First-Order Logic Representation

Previously

$\models$

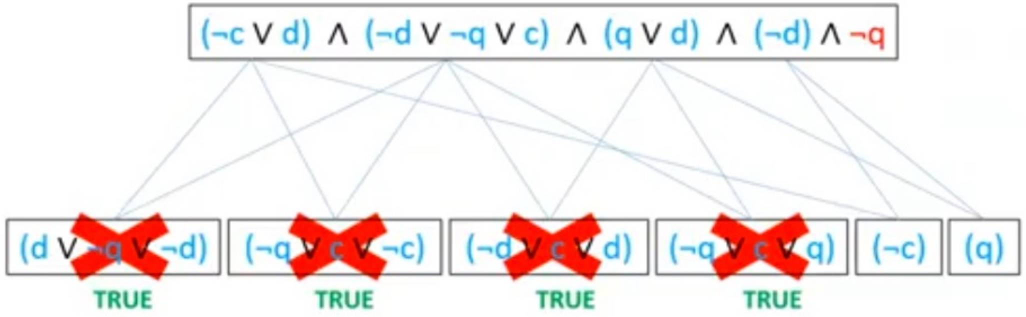
- Propositional Logic (Reasoning):
  - Entailment
  - Arguments
  - Inference Rules
  - Deductive Reasoning From a Knowledge Base

$$\begin{array}{l} \quad p \rightarrow q \\ \quad p \\ \therefore q \end{array}$$

Modus Ponens

p	q	p → q	p ∧ (p → q)	(p ∧ (p → q)) → q
0	0	1	0	1
0	1	1	0	1
1	0	0	0	1
1	1	1	1	1

Resolution

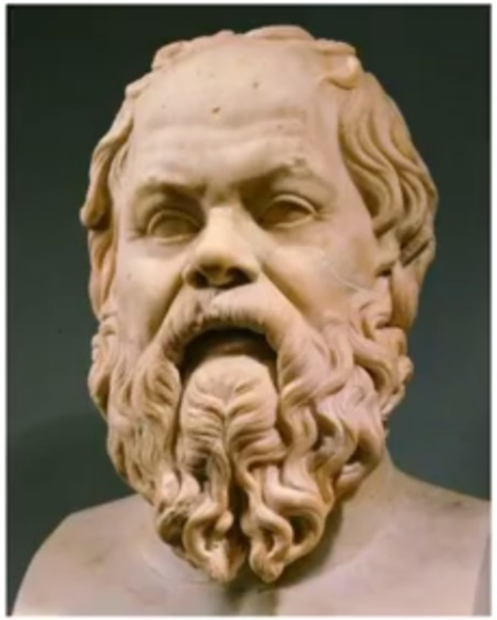


Lecture Outline

- What is Predicate Logic?
- Predicate Logic Syntax
- Predicate Logic Equivalence
- Predicate Logic Semantics

What is
Predicate Logic?

The Socrates Argument



All men are mortal

Socrates is a man

Therefore, Socrates is mortal

Written in propositional logic →

$$\begin{array}{r} p \\ q \\ \therefore r \end{array}$$

Does this make sense?

The validity of this argument comes from the internal structure of these sentences--which propositional logic cannot "see" (indivisible)

The Problem with Propositional Logic

- Not possible to write general statements as propositions
  - “Finn eats everything that he likes”

The Problem with Propositional Logic

- Not possible to write general statements as propositions
  - "Finn eats everything that he likes"
- This would require an enumeration of propositions:
  - "Finn eats what he likes"
  - "Finn likes cake", "Finn likes bananas", "Finn likes kale"…
- Socrates Argument:
  - Would it be practical to enumerate all people who are mortal?

What is Needed?

- We need to split the atom!
- Display the internal structure of those "atomic" sentences by adding new categories of vocabulary items
- Give a semantic account of these new vocabulary items
- Introduce rules for operating with them

Why First Order Predicate Logic?

