Formal Logic

CSE2315, Chapter 1

Logic

- Formal logic
  - Definition – foundation for the organized, careful method of thinking that characterizes any reasoned activity
    - A criminal investigation
    - A scientific experiment
    - A sociological study
  - The study of reasoning – specifically concerned with whether something is true or false
  - Formal logic focuses on the relationship between statements as opposed to the content of any particular statement
  - Applications of formal logic in computer science
    - Prolog – programming language based on logic
    - Circuit logic – logic governing computer science

statement

- Statement (proposition)
  - A sentence that is either true or false, but not both
- Truth value of a statement
  - True, T, 1
  - False, F, 0
- Examples:
  - Ten is less than seven
  - Austin is the capital of Texas
  - He is very talented
  - There are life forms on other planets in the universe

Statements and logical connectives

- Statements letters:
  - Capital letters like A, B, C, D, etc are used to represent statements
- Logical connectives:
  - Symbols like $\wedge$, $\vee$, $\rightarrow$,$\leftrightarrow$, $'$
- $\wedge$: and
- $\vee$: or
- $\rightarrow$: implies
- $\leftrightarrow$: equivalent
- $'$: negation

Conjunction and Disjunction

- Connective #1 : Conjunction (Λ)
  - A and B are statements (or statement variables)
  - A Λ B is the conjunction of A and B
  - A and B are called conjuncts of the expression
  - A Λ B is TRUE when both A and B are true
  - A Λ B is FALSE when at least one of A or B is false
- Connective #2 : Disjunction (V)
  - A and B are statements (or statement variables)
  - A V B is the disjunction of A and B
  - A and B are called disjuncts of the expression
  - A V B is TRUE when at least one of A or B is true
  - A V B is FALSE when both A and B are false

Implication

- Connective #3 : Implication ($\rightarrow$)
  - A and B are statements (or statement variables)
  - Symbolic form of "If A, then B" is $A \rightarrow B$
  - A implies B
  - A : hypothesis or antecedent statement
  - B : conclusion or consequent statement
  - $A \rightarrow B$ is FALSE when A is true and B is false
  - $A \rightarrow B$ is TRUE otherwise
  - Example: If I pass the exam, then I will get the job
    - The statement states that I will get the job if a certain condition (passing the exam) is met; it says nothing about what will happen if the condition is not met. If the condition is not met, the truth of the conclusion cannot be determined; by convention, $A \rightarrow B$ is true when A is false, regardless of the truth value of B.

Truth Table (1)

- Truth table is a table that shows the truth values of a statement form which correspond to the different combinations of truth values for the variable

A	B	A ∧ B
T	T	T
T	F	F
F	T	F
F	F	F

A	B	A ∨ B
T	T	T
T	F	T
F	T	T
F	F	F

A	B	A→B
T	T	T
T	F	F
F	T	T
F	F	T

Equivalence

- Connective #4: Equivalence ($\leftrightarrow$)
- $A \leftrightarrow B$ stands for $(A \rightarrow B) \wedge (B \rightarrow A)$
- It is TRUE when both A and B have the same truth values
- It is FALSE when A and B have different truth values

A	B	$A \rightarrow B$	$B \rightarrow A$	$(A \rightarrow B) \wedge (B \rightarrow A)$ / $A \leftrightarrow B$
T	T	T	T	T
T	F	F	T	F
F	T	T	F	F
F	F	T	T	T

Negation

- Connective #5: Negation (’)
- So far, connectives for AND, OR, Implication, and Equivalence were binary connectives, because they join two expressions to produce the third one
- Unary connective
- The negation of A is “not A” and is denoted A’
- It has the opposite truth value from A
  - If A is true, then A’ is false; if A is false, then A’ is true

A	A′
T	F
F	T

Negation

- Examples:
- A: 5 is greater than 2
  - A’: 5 is less than 2
- B: Jane likes butter
  - B’: Jane dislikes butter / hates / doesn’t like
- C: John hates butter but (and) likes cream
  - C’: John likes butter or hates cream
- In a negation, AND becomes OR, OR becomes AND

Another from of implication

- A: You do not do your homework
- A’: You do your homework
- B: You will fail
- A’ V B ??
  - You do your homework or you will fail
- If you do not do your homework, then you will fail
  - $A \rightarrow B$
- Therefore, $A \rightarrow B \equiv A' \vee B$

