can be formed with just $\neg$, $\vee$, $\wedge$
- actually, just $\neg$, $\vee$ (De Morgan's laws)
- actually, just A (HW1 Q4)

  NAND and NOR also have this property

Last class: Predicates

Predicate

– **A function that returns a truth value, e.g.,**

Cat(x) ::= "x is a cat"

Prime(x) ::= "x is prime"

HasTaken(x, y) ::= "student x has taken course y"

LessThan(x, y) ::= "$x < y$"

Sum(x, y, z) ::= "$x + y = z$"

GreaterThan5(x) ::= "$x > 5$"

HasNChars(s, n) ::= "string s has length n"

Predicates can have varying numbers of arguments and input types.

Last class: Domain of Discourse

For ease of use, we define one "type"/"domain" that we work over. This non-empty set of objects is called the "**domain of discourse**".

For each of the following, what might the domain be?

(1) "x is a cat", "x barks", "x ruined my couch"

"mammals" or "sentient beings" or "cats and dogs" or …

(2) "x is prime", "x = 0", "x < 0", "x is a power of two"

"numbers" or "integers" or "integers greater than 5" or …

(3) "x is a pre-req for z"

"courses"

Quantifiers

We use ***quantifiers*** to talk about collections of objects.

$\forall x\ P(x)$

$P(x)$ **is true for every** x **in the domain**

read as "for all x, P of x"

$\exists x\ P(x)$

There is an x **in the domain for which** $P(x)$ **is true**

read as "there exists x, P of x"

Statements with Quantifiers

Domain of Discourse
Positive Integers

Predicate Definitions	
Even(x) ::= "x is even"	Greater(x, y) ::= "x > y"
Odd(x) ::= "x is odd"	Equal(x, y) ::= "x = y"
Prime(x) ::= "x is prime"	Sum(x, y, z) ::= "x + y = z"

Determine the truth values of each of these statements:

$\exists x$ Even(x)	**T**	**e.g. 2, 4, 6, ...**
$\forall x$ Odd(x)	**F**	**e.g. 2, 4, 6, ...**
$\forall x$ (Even(x) $\vee$ Odd(x))	**T**	**every integer is either even or odd**
$\exists x$ (Even(x) $\wedge$ Odd(x))	**F**	**no integer is both even and odd**
$\forall x$ Greater(x+1, x)	**T**	**adding 1 makes a bigger number**
$\exists x$ (Even(x) $\wedge$ Prime(x))	**T**	**Even(2) is true and Prime(2) is true**

Statements with Quantifiers (Literal Translations)

Domain of Discourse
Positive Integers

Predicate Definitions	
Even(x) ::= "x is even"	Greater(x, y) ::= "x > y"
Odd(x) ::= "x is odd"	Equal(x, y) ::= "x = y"
Prime(x) ::= "x is prime"	Sum(x, y, z) ::= "x + y = z"

Translate the following statements to English

$\forall x\ \exists y\ \text{Greater}(y, x)$

For every positive integer x, there is a positive integer y, such that y > x.

$\exists y\ \forall x\ \text{Greater}(y, x)$

There is a positive integer y such that, for every pos. int. x, we have y > x.

$\forall x\ \exists y\ (\text{Greater}(y, x) \wedge \text{Prime}(y))$

For every positive integer x, there is a pos. int. y such that y > x and y is prime.

$\forall x\ (\text{Prime}(x) \rightarrow (\text{Equal}(x, 2) \vee \text{Odd}(x)))$

For each positive integer x, if x is prime, then x = 2 or x is odd.

$\exists x\ \exists y\ (\text{Sum}(x, 2, y) \wedge \text{Prime}(x) \wedge \text{Prime}(y))$

There exist positive integers x and y such that x + 2 = y and x and y are prime.

Statements with Quantifiers (Natural Translations)

Domain of Discourse
Positive Integers

Predicate Definitions	
Even(x) ::= "x is even"	Greater(x, y) ::= "x > y"
Odd(x) ::= "x is odd"	Equal(x, y) ::= "x = y"
Prime(x) ::= "x is prime"	Sum(x, y, z) ::= "x + y = z"

Translate the following statements to English

$\forall x\ \exists y\ \text{Greater}(y, x)$

For every positive integer, there is a larger positive integer.

$\exists y\ \forall x\ \text{Greater}(y, x)$

There is a positive integer that is larger than every other positive integer.