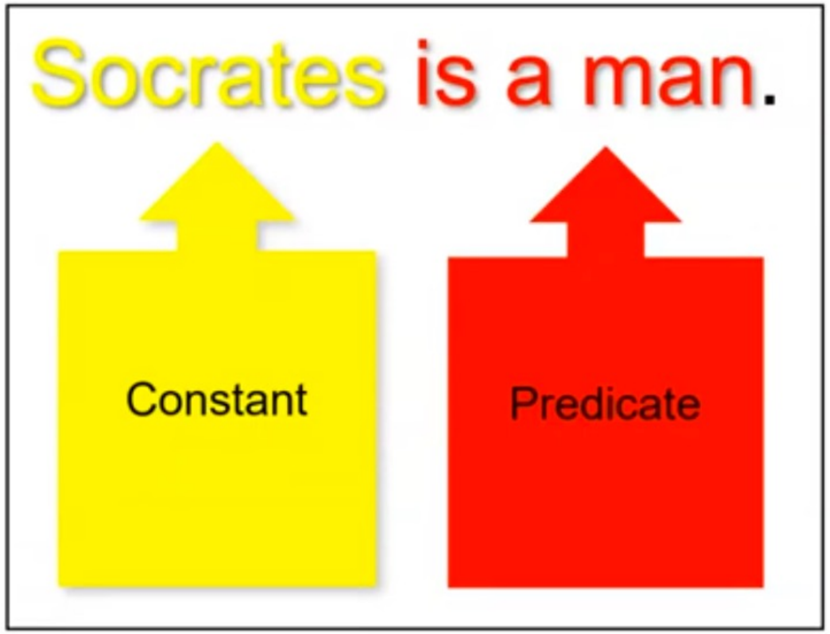
- Flexibility and conciseness!
- Predicate logic makes general statements possible
- Rather than having simple, concrete propositions:
  - Predicate logic allows for conditional and collective propositions:
    - Statements of existence and statements of number
- Important for dealing with more complex problems and environments

First-Order Predicate Logic

- Aka: Predicate Logic, First-Order Logic (FOL), Quantificational Logic, First-Order Predicate Calculus

- Abstraction of propositional logic

- Requires additional syntax
  - Constants
  - Variables
  - Predicates
  - Functions
  - Equality
  - Quantifiers



- Syllogistic logic is like learning to count
- Propositional logic is like learning to do basic arithmetic
- Predicate logic is like learning algebra
- They are a progressive elaboration

‘First-Order’

- Distinguishes first-order logic from higher-order logic:
  - First-order: (Simpler)
    - Predicates quantify objects or things (work with a set)
  - Higher-order: (More complex)
    - Quantifiers can refer to predicates or functions
    - Predicates can have nested predicates or functions (sets of sets)

Predicate Logic Syntax

Individual 'Constants'

- Are invariable nouns, i.e. their value/meaning does not change

- Represent objects: persons, places, things, times…
  - E.g. Ryan, Socrates, Ohio, Saturday, cake, Philadelphia

- Can be represented by most any symbol (like a proposition):
  - Some conventions use lower case letters (e.g. a,b,c,…u,v,w)
  - For natural interpretability, can use the original noun name

- Truth Symbols (T & F) are preserved from propositional logic

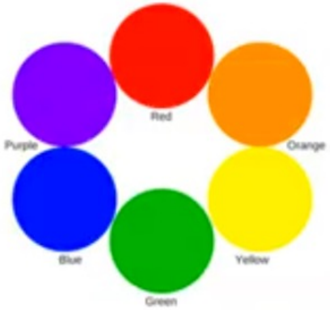




Variables

- Are symbols that stands for an individual in a collection or set

- E.g. The variable *x* may stand for one of the days
  - *x* = Monday, *x* = Tuesday, …
- Normally use letters at the end of the alphabet such as *x*, *y*, *z*
- A collection of objects is called the domain of objects
- For the above example, the days in the week is the domain of variable *x*



Predicates

- Aka: propositional functions, function expressions

- Verbal statement which describes:
  - The property of some individual (constant or variable)
    - E.g. "__ is human"
  - The relation between individuals
    - E.g. "__ the teacher of __"
    - "__ is between __ and __"

  - Returns a Boolean value; True or False

Predicates

- Aka: propositional functions, function expressions

- Verbal statement which describes:
  - The property of some individual (constant or variable)
    - E.g. "__ is human"
  - The relation between individuals
    - E.g. "__ the teacher of __"
    - "__ is between __ and __"

  - Returns a Boolean value; True or False

  - Can be represented by most any symbol:
    - For natural interpretability, can use the original noun name
    - Some conventions use upper case letters (e.g. A,B,C,…U,V,W)

- Atomic Sentences:
  - "Finn likes chocolate" represented as:
    - *likes(Finn, chocolate)*
  - "Finn likes all food" represented as:
    - *likes(Finn, x)* where *x* is a variable within the domain 'food'

Statements in Predicate Logic

- Are constructed (much as in propositional logic)
  - By combining atomic sentences with logical connectives
    - Conjunction (∧): AND
    - Disjunction (∨): OR
    - Negation (¬): NOT
    - Implication (→): IF….THEN
    - Equivalence (⇔): IF AND ONLY IF
- Examples:
  - "If Alison is friends with Richard then she likes him."
    - friends(alison,richard) → likes(alison,richard)
  - "Alison likes Richard or chocolate."
    - likes(alison,richard) ∨ likes(alison,chocolate)
  - "Alison doesn't like rain and snow."
    - ¬likes(alison,rain) ∧ ¬likes(alison,snow)

