Predicate Logic

CS 5012

UNIVERSITY OF VIRGINIA
DATA SCIENCE
INSTITUTE

Brief Review of Propositional Logic

- Since Propositional Logic is part of Predicate Logic, the first few slides will provide a quick review of Propositional Logic

Review!

Review: Propositional Logic

- **Logical constants**: true, false
- **Propositional symbols**: P, Q, R, S, … (*atomic sentences*)
- Wrapping **parentheses**: (..)
- Sentences are combined by **symbolic connectives** (also called *operators*):

$\wedge$	and	[conjunction]
$\vee$	or	[disjunction]
$\rightarrow$	implies	[implication / conditional]
$\leftrightarrow$ $\equiv$	is equivalent	[biconditional]
$\neg$	not	[negation]

Review!

Propositional Logic (2)

Some useful Terms

- **Proposition**: A proposition is a sentence that is either true or false
- **An Argument**: An argument consists of a sequence of statements called *premises* and a statement called a *conclusion*
- **A Valid Argument**: An argument is valid if the conclusion is true whenever the premises are all true
- The **Syntax** of a sentence is its *structure* such as use of connectives and other symbols
- The meaning (**semantics**) of a sentence determines its *interpretation*
- Given the truth values of all symbols in a sentence, it can be "evaluated" to determine its **truth value** (True or False)

Review!

Propositional Logic (3)

Some useful Terms (cont'd)

- A **valid sentence** or **tautology** is a sentence that is *True under all interpretations*, no matter what the world is actually like or what the semantics is. Example: "It's raining or it's not raining"
- An **inconsistent sentence** or **contradiction** is a sentence that is *False under all interpretations*. The world is never like what it describes, as in "It's raining and it's not raining"
- **Zero Order Logic:** It's another name for Propositional Logic

Review!

Propositional Logic (4)

Some useful Terms (cont'd)

- **Truth Values**: If two statements have the *same* truth values, these statements don't have to be *true*. Two statements have the same truth values means "*either both statements are true or both statements are false*"

- **Truth Tables** represent the relationship between the truth values of simple (Atomic) propositions and the compound propositions formed from these simple proposition

Review!

Propositional Logic (5)

Some Truth Tables

		Conjunction	Disjunction	Implication	Negation
P	Q	$P \wedge Q$	$P \vee Q$	$P \rightarrow Q$	$\neg P$
T	T	T	T	T	F
T	F	F	T	F	F
F	T	F	T	T	T
F	F	F	F	T	T

Review!

Review: **Conditional** Statements

- Understanding of conditional statements provides a unique challenge for the students of Logic and Computer Science, a discussion of relevant issues will provide some clarity
- As a first step let us have another look at the **truth table**
- Let there be two atomic statements P and Q

P	Q	P → Q
T	T	T
T	F	F
F	T	T
F	F	T

- Pay special attention to the last column and note that **P→ Q is false only in one case** that is when
  <u>**P is true and Q is false**</u>

Review!

Conditional Statements (2)

- In p→q there does not need to be any connection between P and Q

- To analyze the statements, we just look at the truth values of the antecedent and the consequent. Actual meanings of the statements are not always relevant in this situation

- It is important to note that the conditional statements **do not imply any causation** i.e., that P causes Q

Review!

Conditional Statements (3)

How to grasp and remember these truth values?

- If we consider the conditional statement as a **promise**, an obligation, a contract, a commitment or an order, then remembering the truth table becomes easier.
  - If I become president, then I will reduce the taxes
  - If you pay me 10,000, then I will fix the roof
  - If I whistle, the dog comes
  - If the commander orders, then the soldiers will march

- If the politician was actually elected but did not reduce the taxes, it's a breach of his contract with the voters. The statement he made while campaigning is false. Now, it does not matter, if he did not get elected. (*We can only speculate whether he would have reduced the taxes or not. He may have reduced the taxes. Maybe not…*)

Review!