A	B	$A \rightarrow B$
T	T	T
T	F	F
F	T	T
F	F	T

A	A′	B	A′ V B
T	F	T	T
T	F	F	F
F	T	T	T
F	T	F	T

Tables

- Table 1.5 – Common English words associated with various logical connectives

English Words	Logical Connectives	Logical Expression
And; but; also; in addition; moreover	Conjunction	$A \land B$
Or	Disjunction	$A \lor B$
If A, then B. A implies B A, therefore B A only if B B follows from A A is a sufficient condition for B B is necessary condition for A	Implication	$A \to B$
A if and only if B A is necessary and sufficient for B	Equivalence	$A \leftrightarrow B$
Not A It is false that A … It is not true that A …	Negation	A'

Tables

- Table 1.6 – Examples for the negation of a statement

Statement	Correct Negation	Incorrect Negation
It will rain tomorrow.	It is false that it will rain tomorrow. It will not rain tomorrow.	
Peter is tall and thin.	It is false that Peter is tall and thin. Peter is not tall or he is not thin. Peter is short or fat.	Peter is short and fat.
The river is shallow or polluted.	It is false that the river is shallow or polluted. The river is neither shallow nor polluted. The river is deep and unpolluted.	The river is not shallow or not polluted.

Well Formed Formula (wff)

- An expression that is a legitimate string is called well-formed formula or wff.
- Combining letters, connectives, and parentheses can generate an expression which is meaningful.
  - Example 1: $(A \to B) \vee (B \to A)$ (wff)
  - Example 2: $A\ ))\ \vee B\ (\to C)$ (not a wff)
- To reduce the number of parentheses, an order is stipulated in which the connectives can be applied, called the order of precedence
  1. Connectives within innermost parentheses first and then progress outwards
  2. Negation (′)
  3. Conjunction ($\wedge$), Disjunction ($\vee$)
  4. Implication ($\to$)
  5. Equivalence ($\leftrightarrow$)

Well Formed Formula (wff)

- Hence, $A \vee B \rightarrow C$ is the same as $(A \vee B) \rightarrow C$
- Main connective: the connective to be applied last
  - $A \wedge (B \rightarrow C)'$
  - $\wedge$ is the main connective
- Capital letters, like P,Q,R,S etc. are used to represent wffs
  - $[(A \vee B) \wedge C] \rightarrow (B \vee C')$ can be represented by $P \rightarrow Q$ where
  - P is the wff $[(A \vee B) \wedge C]$ and Q represents $B \vee C'$

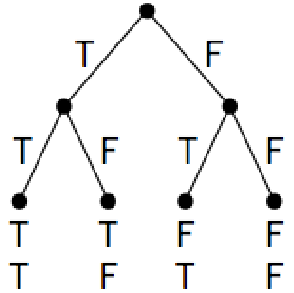
Truth tables for some wffs

- The truth table for the wff A V B′ → (A V B)′ shown below. The main connective, according to the rules of precedence, is implication.

A	B	B′	A V B′	A V B	(A V B)′	A V B′ → (A V B)′
T	T					
T	F					
F	T					
F	F					

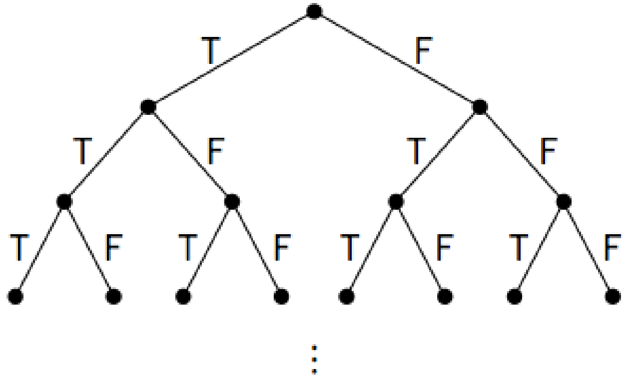
Wff with *n* statement letters

- The total number of rows in a truth table for *n* statement letters is 2^n.