Functions

- Recall that constants are 'fixed' objects
  - Its not always concise to enumerate constants for every object

- Functions are similar (at first glance) to predicates but flexibly map terms to other terms (i.e. don't return a truth)

- Examples:
  - Mathematical function: $p(x) \longrightarrow x^2$
    - Any value of x get's squared
  - Object: LeftLeg(John)
    - Interpreted as "John's left leg"
    - Rather than specifying a constant name for John's left leg

Quantifiers

- An operator allowing more general statements that indicate how many objects have a certain property

- Two kinds:
  - Universal Quantifier: represented by $\forall$, "for all", "for every", "for each", or "for any"
    - Implication is the main connective
    - E.g. "Finn eats everything that he likes."
      $\forall x(\text{likes}(\text{Finn},x) \rightarrow \text{eats}(\text{Finn},x))$
  - Existential Quantifier: represented by $\exists$, "for some", "there exists", "there is a", or "for at least one"
    - Conjunction is the main connective
    - E.g. "There exists some bird that doesn't fly."
      $\exists x(\text{bird}(x) \land \neg\text{flies}(x))$
    - E.g. "Every person has something that they love."
      $\forall x(\text{person}(x) \rightarrow \exists y\ \text{loves}(x,y))$

Quantifiers: Common Mistakes

- Using ∧ as the main connective with ∀:
  - $\forall x(\text{likes}(\text{Finn},x) \land \text{eats}(\text{Finn},x))$
  - Translates to: "Finn likes everything and eats everything."
  - Intended: "Finn eats everything that he likes."
- Using → as the main connective with ∃:
  - $\exists x(\text{bird}(x) \rightarrow \neg\text{flies}(x))$
  - Translation very difficult, but this statement is only true if there is something other than a bird:
  - Intended: "There exists some bird that doesn't fly."

Unary, Binary,.. N-ary Predicates

- Unary Predicates:
  - Involving properties of a single variable

  - $\forall x\ (house(x) \rightarrow physical\_object(x))$
  - “Every house is a physical object”

- Binary Predicate:
  - Involving properties of two variables

  - $\forall x,y\ (Brother(x,y) \rightarrow Sibling(x,y))$
  - “Brothers are siblings.”

- Ternary and n-ary predicates are also possible

Sentences with Quantifiers

- Brothers are siblings.
  - $\forall x,y\ (Brother(x,y) \longrightarrow Sibling(x,y))$
- 'Sibling' is symmetric
  - $\forall x,y\ (Sibling(x,y) \Leftrightarrow Sibling(y,x))$
- One's mother is one's female parent
  - i.e. defining mother in terms of parentage
  - $\forall x,y\ (Mother(x,y) \Leftrightarrow (Female(x) \land Parent(x,y)))$
- A first cousin is a child of a parent's sibling
  - i.e. defining 'first cousin' in terms of parentage and siblingship
  - $\forall x,y\ (FirstCousin(x,y) \Leftrightarrow \exists p,ps\ Parent(p,x) \land Sibling(ps,p) \land Parent(ps,y))$

Scope of a Variable in an Expression

- The scope of a variable is the sentence to which the quantifier syntactically applies

- Examples:
  - $\exists x\,(Cat(x) \land \forall x(Black\,(x)))$
    - The x in Black(x) is universally quantified, but elsewhere it's not
  - "There is a cat that is all black"

  - $\forall x\,((\exists y)(P(x,y) \land Q(x,y)) \rightarrow R(x))$
    - Scope of $\forall x$ is entire expression
    - Scope of $\exists y$ is $(P(x,y) \lor Q(x,y))$

  - $\forall x\,(P(x,y) \rightarrow (\exists y)\,Q(x,y))$
    - Scope of $\forall x$ is entire expression
    - Scope of y not defined for $P(x,y)$ hence y is a free variable

Bound vs Free Variables

- Variables are bound or free based on their connection to some quantifier

- Bound variable: a variable is in the scope of some quantifier
  - Variables can be given specific values (as constants) or can be constrained by quantifiers
  - $\forall x\ (A(x) \rightarrow B(x))$
  - x is bound by $\forall$
  - A formula (wff) with no free variables is called a closed formula, a sentence, or a proposition, and has a truth value

- Free variable: any variable that is not bound by a quantifier
  - To evaluate the formula the free variable must be replaced by an object/value
  - $\forall x\ ((\exists y\ A(x,y)) \rightarrow B(z))$
  - Such an expression is simply a formula