Conditional Statements (4)

- Let there be two atomic statements P and Q. The truth table for the implication/conditional is:
  (an explanation of each of the resulting truth values are included)

P	**Q**	**P → Q**	
T	T	T	*(certainly true **T**)*
T	F	F	*(Q **false**; a "breach" occurred! **F**)*
F	T	T	*(P **false**; anything can be result **T**)*
F	F	T	*(P **false**; anything can be result **T**)*

UNIVERSITY OF VIRGINIA DATA SCIENCE INSTITUTE

PREDICATE LOGIC

Propositional vs. Predicate Logic (1)

- We spent some time studying Propositional Logic. So what is **predicate Logic**? Is this an alternative way of thinking?
- Shouldn't we be content with one kind of logic?
- Isn’t Propositional logic good?
- **The answer is “It’s good but not good enough”**
- Propositional Logic can handle certain types of statements but not others. Predicate Logic is just an enhanced version of Propositional Logic *to handle a greater variety of real life statements, quantities, relations and functions*

Propositional vs. Predicate Logic (2)

- Propositional logic can handle declarative statements such as “Joe has a blue car”
- It can also handle combinations of such statements as we discussed earlier. However, statements involving formulations such as “**there exists**..." or “**every**..." or “**among**..." are not easy to deal with
- A statement of the form “Every country has some poor citizens” talks about concepts such as country and being poor. These are properties of **elements** of a **set of objects**. We express them in <u>**predicate logic**</u> using **predicates**

What is a Predicate?

- In English, the **predicate** is the part of the sentence that tells you something about the subject
- **Definition of Predicate**: A predicate is a property that a variable or a finite collection of variables can have. A predicate becomes a proposition when specific values are assigned to the variables. $P(x_1, x_2, ..., x_n)$ is called a predicate of n variables or n arguments

  x>100 is a predicate (if x=101 it's true, if x=99 it's false)

  x = 100 is a proposition ("=" means "**is**" … x is fixed)
- In computer programming, predicates are often used to create loops or conditional statements

```
if (x>10) …
do while (x =< 100) …
```

Predicate Logic

Some Definitions and Descriptions

- **Predicate logic** is also called **first order logic**, as well as many other names such as **predicate calculus** and functional calculus
- Generally, predicates are used to describe certain properties or relationships between individuals or objects
- **Quantifiers** indicate how frequently a certain statement is true
- Predicate logic is a generalization of propositional logic

Some definitions and descriptions

- The **universe** of discourse (or *domain*) is the collection of all persons, ideas, symbols, data structures, and so on, that affect the logical argument under consideration. The elements of the domain are called individuals
- The truth of a statement may depend on the **domain** selected. The statement "**there is a smallest number**" is true in the domain of natural numbers, but false in the domain of integers
- The elements of the domain are called **individuals**. An individual can be a person, a number, a data structure, or anything else one wants to reason about

Some definitions and descriptions

- These properties or relations are referred to as **predicates**
- The variables given in the parentheses after the predicate name are called the **arguments**. If May is the mother of Ann, then in **Mother(May, Ann)**, *Mother* is the name of the **predicate** with two arguments
- The number of elements in the argument list of a predicate is called the **arity** of the predicate. In Mother(May, Ann) has arity 2
- A predicate with arity n is often called an **n-place predicate**. A one-place predicate is called a property

Some definitions and descriptions

- A predicate name, followed by an argument list in parentheses is called an **atomic formula**
- Atomic formulas can be *combined by logical connectives* to form new statements:

  Professor(John) → genius(John)
- If all arguments of a predicate *are individual constants*, then the resulting atomic formula must be either **true** or **false**
- If the universe of discourse (domain) is a *collection of things*, **∀x blue(x)** should be read as "*All objects are blue*" and **∃ x blue(x)** should be understood as "*There exist objects that are blue*" or "*Some objects are blue*" or "*There exists at least one object that is blue*"

Use of Predicate Logic for Databases

- In the previous lectures we have discussed the importance of studying logic, it is important to highlight that Predicate logic is extensively used in computer **databases** and the more general notion of "knowledge base", defined to be a database plus various computation rules. In this application, it is common to use predicate expressions containing variables "**queries**". The predicates themselves represent the underlying stored data, computable predicates, or combinations thereof
- A query asks the system to find all individuals corresponding to the variables in the expression *such that the expression is satisfied*

Symbolic elements of Predicate Logic

- Variables:
  - j, k, x, y, …
- Predicates:
  - One argument: female(x) x is a female
  - Two arguments: sister(x,y) x is the sister of y
  - Many arguments: …
- Quantifiers:
  - ∀x "for all x"
  - ∃x "for some x" or "there exists an x such that …"
- Logical Connectives:
  - ¬ ∧ ∨ →
  - Note: As we discussed before, propositional logic is part of predicate logic (a *subset* of predicate logic)