Statement letters	Choices
1	$2 = 2^1$ branches
2	$4 = 2^2$ branches

(a)



Statement letters	Choices
1	$2 = 2^1$ branches
2	$4 = 2^2$ branches
3	$8 = 2^3$ branches
⋮	⋮
n	2^n branches

(b)

Tautology and Contradiction

- Definition of tautology:
- A wff that is intrinsically true, i.e. no matter what the truth value of the statements that comprise the wff.
  - e.g. It will rain today or it will not rain today ($A \vee A'$)
  - $P \leftrightarrow Q$ where P is $A \to B$ and Q is $A' \vee B$

- Definition of a contradiction:
- A wff that is intrinsically false, i.e. no matter what the truth value of the statements that comprise the wff.
  - e.g. It will rain today and it will not rain today ($A \wedge A'$)
  - $(A \wedge B) \wedge A'$

- Usually, tautology is represented by 1 and contradiction by 0

Tautological Equivalences

- Two statement forms are called *logically equivalent* if, and only if, they have identical truth values for each possible substitution of statements for their statement variables.
- The logical equivalence of statement forms P and Q is denoted by writing P ⇔ Q or P ≡ Q. In this case, P and Q are equivalent wffs
- Truth table for (A V B) V C ⇔ A V (B V C)

A	B	C	A V B	B V C	(A V B) V C	A V (B V C)
T	T	T	T	T	T	T
T	T	F	T	T	T	T
T	F	T	T	T	T	T
T	F	F	T	F	T	T
F	T	T	T	T	T	T
F	T	F	T	T	T	T
F	F	T	F	T	T	T
F	F	F	F	F	F	F

Some common equivalences

- The equivalences are listed in pairs, hence they are called duals of each other.
- One equivalence can be obtained from another by replacing $\vee$ with $\wedge$ and 0 with 1 or vice versa.

Commutative	$A \vee B \Leftrightarrow B \vee A$	$A \wedge B \Leftrightarrow B \wedge A$
Associative	$(A \vee B) \vee C \Leftrightarrow A \vee (B \vee C)$	$(A \wedge B) \wedge C \Leftrightarrow A \wedge (B \wedge C)$
Distributive	$A \vee (B \wedge C) \Leftrightarrow (A \vee B) \wedge (A \vee C)$	$A \wedge (B \vee C) \Leftrightarrow (A \wedge B) \vee (A \wedge C)$
Identity	$A \vee 0 \Leftrightarrow A$	$A \wedge 1 \Leftrightarrow A$
Complement	$A \vee A' \Leftrightarrow 1$	$A \wedge A' \Leftrightarrow 0$

De Morgan's Laws

1. $(A \vee B)' \Leftrightarrow A' \wedge B'$
2. $(A \wedge B)' \Leftrightarrow A' \vee B'$

A	B	A'	B'	$A \vee B$	$(A \vee B)'$	$A' \wedge B'$
T	T	F	F	T	F	F
T	F	F	T	T	F	F
F	T	T	F	T	F	F
F	F	T	T	F	T	T

Logical Connectives in the Real World

- Conditional Statements in programming use logical connectives with statements.
- Example

```
if((outflow inflow) and not(pressure 1000))
        do something;
else
        do something else;
```

Algorithm

- Definition of an algorithm:
  - A set of instructions that can be mechanically executed in a finite amount of time in order to solve some problems
- Algorithms are the state in between the verbal form of a problem and the computer program
- Algorithms are usually represented by pseudocode
- Pseudocode should be easy to understand even if you have no idea of programming

Pseudocode example

j = 1   // initial value
Repeat
  read a value for *k*
  if (($j < 5$) AND ($2*j < 10$) OR ($(3*j)^{1/2} > 4$)) **then**
    write the value of *j*
  otherwise
    write the value of $4*j$
  end if statement
  increase *j* by 1
Until $j > 6$

Algorithm example – Tautology Test

TautologyTest(wff P; wff Q)

//Given wffs P and Q, decides whether the wff $P \to Q$ is a tautology.

//Assume $P \to Q$ is not a tautology

P = true // assign T to P

Q = false // assign F to Q

repeat

for each compound wff already assigned a truth value, assign the truth values determined for its components

until all occurrences of statements letters have truth values

if some letter has two truth values

then //contradiction, assumption false

write ("$P \to Q$ is a tautology.")

else //found a way to make $P \to Q$ false

write ("$P \to Q$ is not a tautology.")

end if

end *TautologyTest*