Ground and Closed Formulas

- Ground Formula:
  - A formula F is ground if it does not contain variables
  - Example:
    - Fish (Nemo)
- Closed Formula:
  - A formula F is closed if it does not contain free variables
  - Examples:
    - $\forall x\,(P(x) \rightarrow G(x))$ (closed, not ground)
    - $\forall x(P(x) \rightarrow G(y, x))$ (not closed, not ground)

Negation of Quatification

- Be aware of how to translate negations of quantification
- Negation of a universal quantification becomes an existential quantification, and vice versa
- "Everything is beautiful" $\forall x\, P(x)$
- "Not everything is beautiful" $\neg\forall x\, P(x) \equiv \exists x\, \neg P(x)$
  - "There is at least something that is not beautiful"
- "It is not the case that something is beautiful" $\neg\exists x\, P(x) \equiv \forall x\, \neg P(x)$
  - "Everything is not beautiful"
  - "Nothing is beautiful"

De Morgan:

$\neg(P \wedge Q) \overset{val}{=\!=} \neg P \vee \neg Q$

$\neg(P \vee Q) \overset{val}{=\!=} \neg P \wedge \neg Q$

Uniqueness Quantification

- Sometimes we want to say that there is a single, unique object that satisfies a certain condition (i.e. only one)

- "There exists a unique x such that king(x) is true"
  - How is this done with our existing operators and connectives?
    - $\exists x\ king(x) \land \forall y\ (king(y) \rightarrow x=y)$
    - $\exists x\ king(x) \land \neg\exists y\ (king(y) \land x \neq y)$
  - How is this done with a uniqueness quantification?
    - $\exists! x\ king(x)$

- "Every country has exactly one ruler"
  - $\forall c\ (country(c) \rightarrow \exists! r\ ruler(c,r))$

- Iota operator: "ι x P(x)" means "the unique x such that p(x) is true"
  - "The unique ruler of Freedonia is dead"
  - $dead(\iota\ x\ ruler(freedonia,x))$

Equality (=)

- Not strictly the same as equivalence!
  - (=) vs (≡)

- Equality indicates that two terms refer to the same objects

- Examples:
  - Consider the function: father(John)
    - father(John) = Henry
    - This function refers to the same object as the constant Henry
  - $\exists x,y\ (Brother(x,Richard) \land Brother(y,Richard) \land \neg(x=y))$
    - "Richard has at least two brothers"

Precedence

- The quantifiers $\forall$ and $\exists$ have higher precedence than all logical operators from propositional calculus
- Then other connectives as in predicate logic
- As in propositional logic use parentheses to disambiguate

Operator	Precedence
$\neg$	1
$\wedge$ $\vee$	2 3
$\rightarrow$ $\leftrightarrow$	4 5

Translation Practice Examples

"Every gardener likes the sun."

$$\forall x\ (gardener(x) \rightarrow likes(x,Sun))$$

"You can fool some of the people all of the time."

$$\exists x\ \forall y\ \ (person(x) \wedge time(y) \rightarrow can\_fool(x,y))$$

"You can fool all of the people some of the time."

$$\forall x\ \exists y\ (person(x) \rightarrow time(y) \wedge can\_fool(x,y))$$

$$\forall x\ (person(x) \rightarrow \exists y\ (time(y) \wedge can\_fool(x,y)))$$

Equivalent

"All purple mushrooms are poisonous."

$$\forall x\ (mushroom(x) \wedge purple(x)) \rightarrow poisonous(x)$$

"No purple mushroom is poisonous."

$$\neg\exists x\ (purple(x) \wedge mushroom(x) \wedge poisonous(x))$$

$$\forall x\ \ ((mushroom(x) \wedge purple(x)) \rightarrow \neg poisonous(x))$$

Equivalent

"There are exactly two purple mushrooms."

$$\exists x\ \exists y\ (mushroom(x) \wedge purple(x) \wedge mushroom(y) \wedge purple(y) \wedge \neg(x=y) \wedge \forall z\ (mushroom(z) \wedge purple(z)) \rightarrow ((x=z) \vee (y=z)))$$

Predicate Logic Equivalence

Return to Equivalence

- All logical equivalences defined in propositional logic still hold in FOL

Distributivity:
$P \wedge (Q \vee R) \stackrel{val}{=} (P \wedge Q) \vee (P \wedge R)$,
$P \vee (Q \wedge R) \stackrel{val}{=} (P \vee Q) \wedge (P \vee R)$