Subjects and Predicates

- In the sentence "The cat is black"
  - The phrase "the cat" denotes the **subject** – the object or entity the sentence is about. The part "is black" denotes the **predicate**, a property that is true of the subject

- In predicate logic, a **predicate** is modeled as a function P(·) from objects to propositions

  P(x) = "x is black" (where x is any object)

Subjects and Predicates (Cont'd)

Conventions:

- Lowercase variables x, y, z... denote **objects/entities**
- Uppercase variables P, Q, R… denote **propositional functions (predicates)**
- Note 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 even number",
P(2) is the proposition "2 is an even number"

Predicates and Propositional Functions

- Predicate logic can 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 studied with y in room z", then if x="Joe", y="Jeff", z="room 29", then P(x,y,z) = "Jeff studied with Joe in room 29"
- Note that "Jeff studied with Joe in room 29" is a **proposition**. We can change the names and the room number and express many such propositions by the predicate P(x,y,z)

Free and Bound Variables

- An expression like P(x) is said to have **a free variable x** (meaning, x is undefined)
- A quantifier (either ∀ or ∃) operates on an expression having free variable(s), and binds one or more of those variables, to produce an expression having **bound variable(s)**
  - P(x,y) has 2 free variables, x and y
  - ∀x P(x,y) has 1 free variable, and one bound variable
  - A variable is **free** in a formula iff it is not quantified. Above, x is bound and y is free
  - "P(x), where x=3" is another way to bind x

Some Examples of Predicate Logic

- Example 1:
  - P is a Predicate to describe a place of residence. It has two arguments (person, residence)
  - P(Jane, Richmond) is a predicate with 2 arguments: Jane and Richmond (x,y): x lives in y
- Example 2:
  - Score(x): x >97 Grade is greater than 97
- Example 3:
  - Dan is a student at UVA
  - We can form two different predicates
  - Let P(x) be “x is a student at UVA”
  - Let Q(x, y) be “x is a student at y”

From propositions to predicates

- A predicate is a generalization of propositional variables

1. R= “It’s raining”, U= “Joe takes an umbrella”, W= “Joe gets wet”
2. R= “It’s raining”, U= “May takes an umbrella”, W= “May gets wet”
3. R= “It’s raining”, U= “Ken takes an umbrella”, W= “Ken gets wet”

- Above, we see **7** separate simple (atomic) proposition

****** Let’s create some relationships symbolically ******

- $R \rightarrow U_{Joe}$, $U_{Joe} \rightarrow \neg W_{Joe}$, $\neg R \rightarrow \neg W_{Joe}$ The subject is Joe
- $R \rightarrow U_{May}$, $U_{Ken} \rightarrow \neg W_{May}$, $\neg R \rightarrow \neg W_{May}$ The subject is May
- $R \rightarrow U_{Ken}$, $U_{Ken} \rightarrow \neg W_{Ken}$, $\neg R \rightarrow \neg W_{Ken}$ The subject is Ken

- Above, we created **9** compound propositions for 3 people. We could go on this way for Fae, Eva, Pat, Tim, Bob, …

(We could go on to create propositions U and W for all the people we know)

Now let us define U to be a predicate that takes the argument x : U(x)
Now let us define W to be a predicate that takes the argument x : W(x)

$R \rightarrow U(x)$, $U(x) \rightarrow \neg W(x)$, $\neg R \rightarrow \neg W(x)$

****** Now we can see the beauty of predicate logic ******

Components of Predicate Logic

- **Objects**, which are things with individual identities
- **Properties** of objects that distinguish them from other objects
- **Relations** that hold among sets of objects
- **Functions**, which are a subset of relations where there is only one “value” for any given “input”

- Examples:
  - **Objects**: Students, lectures, companies, cars, ...
  - **Properties**: blue, oval, even, large, ...
  - **Relations**: Brother-of, bigger-than, outside, part-of, has-color, occurs-after, owns, visits, precedes, ...
  - **Functions**: father-of, best-friend, second-half, one-more-than, ...

