CSI 2101 / Predicate logic (§1.3-1.4):

- **Motivation**
- **Predicates**
- **Quantifiers**
  - universal quantifier, existential quantifier, other quantifiers
  - domain of a quantifier
  - binding variable by a quantifier
- **Quantified expressions**
  - equivalence of quantified expressions
  - negating quantified expressions
  - translating into/from English
  - nested quantifiers
  - order of quantifiers

Motivation

- *Predicate logic* is an extension of propositional logic that permits concisely reasoning about whole *classes* of entities.
- Propositional logic (recall) treats simple *propositions* (sentences) as atomic entities.
- In contrast, *predicate* logic distinguishes the *subject* of a sentence from its *predicate.*

Applications of Predicate Logic

- It is *the* formal notation for writing perfectly clear, concise, and unambiguous mathematical *definitions*, *axioms*, and *theorems* (more on subsequent lectures) for *any* branch of mathematics.
- Predicate logic with function symbols, the "=" operator, and a few proof-building rules is sufficient for defining *any* conceivable mathematical system, and for proving anything that can be proved within that system!

Other Applications

- Predicate logic is the foundation of the field of *mathematical logic*, which culminated in *Gödel's incompleteness theorem*, which revealed the ultimate limits of mathematical thought:
  - Given any finitely describable, consistent proof procedure, there will always remain *some* true statements that will *never be proven* by that procedure.

Kurt Gödel
1906-1978

Practical Applications of Predicate Logic

- It is the basis for clearly expressed formal specifications for any complex system.
- It is basis for *automatic theorem provers* and many other Artificial Intelligence systems.
  - *E.g.* automatic program verification systems.
- Predicate-logic like statements are supported by some of the more sophisticated *database query engines* and *container class libraries*
  - these are types of programming tools.

Subjects and Predicates

- In the sentence "The dog is sleeping":
  - The phrase "the dog" denotes the *subject* - the *object* or *entity* that the sentence is about.
  - The phrase "is sleeping" denotes the *predicate*- a property that is true **of** the subject.
- In predicate logic, a *predicate* is modeled as a *function* $P(\cdot)$ from objects to propositions.
  - $P(x)$ = "x is sleeping" (where x is any object).

More About Predicates

- Convention: Lowercase variables x, y, z... denote objects/entities; uppercase variables P, Q, R... denote propositional functions (predicates).
- Keep in mind that the *result of applying* a predicate P to an object x is the *proposition* $P(x)$. But the predicate P **itself** (*e.g.* P="is sleeping") is **not** a proposition (not a complete sentence).
  - *E.g.* if $P(x)$ = "x is a prime number", $P(3)$ is the *proposition* "3 is a prime number."

Propositional Functions

- Predicate logic *generalizes* the grammatical notion of a predicate to also include propositional functions of **any** number of arguments, each of which may take **any** grammatical role that a noun can take.
  - *E.g.* let $P(x,y,z)$ = "x gave y the grade z", then if x="Mike", y="Mary", z="A", then

$P(x,y,z)$ = "Mike gave Mary the grade A."

Predicates

The predicate can have several variables:

- Friends(X,Y): X and Y are friends
- Parents(X,Y,Z): X is a child of Y and Z
- SumsTo7(x, y): $x+y=7$
- P(x,y,z): $x > y^2+z^3-5$

So, what are the truth values of:

- SumsTo7(4,5)
- P(27, 2, 3)
- WillPassCSI2101(Daniel Sousa)

Predicates in computing

The predicates are very useful in capturing the properties the variable values must satisfy before/during/after executing a code segment

Consider the code segment

```
temp = x;
x = y;
y = temp;
```

Precondition – what holds before the code segment:

- Q(x, y): x = a, y = b for some values a and b

Postcondition – what holds after the code segment:

- R(x,y): x = b, y = a
- also can be expressed as Q(y, x)

To verify that the postcondition holds, we must start from the assumption that the precondition holds and go over every stop of the code ad examine what it does to see whether the postcondition is really true.