Inversion:
$\neg\texttt{True} \stackrel{val}{=} \texttt{False}$,
$\neg\texttt{False} \stackrel{val}{=} \texttt{True}$

Contradiction:
$P \wedge \neg P \stackrel{val}{=} \texttt{False}$

Excluded Middle:
$P \vee \neg P \stackrel{val}{=} \texttt{True}$

Bi-implication:
$P \Leftrightarrow Q \stackrel{val}{=} (P \Rightarrow Q) \wedge (Q \Rightarrow P)$

Double Negation:
$\neg\neg P \stackrel{val}{=} P$

Negation:
$\neg P \stackrel{val}{=} P \Rightarrow \texttt{False}$

Associativity:
$(P \wedge Q) \wedge R \stackrel{val}{=} P \wedge (Q \wedge R)$,
$(P \vee Q) \vee R \stackrel{val}{=} P \vee (Q \vee R)$,
$(P \Leftrightarrow Q) \Leftrightarrow R \stackrel{val}{=} P \Leftrightarrow (Q \Leftrightarrow R)$

De Morgan:
$\neg(P \wedge Q) \stackrel{val}{=} \neg P \vee \neg Q$,
$\neg(P \vee Q) \stackrel{val}{=} \neg P \wedge \neg Q$

Implication:
$P \Rightarrow Q \stackrel{val}{=} \neg P \vee Q$

Contraposition:
$P \Rightarrow Q \stackrel{val}{=} \neg Q \Rightarrow \neg P$

Self-equivalence:
$P \Leftrightarrow P \stackrel{val}{=} \texttt{True}$

True/False-elimination:
$P \wedge \texttt{True} \stackrel{val}{=} P$,
$P \wedge \texttt{False} \stackrel{val}{=} \texttt{False}$,
$P \vee \texttt{True} \stackrel{val}{=} \texttt{True}$,
$P \vee \texttt{False} \stackrel{val}{=} P$

Commutativity:
$P \wedge Q \stackrel{val}{=} Q \wedge P$,
$P \vee Q \stackrel{val}{=} Q \vee P$,
$P \Leftrightarrow Q \stackrel{val}{=} Q \Leftrightarrow P$

Idempotence:
$P \wedge P \stackrel{val}{=} P$,
$P \vee P \stackrel{val}{=} P$

Absorption Laws
$p \vee (p \wedge q) \equiv p$
$p \wedge (p \vee q) \equiv p$

- However, there are additional rules for formulas containing quantifiers

Equivalence in FOL

- Statements involving predicates and quantifiers are logically equivalent **iff** they have the same truth value no matter which:
  - Predicates are substituted into these statements
  - Domain of discourse is used for the variables in these propositional functions

- Applying the previous propositional equivalences
  - Outside a propositional function with quantifier:
    - $\neg\neg(\forall x\, A(x)) \equiv (\forall x\, A(x))$
  - Inside the scope of a quantifier:
    - $\forall x\, (\neg\neg P(x)) \equiv \forall x\, (P(x))$
    - $\forall x \exists y\, (A(x) \rightarrow B(y)) \equiv \forall x \exists y\, (\neg B(y) \rightarrow \neg A(x))$

Double Negation:
$\neg\neg P \overset{val}{=\!=} P$

Contraposition:
$P \Rightarrow Q \overset{val}{=\!=} \neg Q \Rightarrow \neg P$

- But be careful not to assume too much here
  - Check equivalence rules first!

Basic Equivalences with Quantifiers

- Formulas are logically equivalent if they differ in:
  - The name of variables in the scope of quantifiers
    - $\forall x\, P(x) \equiv \forall y\, P(y)$
  - Addition or elimination of quantifiers whose variable does not occur in their scope
    - $\forall x\, P(x) \equiv P(y)$
  - The order of quantifiers of the same kind (nested quantifiers)
    - $\forall x \forall y\, P(x,y) \equiv \forall y \forall x\, P(x,y) \equiv \forall y,x\, P(x,y)$

Equivalence of Nested Quantifiers

- $\forall x \forall y\ P(x,y) \equiv \forall y \forall x\ P(x,y)$

  - $\forall x\ \forall y\ Loves(x,y) \equiv \forall y\ \forall x\ Loves(x,y)$
    - "Everyone in the world loves everyone in the world"

- $\exists x \exists y\ P(x,y) \equiv \exists y \exists x\ P(x,y)$

  - $\exists x\ \exists y\ Loves(x,y) \equiv \exists y\ \exists x\ Loves(x,y)$
    - "At least one person in the world loves at least one other person in the world"

