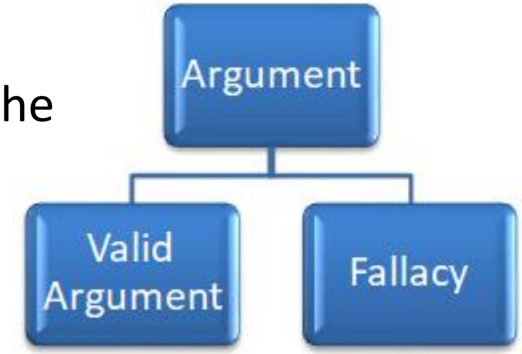
CS 2305: Discrete Mathematics for Computing I

Lecture 05

- KP Bhat

Arguments (Recap)

- An argument is a sequence of statements (premises) that end with a conclusion
- A <u>valid</u> argument is a conclusion that follows from the truth of the premises of the argument
  - an argument is valid if and only if it is impossible for all the premises to be true and the conclusion to be false
    - But no claim that if the conclusion is true then the premises are true as well
- An argument which is not valid is called a <u>fallacy</u>



Establishing Validity of Argument Forms

- Brute force approach
    - truth table
  - Major issue
    - Number of rows in the truth table increases exponentially with the number of propositional variables
- Rules of inference based approach
  - first establish the validity of some relatively simple argument forms (“rules of inference”)
  - use rules of inference as building blocks to validate more complicated valid argument forms

Important Rules of Inference

TABLE 1 Rules of Inference.

Rule of Inference	*Tautology*	*Name*
p $p \to q$ $\therefore q$	$(p \land (p \to q)) \to q$	Modus ponens
$\neg q$ $p \to q$ $\therefore \neg p$	$(\neg q \land (p \to q)) \to \neg p$	Modus tollens
$p \to q$ $q \to r$ $\therefore p \to r$	$((p \to q) \land (q \to r)) \to (p \to r)$	Hypothetical syllogism
$p \lor q$ $\neg p$ $\therefore q$	$((p \lor q) \land \neg p) \to q$	Disjunctive syllogism
p $\therefore p \lor q$	$p \to (p \lor q)$	Addition
$p \land q$ $\therefore p$	$(p \land q) \to p$	Simplification
p q $\therefore p \land q$	$((p) \land (q)) \to (p \land q)$	Conjunction
$p \lor q$ $\neg p \lor r$ $\therefore q \lor r$	$((p \lor q) \land (\neg p \lor r)) \to (q \lor r)$	Resolution

Note: The *Rule of Inference* column of this chart by itself (i.e. without the *Tautology* and *Name* columns) will be provided to the students in the mid-term exam

Modus Ponens

$$\begin{array}{l} p \to q \\ \underline{\;p\;} \\ \therefore q \end{array}$$

Corresponding Tautology:

$$(p \wedge (p \to q)) \to q$$

Example:

Let p be "It is snowing."

Let q be "I will study discrete math."

P_1: "If it is snowing, then I will study discrete math."

P_2: "It is snowing."

∴ "Therefore , I will study discrete math."

Evaluating the Modus Ponens Tautology

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

$$((p \rightarrow q) \land p) \rightarrow q$$

p	q	p → q	(p → q) ∧ p	((p → q) ∧ p) → q
T	T	T	T	T
T	F	F	F	T
F	T	T	F	T
F	F	T	F	T

Tautology

Modus Tollens

$$\begin{array}{l} p \to q \\ \neg q \\ \hline \therefore \neg p \end{array}$$

Corresponding Tautology:

$$(\neg q \wedge (p \to q)) \to \neg p$$

Example:

Let *p* be "it is snowing."

Let *q* be "I will study discrete math."

P_1: "If it is snowing, then I will study discrete math."

P_2: "I will not study discrete math."

∴ "Therefore , it is not snowing."

Hypothetical Syllogism

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

Corresponding Tautology:

$$\left(\left(p \rightarrow q\right) \wedge \left(q \rightarrow r\right)\right) \rightarrow \left(p \rightarrow r\right)$$

Example:

Let *p* be "it snows."

Let *q* be "I will study discrete math."

Let *r* be "I will get an A."

P_1: "If it snows, then I will study discrete math."

P_2: "If I study discrete math, I will get an A."

∴ "Therefore , If it snows, I will get an A."

Disjunctive Syllogism

$$\begin{array}{l} p \vee q \\ \underline{\neg p} \\ \therefore q \end{array}$$

Corresponding Tautology:

$$(\neg p \wedge (p \vee q)) \rightarrow q$$

Example:

Let *p* be "I will study discrete math."

Let *q* be "I will study English literature."

P_1: "I will study discrete math or I will study English literature."

P_2: "I will not study discrete math."

∴ "Therefore , I will study English literature."