$\forall x\ \exists y\ (\text{Greater}(y, x) \land \text{Prime}(y))$

For every positive integer, there is a prime that is larger.

$\forall x\ (\text{Prime}(x) \rightarrow (\text{Equal}(x, 2) \lor \text{Odd}(x)))$

Every prime number is either 2 or odd.

$\exists x\ \exists y\ (\text{Prime}(x) \land \text{Prime}(y) \land \text{Sum}(x, 2, y))$

There exist prime numbers that differ by two.

English to Predicate Logic

Domain of Discourse
Mammals

Predicate Definitions
Cat(x) ::= "x is a cat"
Red(x) ::= "x is red"
LikesTofu(x) ::= "x likes tofu"

"All red cats like tofu"

$$\forall x\ ((\text{Red}(x) \land \text{Cat}(x)) \rightarrow \text{LikesTofu}(x))$$

"Some red cats don't like tofu"

$$\exists y\ ((\text{Red}(y) \land \text{Cat}(y)) \land \neg\text{LikesTofu}(y))$$

English to Predicate Logic

Domain of Discourse
Mammals

Predicate Definitions
Cat(x) ::= "x is a cat"
Red(x) ::= "x is red"
LikesTofu(x) ::= "x likes tofu"

When putting two predicates together like this, we use an "and".

"All Red cats like tofu"

When restricting to a smaller domain in a "for all" we use **implication.**

"Some red cats don't like tofu"

When restricting to a smaller domain in an "exists" we use **and.**

"Some" means "there exists".

English to Predicate Logic

Domain of Discourse
Mammals

Predicate Definitions
Cat(x) ::= "x is a cat"
Red(x) ::= "x is red"
LikesTofu(x) ::= "x likes tofu"

"Red cats like tofu"

When there's no leading quantification, it means "for all".

"A red cat doesn't like tofu"

"A" means "there exists".

Negations of Quantifiers

Predicate Definitions
PurpleFruit(x) ::= “x is a purple fruit”

(*) $\forall$x PurpleFruit(x) (“All fruits are purple”)

What is the negation of (*)?

(a) “there exists a purple fruit”

(b) “there exists a non-purple fruit”

(c) “all fruits are not purple”

Try your intuition! Which one “feels” right?

Key Idea: In every domain, exactly one of a statement and its negation should be true.

Negations of Quantifiers

Predicate Definitions
PurpleFruit(x) ::= “x is a purple fruit”

(*) $\forall$x PurpleFruit(x) (“All fruits are purple”)

What is the negation of (*)?

(a) “there exists a purple fruit”

(b) “there exists a non-purple fruit”

(c) “all fruits are not purple”

Key Idea: In every domain, exactly one of a statement and its negation should be true.

Domain of Discourse
{plum}

(*), (a)

Domain of Discourse
{apple}

(b), (c)

Domain of Discourse
{plum, apple}

(a), (b)

Negations of Quantifiers

Predicate Definitions
PurpleFruit(x) ::= "x is a purple fruit"

(*) ∀x PurpleFruit(x) ("All fruits are purple")

What is the negation of (*)?

(a) "there exists a purple fruit"
(b) "there exists a non-purple fruit"
(c) "all fruits are not purple"

Key Idea: In every domain, exactly one of a statement and its negation should be true.

Domain of Discourse
{plum}

(*), (a)

Domain of Discourse
{apple}

(b), (c)

Domain of Discourse
{plum, apple}

(a), (b)

The only choice that ensures exactly one of the statement and its negation is (b).

De Morgan's Laws for Quantifiers

$$\neg \forall x\, P(x) \equiv \exists x\, \neg P(x)$$

$$\neg \exists x\, P(x) \equiv \forall x\, \neg P(x)$$

De Morgan's Laws for Quantifiers

$$\neg \forall x\, P(x) \equiv \exists x \neg P(x)$$
$$\neg \exists x\, P(x) \equiv \forall x \neg P(x)$$

"There is no integer larger than every other integer"

$$\neg \exists x \forall y \; (x \geq y)$$
$$\equiv \forall x \neg \forall y \; (x \geq y)$$
$$\equiv \forall x \; \exists y \neg (x \geq y)$$
$$\equiv \forall x \; \exists y \; (y > x)$$

"For every integer, there is a larger integer"

Scope of Quantifiers

$\exists x \; (P(x) \land Q(x))$ **vs.** $\exists x \, P(x) \land \exists x \, Q(x)$

scope of quantifiers

$\exists x \ (P(x) \wedge Q(x))$ **vs.** $\exists x \ P(x) \wedge \exists x \ Q(x)$

This one asserts P and Q of the *same* x.