Equivalence of Nested Quantifiers

- $\forall x \forall y\, P(x,y) \equiv \forall y \forall x\, P(x,y)$
  - $\forall x\, \forall y$ Loves(x,y) $\equiv$ $\forall y\, \forall x$ Loves(x,y)
    - "Everyone in the world loves everyone in the world"
- $\exists x \exists y\, P(x,y) \equiv \exists y \exists x\, P(x,y)$
  - $\exists x\, \exists y$ Loves(x,y) $\equiv$ $\exists y\, \exists x$ Loves(x,y)
    - "At least one person in the world loves at least one other person in the world"
- $\forall x \exists y\, P(x,y)$ is not the same as $\forall y \exists x\, P(x,y)$
  - $\forall x\, \exists y$ Loves(x,y)
    - "Everyone in the world loves at least one person"
  - $\forall y\, \exists x$ Loves(x,y)
    - "At least one person loves everyone in the world."

Equivalence of Negated Quatification

- Saw these first two earlier…

De Morgan:

$$\neg(P \wedge Q) \stackrel{val}{=\!=} \neg P \vee \neg Q$$
$$\neg(P \vee Q) \stackrel{val}{=\!=} \neg P \wedge \neg Q$$

- From De Morgan's laws:
  - $\neg\forall x\, P(x) \equiv \exists x\, \neg P(x)$ "Someone doesn't love Finn"
  - $\neg\exists x\, P(x) \equiv \forall x\, \neg P(x)$ "Everyone doesn't love Finn"

- Related Equalities
  - $\forall x\, \neg P(x) \equiv \neg\exists x\, P(x)$ "No one loves Finn"
  - $\forall x\, P(x) \equiv \neg\exists x\, \neg P(x)$ "Everyone loves Finn"
  - $\exists x\, P(x) \equiv \neg\forall x\, \neg P(x)$ "Someone loves Finn"

Other Equivalences with Quantifiers

- Trivial Conjunctions
  - $\forall x\,(P(x) \land Q(x)) \equiv \forall x P(x) \land \forall x Q(x)$
    - "All things are and were"
  - $\exists x\,(P(x) \lor Q(x)) \equiv \exists x P(x) \lor \exists x Q(x)$
    - "Some things are or were"
- However:
  - $\forall x\,(P(x) \lor Q(x)) \not\equiv \forall x P(x) \lor \forall x Q(x)$
  - $\exists x\,(P(x) \land Q(x)) \not\equiv \exists x P(x) \land \exists x Q(x)$

Predicate Logic Semantics

Interpretation vs Model

- An interpretation specifies/assigns/instantiates the meaning and values of a formula's signature given a domain

- The signature includes all non-logical symbols
  - Predicates
  - Constants
  - Functions

- This yields a proposition with a corresponding truth value

- A model is an interpretation that satisfies a sentence

Domain of Predicate Variables

- The domain ($\mathscr{U}$) is the set of all possible values that the variable may take
  - E.g. $\mathscr{U} = \{1,2,3,4\}$ …or… $\mathscr{U} = \{\mathbb{R}\}$ (i.e. all real numbers}
  - Each predicate variable may have different domains

- Example:
  - Given predicate, P(x, y) = "x > y"
  - $\mathscr{U}$ = {all integers}
  - Instantiated:
    - P(4, 3) means "4 >3", so TRUE
    - P(1, 2) means "1>2" , so FALSE

- Can explicitly define the domain in a formula:
  - i.e. Conditional quantification
  - $(\forall x \in \mathbb{R})\ P(x)$
    - i.e. the domain is the set of all real numbers

- A predicate instantiated is a proposition
  - Where variables are evaluated with specific values

Sets

- A set { } is a well defined collection of objects

- The elements of a set, i.e. members, can be anything:
  - Numbers, people, letters of the alphabet, other sets, etc

- Sets A and B are equal **iff** they have precisely the same elements

- Sets are “abstract objects”:
  - They don’t occupy time or space, or have causal powers even if their members do

Substitution

- Replacement of a formula's signature with objects or values from the domain

- Important part of determining formula truth needed for inference in FOL

- Example:
  - $\forall x\,(\text{Fish}(x) \rightarrow \text{Swim}(x))$, Fish(Nemo)
  - To determine if Nemo can swim we must replace every occurrence of the variable x in the implication by the term Nemo (an object from the real world)
  - $\forall x\,(\text{Fish}(\textit{Nemo}) \rightarrow \text{Swim}(\textit{Nemo}))$
  - Now we can evaluate truth

- Formally, substitution takes place within each atomic sentence
  - Only applies to free variables within each atomic sentence
    - Ignore bindings from the sentence, as a whole, in determining 'free variables' here
  - Substitution is formally denoted here by: Fish[Nemo/x]