Addition

$$\frac{p}{\therefore\ p \vee q}$$

Corresponding Tautology:

$$p \rightarrow (p \vee q)$$

Example:

Let *p* be "I will study discrete math."

Let *q* be "I will visit Las Vegas."

P: "I will study discrete math."

∴ "Therefore, I will study discrete math or I will visit Las Vegas."

Simplification

$$\frac{p \wedge q}{\therefore\ p}$$

Corresponding Tautology:

$$(p \wedge q) \rightarrow p$$

Example:

Let *p* be "I will study discrete math."

Let *q* be "I will study English literature."

P: "I will study discrete math and English literature"

∴ "Therefore, I will study discrete math."

Conjunction

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

Corresponding Tautology:

$$((p) \wedge (q)) \rightarrow (p \wedge q)$$

Example:

Let *p* be “I will study discrete math.”

Let *q* be “I will study English literature.”

P_1: “I will study discrete math.”

P_2: “I will study English literature.”

∴ “Therefore, I will study discrete math and I will study English literature.”

Resolution

$$\begin{array}{rl} & p \vee q \\ & \neg p \vee r \\ \hline \therefore & q \vee r \end{array}$$

Corresponding Tautology:

$$\left(\left(\neg p \vee r\right) \wedge \left(p \vee q\right)\right) \rightarrow \left(q \vee r\right)$$

Example:

Let *p* be "I will study discrete math."

Let *r* be "I will study English literature."

Let q be "I will study databases."

P_1: "I will not study discrete math or I will study English literature."

P_2: "I will study discrete math or I will study databases."

∴ "Therefore, I will study databases or I will study English literature."

Comparing the Tautologies for Logical Equivalences and Rules of Inference

Logical Equivalences

- $\text{Logical\_Expression}_1 \leftrightarrow \text{Logical\_Expresion}_2$ is a tautology

Rules of Inference

- Conjunction_of_premises $\rightarrow$ Conclusion is a tautology

Argument Validation Using Rules of Inference

P_1: It is not sunny this afternoon and it is colder than yesterday
P_2: We will go swimming only if it is sunny this afternoon
P_3: If we do not go swimming, then we will take a canoe trip
P_4: If we take a canoe trip, then we will be home by sunset

∴ We will be home by sunset

Let us use the following propositional variables:
p: It is sunny this afternoon
q: It is colder than yesterday
r: We will go swimming
s: We will take a canoe trip
t: We will be home by sunset

P_1: $\neg p \land q$
P_2: $r \to p$
P_3: $\neg r \to s$
P_4: $s \to t$

$\therefore t$ $\quad ((\neg p \land q) \land (r \to p) \land (\neg r \to s) \land (s \to t)) \to t$

$((\neg p \land q) \land (r \to p) \land (\neg r \to s) \land (s \to t))$
$((\neg p) \land (r \to p) \land (\neg r \to s) \land (s \to t))$ $\quad (p \land q) \to p$ **Simplification**
$(\neg p \land (r \to p) \land (\neg r \to s) \land (s \to t))$
$(\neg r \land (\neg r \to s) \land (s \to t))$ $\quad (\neg q \land (p \to q)) \to \neg p$ **Modus tollens**
$(\neg r \land (\neg r \to s) \land (s \to t))$
$(s \land (s \to t))$ $\quad (p \land (p \to q)) \to q$ **Modus ponens**
$(s \land (s \to t))$
t $\quad (p \land (p \to q)) \to q$ **Modus ponens**

Remember we are establishing inference, **not** logical equivalence

Fallacies

- Fallacy of affirming the conclusion

$p \rightarrow q$
q

$\therefore p$

- Fallacy of denying the hypothesis

$p \rightarrow q$
$\neg p$

$\therefore \neg q$

Predicate Logic
(Predicate Calculus)

Limitations of Propositional Calculus

- There are limitations to argument validation using propositional calculus
- **Example 1**

  P: Every computer connected to the university network is functioning properly

  ----

  $\therefore^{?}$ Computer MATH3 is functioning properly
- **Example 2**

  P: Computer CS2 is under attack by an intruder

  $\therefore^{?}$ There is a computer on the university network that is under attack by an intruder.
- **Example 3**

  P_1: All men are mortal

  P_2: Socrates is a man

  $\therefore^{?}$ Socrates is mortal

Why Predicate Logic?

- Propositional calculus is inadequate to deal with arguments that deal with all cases, or with some case out of many cases
- In such instances, instead of looking at propositions as a whole, we need to understand their inner structure by
  - breaking propositions up into parts {predicates and variables}
  - analyzing quantifiers like “all” or “some”

Predicate Logic / Predicate Calculus

What is Predicate Logic?