This one asserts P and Q of potentially different x's.

Variables with the same name do not necessarily refer to the same object.

Scope of Quantifiers

Example: NotLargest(x) ::= $\exists$ y Greater (y, x)

$\equiv$ $\exists$ z Greater (z, x)

truth value:

doesn't depend on y or z "bound variables"

does depend on x "free variable"

quantifiers only act on free variables of the formula they quantify

$$\forall x\, (\exists y\, (P(x,y) \rightarrow \forall x\, Q(y, x)))$$

Quantifier "Style"

$$\forall x\,(\exists y\,(P(x,y) \rightarrow \forall x\, Q(y, x)))$$

This isn't "wrong", it's just horrible style.
Don't confuse your reader by using the same variable multiple times…there are a lot of letters…

Nested Quantifiers

- **Bound variable names don't matter**

$$\forall x \, \exists y \, P(x, y) \equiv \forall a \, \exists b \, P(a, b)$$

- **Positions of quantifiers can <u>sometimes</u> change**

$$\forall x \, (Q(x) \wedge \exists y \, P(x, y)) \equiv \forall x \, \exists y \, (Q(x) \wedge P(x, y))$$

- **But: order is important...**

Quantifier Order Can Matter

Domain of Discourse
Integers OR {1, 2, 3, 4}

Predicate Definitions
GreaterEq(x, y) ::= "x ≥ y"

x \ y	1	2	3	4
1	T	F	F	F
2	T	T	F	F
3	T	T	T	F
4	T	T	T	T

"There is a number greater than or equal to all numbers."

$\exists x\ \forall y$ GreaterEq(x, y)))

Quantifier Order Can Matter

Domain of Discourse

Integers
OR
{1, 2, 3, 4}

Predicate Definitions

GreaterEq(x, y) ::= "x ≥ y"

"There is a number greater than or equal to all numbers."

$\exists x \, \forall y \, GreaterEq(x, y)))$

"Every number has a number greater than or equal to it."

$\forall y \, \exists x \, GreaterEq(x, y)))$

x \ y	1	2	3	4
1	T	F	F	F
2	T	T	F	F
3	T	T	T	F
4	T	T	T	T

Quantifier Order Can Matter

Domain of Discourse

Integers
OR
{1, 2, 3, 4}

Predicate Definitions

GreaterEq(x, y) ::= "x ≥ y"

x \ y	1	2	3	4
1	T	F	F	F
2	T	T	F	F
3	T	T	T	F
4	T	T	T	T

"There is a number greater than or equal to all numbers."

$$\exists x\ \forall y\ \text{GreaterEq}(x, y)))$$

"Every number has a number greater than or equal to it."

$$\forall y\ \exists x\ \text{GreaterEq}(x, y)))$$

The purple statement requires **an entire row** to be true.
The red statement requires one entry in **each column** to be true.

Important: both include the case x = y

Different names does not imply different objects!

Quantification with Two Variables

expression	when true	when false
$\forall x \forall y P(x, y)$	Every pair is true.	At least one pair is false.
$\exists x \exists y P(x, y)$	At least one pair is true.	All pairs are false.
$\forall x \exists y P(x, y)$	We can find a specific y for each x. $(x_1, y_1), (x_2, y_2), (x_3, y_3)$	Some x doesn't have a corresponding y.
$\exists y \forall x P(x, y)$	We can find ONE y that works no matter what x is. $(x_1, y), (x_2, y), (x_3, y)$	For any candidate y, there is an x that it doesn't work for.

Logical Inference

- So far we've considered:
  - How to understand and *express* things using propositional and predicate logic
  - How to *compute* using Boolean (propositional) logic
  - How to show that different ways of expressing or computing them are *equivalent* to each other

- Logic also has methods that let us *infer* implied properties from ones that we know
  - Equivalence is a small part of this

Applications of Logical Inference

- Software Engineering
  - Express desired properties of program as set of logical constraints
  - Use inference rules to show that program implies that those constraints are satisfied
- Artificial Intelligence
  - Automated reasoning
- Algorithm design and analysis
  - e.g., Correctness, Loop invariants.
- Logic Programming, e.g. Prolog
  - Express desired outcome as set of constraints
  - Automatically apply logic inference to derive solution