Quantifiers as Conjunctions/Disjunctions

- If the domain is finite then:
  - Universal/existential quantifiers can be expressed by conjunctions/disjunctions
- If the domain $\mathscr{U} = \{1,2,3,4\}$, then…
  - $\forall x\, P(x) = P(1) \wedge P(2) \wedge P(3) \wedge P(4)$
  - $\exists x\, P(x) = P(1) \vee P(2) \vee P(3) \vee P(4)$
- If the domains are infinite the equivalent expressions without quantifiers will be infinitely long
- Helpful when considering sentence truthfulness
  - Consider: $P(x) = x^2 < 10$

Sentence Satisfiability

- A sentence in predicate calculus is satisfiable **iff** it is true for:
  - Some domain
  - Some propositional function substituted for the predicates in the sentence

- An element (e.g. x) for which makes a sentence false is called a counterexample

- Satisfiable example:
  - $\forall x \, \exists y \, P(x,y)$

- Unsatisfiable/inconsistent (no model) example:
  - $\forall x \, (P(x) \land \neg P(x))$
  - Can never be true

Sentence Validity

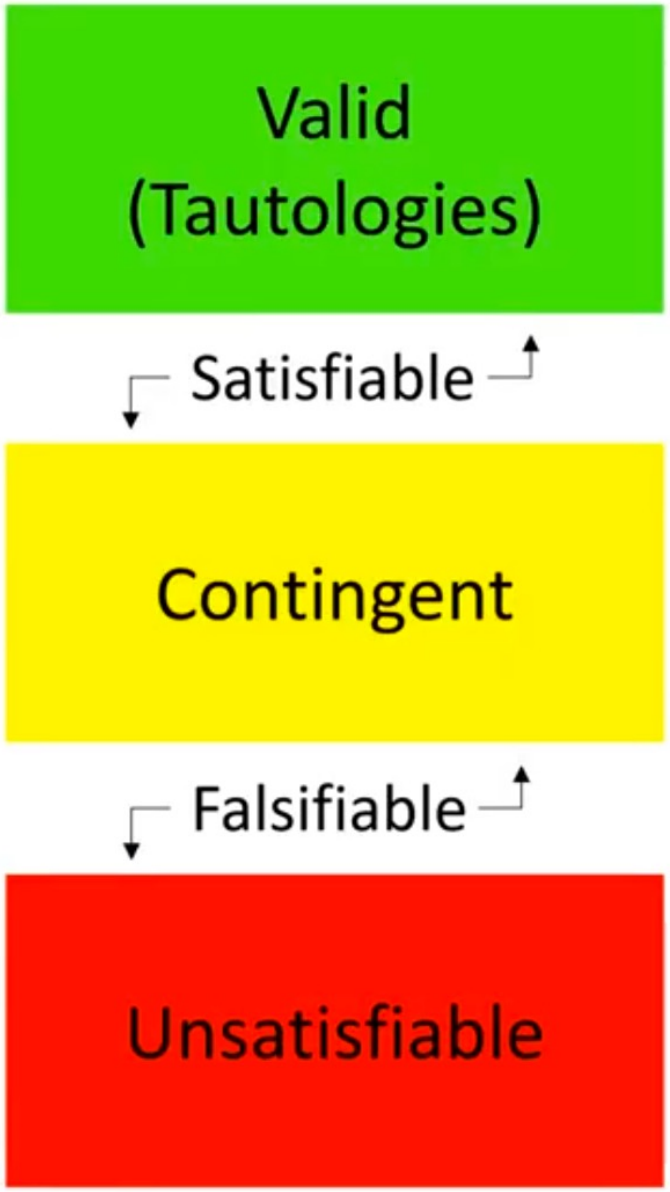
- A sentence in predicate calculus is valid if it is true in every interpretation

  - **iff** it is true for
    - All domains
    - Every propositional function substituted for the predicates in the sentence

  - It is true for all rows of the truth table

- Valid sentences in predicate logic play a role similar to tautologies in propositional logic
  - However they are referred to as first order validity

Properties of Abstract Propositions



- Every interpretation is satisfied (i.e. True)
- Some interpretations satisfy it but others don't
- No interpretations are satisfied

Tautology in Propositional Logic

- An abstract proposition that is always true

- Example:
  - This course is easy `and' this course is not easy

    $p \vee (\neg p) \equiv T$

- Truth table column is always True
  - (0 = False, 1 = True)

a	b	$b \Rightarrow a$	$a \Rightarrow (b \Rightarrow a)$
0	0	1	1
0	1	0	1
1	0	1	1
1	1	1	1