Predicates in computing

Consider the code segment

```
if P(x, y) {
    temp = x;
    x = y;
    y = temp;
}
```

where P(x, y) is "x>y"

Precondition – what holds before the code segment:

- Q(x, y): x = a, y = b for some values a and b

Postcondition – what holds after the code segment:

- ???!

Universes of Discourse (U.D.s)

- The power of distinguishing objects from predicates is that it lets you state things about *many* objects at once.
- E.g., let $P(x)$="$x+1>x$". We can then say, "For *any* number x, $P(x)$ is true" instead of $(\mathbf{0}+1>\mathbf{0}) \wedge (\mathbf{1}+1>\mathbf{1}) \wedge (\mathbf{2}+1>\mathbf{2}) \wedge \ldots$
- The collection of values that a variable x can take is called x's *universe of discourse*.

Quantifier Expressions

- *Quantifiers* provide a notation that allows us to *quantify* (count) *how many* objects in the univ. of disc. satisfy a given predicate.
- "$\forall$" is the FOR$\forall$LL or *universal* quantifier. $\forall x\, P(x)$ means *for all* x in the u.d., P holds.
- "$\exists$" is the $\exists$XISTS or *existential* quantifier. $\exists x\, P(x)$ means there *exists* an x in the u.d. (that is, 1 or more) such that $P(x)$ is true.

The Universal Quantifier $\forall$

- Example:
  Let the u.d. of x be parking spaces at UO.
  Let $P(x)$ be the *predicate* "x is full."
  Then the *universal quantification of* $P(x)$, $\forall x\, P(x)$, is the *proposition:*
  - "All parking spaces at UO are full."
  - *i.e.*, "Every parking space at UO is full."
  - *i.e.*, "For each parking space at UO, that space is full."

The Existential Quantifier ∃

- Example:
  Let the u.d. of x be parking spaces at UO.
  Let $P(x)$ be the *predicate* "x is full."
  Then the *existential quantification of* $P(x)$, $\exists x\, P(x)$, is the *proposition*:
  - "Some parking space at UO is full."
  - "There is a parking space at UO that is full."
  - "At least one parking space at UO is full."

Free and Bound Variables

- An expression like $P(x)$ is said to have a *free variable* x (meaning, x is undefined).
- A quantifier (either $\forall$ or $\exists$) *operates* on an expression having one or more free variables, and *binds* one or more of those variables, to produce an expression having one or more *bound variables*.

Example of Binding

- $P(x,y)$ has 2 free variables, x and y.
- $\forall x\, P(x,y)$ has 1 free variable, and one bound variable. [Which is which?]
- "$P(x)$, where $x=3$" is another way to bind x.
- An expression with <u>zero</u> free variables is a bona-fide (actual) proposition.
- An expression with <u>one or more</u> free variables is still only a predicate: *e.g.* let $Q(y) = \forall x\, P(x,y)$

Nesting of Quantifiers

Example: Let the u.d. of x & y be people.

Let $L(x,y)$="x likes y" (a predicate w. 2 f.v.'s)

Then $\exists y\, L(x,y)$ = "There is someone whom x likes." (A predicate w. 1 free variable, x)

Then $\forall x\, (\exists y\, L(x,y))$ =
"Everyone has someone whom they like."
(A ____________ with ___ free variables.)

Proposition 0

Review: Predicate Logic (§1.3)

- Objects x, y, z, …
- Predicates P, Q, R, … are functions mapping objects x to propositions $P(x)$.
- Multi-argument predicates $P(x, y)$.
- Quantifiers: $[\forall x\, P(x)] :\equiv$ "For all x's, $P(x)$."
  $[\exists x\, P(x)] :\equiv$ "There is an x such that $P(x)$."
- Universes of discourse, bound & free vars.

Quantifier Exercise

If $R(x,y)$="x relies upon y," express the following in unambiguous English:

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

Everyone has *someone* to rely on.

$\exists y(\forall x\, R(x,$

There's a poor overburdened soul whom *everyone* relies upon (including himself)!

$\exists x(\forall y\, R(x,$

There's some needy person who relies upon *everybody* (including himself).

$\forall y(\exists x\, R($

Everyone has *someone* who relies upon them.

$\forall x(\forall y\, R(x,$

Everyone relies upon *everybody*, (including themselves)!