- An advanced form of symbolic logic that incorporates not only propositions, and relations between propositions [*and, or, not, if... then*], but also quantifiers [*all, some, many, none, few etc.*]
- Studies the logical form <u>**INSIDE**</u> atomic propositions
- Enables analysis of statements at a higher resolution
- Also known as <u>predicate calculus</u> and <u>first-order logic</u>

Predicate

- A predicate is a part of a declarative sentence (i.e. proposition) that describes the properties of an object or the relationship between objects
  - Generally obtained by removing some or all of the nouns from a proposition
  - In the proposition "17 is a prime number", "is a prime number" is the predicate

Propositional Function (1)

- The generalized form of a proposition, containing one or more predicates, but not targeting any specific subject, is known as propositional function
  - ➢ Specific subjects are replaced by predicate variables
- Example
  - Proposition: James is a student at Bedford College
  - Predicate: is a student at Bedford College
  - Predicate variable: *x*
  - Propositional Function: *x* is a student at Bedford College {represented as P(x)}
- An even more generalized form
  - Proposition: James is a student at Bedford College
  - Predicate: is a student at
  - Predicate variables: *x, y*
  - Propositional Function: *x* is a student at *y* {represented as P(x, y)}

Propositional Functions (2)

- A predicate in a propositional function {say P(x, y)} contains a finite number of variables {in this case “x” and “y”} and becomes a statement when specific values are substituted for the variables
  - Sometimes the terms propositional function and predicate are used synonymously
- The propositional function does not have a truth value of its own
- The set of all values that may be substituted in place of the variable in a predicate is known as the <u>domain</u> of the predicate variable
  - Also known as the <u>domain of discourse</u> or the <u>universe of discourse</u>
- Often the domain is denoted by U

Propositional Functions (3)

- Propositional functions become propositions (and have truth values) when their variables are each replaced by a value from the *domain* (or *bound* by a quantifier, as we will see later).
  - Given a propositional function P(x), it is only when a value is assigned to the variable x that the truth value can be assessed
- The statement *P(x)* is said to be the value of the propositional function *P* at *x*
- For example, let *P(x)* denote "$x > 0$" and the domain be the integers. Then:

  P(-3) is false.

  P(0) is false.

  P(3) is true.

Examples of Propositional Functions (1)

- Let "$x + y = z$" be denoted by $R(x, y, z)$ and U (for all three variables) be the integers. Find these truth values:

  R(2,-1,5)

  Solution: F

  R(3,4,7)

  Solution: T

  $R(x, 3, z)$

  Solution: Not a Proposition

- Now let "$x - y = z$" be denoted by $Q(x, y, z)$, with U as the integers. Find these truth values:

  Q(2,-1,3)

  Solution: T

  Q(3,4,7)

  Solution: F

  $Q(x, 3, z)$

  Solution: Not a Proposition

Examples of Propositional Functions (2)

- Let A(x) denote the statement “Computer x is under attack by an intruder.” Suppose that of the computers on campus, only CS2 and MATH1 are currently under attack by intruders. What are truth values of A(CS1), A(CS2), and A(MATH1)?
  - Proposition A(CS1): Computer CS1 is under attack by an intruder
    - Truth value: False
  - Proposition A(CS2): Computer CS2 is under attack by an intruder
    - Truth value: True
  - Proposition A(MATH1): Computer MATH1 is under attack by an intruder
    - Truth value: True

Compound Expressions

- Connectives from propositional logic carry over to predicate logic.
- If *P(x)* denotes "$x > 0$," find these truth values:

  $P(3) \vee P(-1)$

  Solution: T

  $P(3) \wedge P(-1)$

  Solution: F

  $P(3) \rightarrow P(-1)$

  Solution: F

  $P(3) \rightarrow \neg P(-1)$

  Solution: T
- Expressions with variables are not propositions and therefore do not have truth values. For example,

  $P(3) \wedge P(y)$

  $P(x) \rightarrow P(y)$
- When used with quantifiers (to be introduced next), these expressions (propositional functions) become propositions.

Multi-variable Propositional Functions

- Propositional functions can have multiple variables
  - Let Q(x,y) denote "x = y + 3"
    - Proposition Q(1, 2) is false
    - Proposition Q(3, 0) is true

Quantifiers

- A quantifier specifies the extent to which a predicate is true over a range of elements from the pertinent domain
  – indicates for how many elements from the domain a given predicate is true
- There are two main types of quantifiers in Predicate Logic:

1. Universal Quantifier, “For all,” symbol: $\forall$
   – $\forall x\, P(x)$, read as “for all x P(x)”, asserts P(x) is true for every x in the domain
2. Existential Quantifier, “There exists,” symbol: $\exists$
   – $\exists x\, P(x)$, read as “for some x, P(x)”, asserts P(x) is true for some x in the domain