First-Order Validity in Predicate Logic

- A first-order predicate logic sentence is first order valid if it is satisfied for every interpretation

$$\forall x.P(x) \rightarrow \exists x.P(x);$$

$$\forall x.P(x) \rightarrow P(c);$$

$$P(c) \rightarrow \exists x.P(x);$$

$$\forall x(P(x) \leftrightarrow \neg\neg P(x));$$

$$\forall x(\neg(P_1(x) \wedge P_2(x)) \leftrightarrow (\neg P_1(x) \vee \neg P_2(x))).$$

Truth in Quantified Statements

Statement	When true	When false
$\forall x \in D, P(x)$	P(x) is true for every x.	There is one x for which P(x) is false.
$\exists x \in D, P(x)$	There is one x for which P(x) is true.	P(x) is false for every x.

Truth in Quantified Statements

Statement	When true	When false
$\forall x \in D, P(x)$	P(x) is true for every x.	There is one x for which P(x) is false.
$\exists x \in D, P(x)$	There is one x for which P(x) is true.	P(x) is false for every x.

TABLE 1 Quantifications of Two Variables.

Statement	*When True?*	*When False?*
$\forall x \forall y P(x, y)$ $\forall y \forall x P(x, y)$	$P(x, y)$ is true for every pair x, y.	There is a pair x, y for which $P(x, y)$ is false.
$\forall x \exists y P(x, y)$	For every x there is a y for which $P(x, y)$ is true.	There is an x such that $P(x, y)$ is false for every y.
$\exists x \forall y P(x, y)$	There is an x for which $P(x, y)$ is true for every y.	For every x there is a y for which $P(x, y)$ is false.
$\exists x \exists y P(x, y)$ $\exists y \exists x P(x, y)$	There is a pair x, y for which $P(x, y)$ is true.	$P(x, y)$ is false for every pair x, y.

How to Determine Truth Value?

- Systematic approaches:
  - Method of exhaustion
  - Method of case
  - Method of logic derivation

Method of Exhaustion

- If the domain contains a small number of elements, try them all!

- Let D={5,6,7,8,9}. Is $\exists\, x \in D,\ x^2 = x$ true or false?
  - $5^2=25\neq 5$, $6^2=36\neq 6$, $7^2=49\neq 7$, $8^2=64\neq 8$, $9^2=81\neq 9$
  - So, false!

- Limitation?
  - Domain may be too large to try out all options
    - E.g., all integers

Method of Case

- Realization: we can't hope to exhaustively search a domain

  - Positive examples to prove existential quantification
  - Let ***Z*** denote all integers. Is $\exists\, x \in Z,\, x^2 = x$ *true or false?*
    - Take $x = 0$ or 1 and we have it. True.

  - Counterexample to disprove universal quantification
  - Let R denote all reals. Is $\forall x \in R,\, x^2 > x$ *true or false?*
    - Take $x = 0.3$ as a counterexample. False.

Positive example is **not** a proof of universal quantification

Negative example **is not** disproof of existential quantification

May be **hard** to find suitable "cases" even if such cases do exist!

Method of Logical Derivation

- This method consists of using logical steps to transform one logical expression into another

Consider an (arbitrary) domain with n members. Is $\exists x(P(x) \vee Q(x))$ logically equivalent to $\exists xP(x) \vee \exists xQ(x)$?

$$\exists x(P(x) \vee Q(x))$$

$$\equiv [P(x_1) \vee Q(x_1)] \vee \dots \vee [P(x_n) \vee Q(x_n)]$$

$$\equiv [P(x_1) \vee \dots \vee P(x_n)] \vee [Q(x_1) \vee \dots \vee Q(x_n)]$$

$$\equiv \exists xP(x) \vee \exists xQ(x)$$

Undecidability of Satisfiability/Validity

- There does not exist any algorithm (no computer program) that decides whether a first-order predicate logic sentence is satisfiable
  - i.e., for input sentence G outputs “Yes” if G is satisfiable and “No” if G is not satisfiable
- Similarly there is no computer program that decides whether a given computer program P halts for a given input number n
  - Called undecidability of the halting problem

Summary of FOL Representation

- What is Predicate Logic?

- Predicate Logic Syntax
  - Constants
  - Variables
  - Predicates
  - Functions
  - Equality
  - Quantifiers
  - Bound vs Free Variables
  - Translating from natural text to FOL

- Predicate Logic Equivalence
  - How does equivalence differ between propositional and FOL?
  - Focus on universal and existential quantifiers

- Predicate Logic Semantics
  - Interpretations and models
  - Satisfiability, validity, and truthfulness
  - How to determine truth in FOL?