Natural language is ambiguous!

- "Everybody likes somebody."
  - For everybody, there is somebody they like,
    - $\forall x\, \exists y\; Likes(x,y)$ [Probably more likely.]
  - or, there is somebody (a popular person) whom everyone likes?
    - $\exists y\, \forall x\; Likes(x,y)$
- "Somebody likes everybody."
  - Same problem: Depends on context, emphasis.

Still More Conventions

- Sometimes the universe of discourse is restricted within the quantification, *e.g.*,
  - $\forall x{>}0\ P(x)$ is shorthand for "For all x that are greater than zero, $P(x)$."
    $=\forall x\,(x{>}0 \rightarrow P(x))$
  - $\exists x{>}0\ P(x)$ is shorthand for "There is an x greater than zero such that $P(x)$."
    $=\exists x\,(x{>}0 \wedge P(x))$

More to Know About Binding

- $\forall x \exists x\, P(x)$ - x is not a free variable in $\exists x\, P(x)$, therefore the $\forall x$ binding isn't used.
- $(\forall x\, P(x)) \wedge Q(x)$ - The variable x is outside of the *scope* of the $\forall x$ quantifier, and is therefore free. Not a complete proposition!
- $(\forall x\, P(x)) \wedge (\exists x\, Q(x))$ – This is legal, because there are 2 different x's!

Quantifier Equivalence Laws

- Definitions of quantifiers: If u.d.=a,b,c,…

  $\forall x\, P(x) \Leftrightarrow P(\mathrm{a}) \wedge P(\mathrm{b}) \wedge P(\mathrm{c}) \wedge \ldots$

  $\exists x\, P(x) \Leftrightarrow P(\mathrm{a}) \vee P(\mathrm{b}) \vee P(\mathrm{c}) \vee \ldots$

- From those, we can prove the laws:

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

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

- Which *propositional* equivalence laws can be used to prove this?

DeMorgan's

More Equivalence Laws

- $\forall x \forall y\, P(x,y) \Leftrightarrow \forall y \forall x\, P(x,y)$
  $\exists x \exists y\, P(x,y) \Leftrightarrow \exists y \exists x\, P(x,y)$
- $\forall x\, \mathbf{(}P(x) \wedge Q(x)\mathbf{)} \Leftrightarrow \mathbf{(}\forall x\, P(x)\mathbf{)} \wedge \mathbf{(}\forall x\, Q(x)\mathbf{)}$
  $\exists x\, \mathbf{(}P(x) \vee Q(x)\mathbf{)} \Leftrightarrow \mathbf{(}\exists x\, P(x)\mathbf{)} \vee \mathbf{(}\exists x\, Q(x)\mathbf{)}$
- Exercise: See if you can prove these yourself.
  - What propositional equivalences did you use?

Review: Predicate Logic (§1.3)

- Objects x, y, z, …
- Predicates P, Q, R, … are functions mapping objects x to propositions $P(x)$.
- Multi-argument predicates $P(x, y)$.
- Quantifiers: $(\forall x\, P(x))$ =“For all x's, $P(x)$.” $(\exists x\, P(x))$=“There is an x such that $P(x)$.”

More Notational Conventions

- Quantifiers bind as loosely as needed: parenthesize $\forall x \, \big(P(x) \wedge Q(x) \big)$
- Consecutive quantifiers of the same type can be combined: $\forall x \, \forall y \, \forall z \, P(x,y,z) \Leftrightarrow \forall x,y,z \, P(x,y,z)$ or even $\forall xyz \, P(x,y,z)$
- All quantified expressions can be reduced to the canonical *alternating* form $\forall x_1 \exists x_2 \forall x_3 \exists x_4 \ldots \, P(x_1, x_2, x_3, x_4, \ldots)$

Defining New Quantifiers

- As per their name, quantifiers can be used to express that a predicate is true of any given *quantity* (number) of objects.
- Define $\exists! x\, P(x)$ to mean "$P(x)$ is true of *exactly one* x in the universe of discourse."
- $\exists! x\, P(x) \Leftrightarrow \exists x\, (P(x) \wedge \neg \exists y\, (P(y) \wedge y \neq x))$
  "There is an x such that $P(x)$, where there is no y such that $P(y)$ and y is other than x."