Quantifiers

- Predicate Logic is distinguished from Propositional Logic by its use of Quantifiers (Quantified Variables: $\forall$, $\exists$)
- **Universal quantification**
  - $(\forall x)P(x)$ means that P holds for **all** values of x in the domain associated with that variable
  - E.g., $(\forall x)$ dolphin(x) → mammal(x)
- **Existential quantification a**
  - $(\exists\, x)P(x)$ means that P holds for **some** value of x in the domain associated with that variable
  - E.g., $(\exists\, x)$ mammal(x) ∧ lays-eggs(x)
  - Permits one to make a statement about some object without naming it

Illustration of use of Quantifiers (1)

- Let there be a statement **S(x)** of Predicate logic

Use of Universal Quantifier ∀: (∀x) S (x)

- The expression above has the reading that for all (possible values of) x, the predicate S holds of x (*every x has the property S*)
- If S stands for the property of being smart and x ranges over individuals in the world, then the statement has the meaning that for all individuals x in the domain, x is smart
- I.e. if there is even one individual in the domain of S who is not a smart, then the expression would be ***False***
- On the other hand, if it does turn out to be the case that every individual in the domain of S is indeed smart, then the statement would be ***True***

Illustration of use of Quantifiers (2)

- Let there be a statement **S(x)** of Predicate logic

Use of Existential quantification ∃ : (∃ x) S (x)

- The expression above states that there is at least one x in the domain of S such that S(x) is *True*
- Again, if S is a predicate that stands for *smart* and the domain of S is a set of individuals in the world, the statement has the meaning that there is at least one individual in the domain of S who is smart (*at least some x is smart*)
- If there is no x such that x is smart, then the statement is *False*; if there is even just **one** x that is smart, then that is enough to make the above statement to be **True**

Some properties of Quantifiers

- ∀x ∀y is the same as ∀y ∀x
- ∃x ∃y is the same as ∃y ∃x
- ∃x ∀y is not the same as ∀y ∃x
- ∃x ∀y Loves(x,y)
- "There is a person who loves everyone in the world"
- ∀y ∃x Loves(x,y)
- "Everyone in the world is loved by at least one person"

Quantifier duality: each can be expressed using the other

- ∀x Likes(x,IceCream)   ¬∃x ¬Likes(x,IceCream)
- ∃x Likes(x,Broccoli)   ¬∀x ¬Likes(x,Broccoli)

Quantifier Variables (1)

- The variable inside a formula (if there is one) is said to be **bound** by the quantifier that occurs with that variable
- If there is no quantifier, the variable is said to be **free** (E.g. the simple expression P(x) contains the free variable x)
- All open statements contain at least one free variable

Quantifier Variables (2)

- A **variable** is, always, either *bound* or *free*
- A **variable** is associated with **exactly <u>one</u> quantifier**; it cannot be bound by more than one
- Finally, we <u>only</u> talk about binding between quantifiers and variables: individual **constants** (e.g. *joe, mary, a, b*, . . . are not bound or free), they simply do not interact with quantifiers
- An occurrence of a **variable** x is bound if and only if it occurs in the scope of (∃x) or (∀y)
- A **variable** is free if it is *not bound*

Use of Quantifiers

- Universal quantifiers are often used with "implies" to form "rules":
  - **$(\forall x)$ student(x) → smart(x)** means "All students are smart" ("*For all x where x is a student, x is smart*")
- Universal quantification is *rarely* used to make blanket statements about every individual in the world:
  - **$(\forall x)$ student(x) ∧ smart(x)** means "Everyone in the world is a student and is smart"
- Existential quantifiers are usually used with "and" to specify a list of properties about an individual:
  - **$(\exists x)$ student(x) ∧ smart(x)** means "There is a student who is smart"

Switching the Quantifiers

- *Switching the order* of **universal quantifiers** ***does not*** change the meaning:
  - $(\forall x)(\forall y)P(x,y) \leftrightarrow (\forall y)(\forall x)\ P(x,y)$
- Similarly, you *can switch the order* of **existential quantifiers**:
  - $(\exists x)(\exists y)P(x,y) \leftrightarrow (\exists y)(\exists x)\ P(x,y)$

- *Switching the order* of **universals and existentials** ***DOES*** change meaning:
  - Everyone likes someone: $(\forall x)(\exists y)$ likes(x,y)
  - Someone is liked by everyone: $(\exists y)(\forall x)$ likes(x,y)

Connections between All and Exists

- We can relate sentences involving ∀ and ∃ using De Morgan's laws:
  - $(\forall x)\ \neg P(x) \leftrightarrow \neg(\exists x)\ P(x)$
  - $\neg(\forall x)\ P \leftrightarrow (\exists x)\ \neg P(x)$
  - $(\forall x)\ P(x) \leftrightarrow \neg\ (\exists x)\ \neg P(x)$
  - $(\exists x)\ P(x) \leftrightarrow \neg(\forall x)\ \neg P(x)$