Some Number Theory Examples

- Let u.d. = the *natural numbers* 0, 1, 2, …
- "A number x is *even*, $E(x)$, if and only if it is equal to 2 times some other number."
  $\forall x\,(E(x) \leftrightarrow (\exists y\;\; x=2y))$
- "A number is *prime*, $P(x)$, iff it's greater than 1 and it isn't the product of any two non-unity numbers."

  $\forall x\,(P(x) \leftrightarrow (x>1 \land \neg\exists yz\;\; x=yz \land y\neq 1 \land z\neq 1))$

Goldbach's Conjecture (unproven)

Using $E(x)$ and $P(x)$ from previous slide,

$$\forall E(x{>}2): \exists P(p), P(q): p+q = x$$

or, with more explicit notation:

$$\forall x\,[x{>}2 \wedge E(x)] \rightarrow$$
$$\exists p\,\exists q\, P(p) \wedge P(q) \wedge p+q = x.$$

"Every even number greater than 2 is the sum of two primes."

Calculus Example

- One way of precisely defining the calculus concept of a *limit*, using quantifiers:

$$\left(\lim_{x \to a} f(x) = L\right) \Leftrightarrow$$

$$\left(\begin{array}{l}\forall \varepsilon > 0 : \exists \delta > 0 : \forall x : \\ \left(|x - a| < \delta\right) \rightarrow \left(|f(x) - L| < \varepsilon\right)\end{array}\right)$$

Deduction Example

- Definitions:

  s :≡ Socrates (ancient Greek philosopher);

  $H(x)$:≡ "x is human";

  $M(x)$:≡ "x is mortal".

- Premises:

  $H(s)$ *Socrates is human.*

  $\forall x\, H(x) \rightarrow M(x)$ *All humans are mortal.*

Deduction Example Continued

Some valid conclusions you can draw:

$H(s) \rightarrow M(s)$ **[Instantiate universal.]**

If Socrates is human then he is mortal.

$\neg H(s) \vee M(s)$ *Socrates is inhuman or mortal.*

$H(s) \wedge (\neg H(s) \vee M(s))$

Socrates is human, and also either inhuman or mortal.

$(H(s) \wedge \neg H(s)) \vee (H(s) \wedge M(s))$ **[Apply distributive Law.]**

$\mathbf{F} \vee (H(s) \wedge M(s))$ **[Trivial contradiction.]**

$H(s) \wedge M(s)$ **[Use identity law.]**

$M(s)$ *Socrates is mortal.*

Another Example

- Definitions:
- $H(x) :\equiv$ "x is human"; $M(x) :\equiv$ "x is mortal"; $G(x) :\equiv$ "x is a god"
- Premises:
  - $\forall x\, H(x) \rightarrow M(x)$ ("Humans are mortal") and
  - $\forall x\, G(x) \rightarrow \neg M(x)$ ("Gods are immortal").
- Show that $\neg\exists x\, (H(x) \wedge G(x))$
  ("No human is a god.")

The Derivation

- $\forall x\, H(x) \rightarrow M(x)$ and $\forall x\, G(x) \rightarrow \neg M(x)$.
- $\forall x\, \neg M(x) \rightarrow \neg H(x)$ **[Contrapositive.]**
- $\forall x\, [G(x) \rightarrow \neg M(x)] \wedge [\neg M(x) \rightarrow \neg H(x)]$
- $\forall x\, G(x) \rightarrow \neg H(x)$ **[Transitivity of $\rightarrow$.]**
- $\forall x\, \neg G(x) \vee \neg H(x)$ **[Definition of $\rightarrow$.]**
- $\forall x\, \neg(G(x) \wedge H(x))$ **[DeMorgan's law.]**
- $\neg \exists x\, G(x) \wedge H(x)$ **[An equivalence law.]**

End of §1.3-1.4, Predicate Logic

- From these sections you should have learned:
  - Predicate logic notation & conventions
  - Conversions: predicate logic $\leftrightarrow$ clear English
  - Meaning of quantifiers, equivalences
  - Simple reasoning with quantifiers