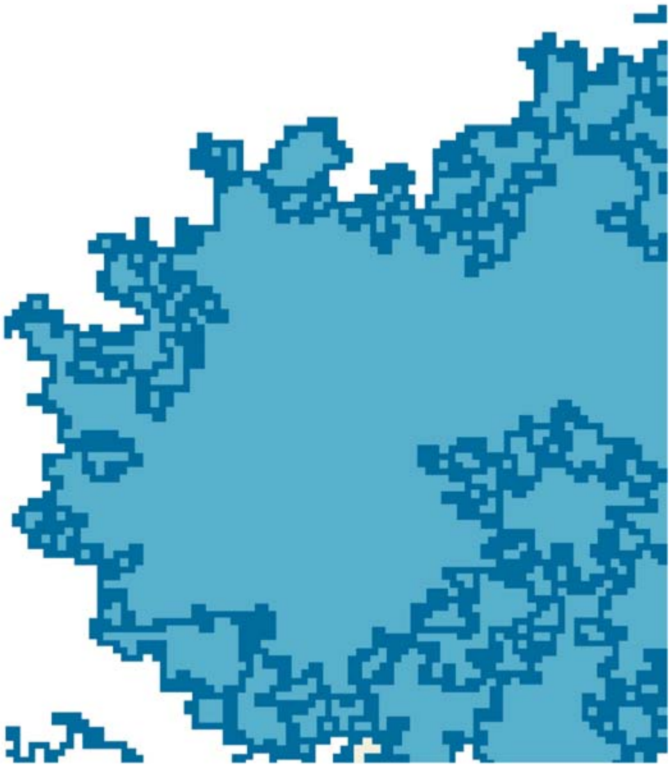
Formal Logic

Mathematical Structures for Computer Science

Chapter 1



Logic: The foundation of reasoning

- What is formal logic? Multiple definitions

Foundation for organized and careful method of thinking that characterizes reasoned activity

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 languages based on logic.
  - Circuit Logic: logic governing computer circuitry.

Statement

- <u>Definition of a statement:</u>

  A *statement*, also called a *proposition*, is a sentence that is either *true* or *false*, but not both.
- Hence the truth value of a statement is T (1) or F (0)
- Examples: Which ones are statements?
  - All mathematicians wear sandals.
  - 5 is greater than –2.
  - Where do you live?
  - You are a cool person.
  - Anyone who wears sandals is an algebraist.

Statements and Logic

- An example to illustrate how logic really helps us (3 statements written below):

  - All mathematicians wear sandals.
  - Anyone who wears sandals is an algebraist.
  - Therefore, all mathematicians are algebraists.

- Logic is of no help in determining the individual truth of these statements.

- However, if the first two statements are true, logic assures the truth of the third statement.

- Logical methods are used in mathematics to prove theorems and in computer science to prove that programs do what they are supposed to do.

Statements and Logical Connectives

- Usually, letters like A, B, C, D, etc. are used to represent statements.
- Logical connectives are symbols such as Λ, V, ↔, →
  - Λ represents *and*
  - → represents *implication*
  - ↔ represents *equivalence*

- A <u>statement form</u> or <u>propositional form</u> is an expression made up of statement variables (such as A and B) and logical connectives (such as Λ, V, ↔, →) that becomes a statement when actual statements are substituted for the component statement variables.
  - Example: (A V A′) → (B Λ B′)

Definitions for Logical Connectives

- Connective # 1: Conjunction (symbol ∧)
  - If A and B are statement variables, the conjunction of A and B is A ∧ B, which is read "A and B".
  - A ∧ B is true when both A and B are true.
  - A ∧ B is false when at least one of A or B is false.
  - A and B are called the conjuncts of A ∧ B.

- Connective # 2: Disjunction (symbol V)
  - If A and B are statement variables, the disjunction of A and B is A V B, which is read "A or B".
  - 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.
  - A and B are called the disjuncts of A V B.

Definitions for Logical Connectives

- Connective # 3: Implication (symbol $\rightarrow$)
- If A and B are statement variables, the symbolic form of "if A then B" is $A \rightarrow B$. This may also be read "A implies B" or "A only if B."Here A is called the hypothesis/antecedent statement and B is called the conclusion/consequent statement.
- "If A then B" is false when A is true and B is false, and it is true otherwise.
- Note: $A \rightarrow B$ is true if A is false, regardless of the truth of B
- Example: If Ms. X passes the exam, then she will get the job
- Here B is *She will get the job* and A is *Ms. X passes the exam.*
  - The statement states that Ms. X 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; the conditional statement is therefore considered to be vacuously true, or true by default.

Another form of *implication*

- Representation of *If-Then* as **Or**
- Let A′ be "You do your homework" and B be "You will flunk."
- The given statement is "Either you do your homework or you will flunk," which is A′ V B.
- In ***if-then*** form, $A \rightarrow B$ means that "If you do not do your homework, then you will flunk," where A (which is equivalent to A″) is "You do not do your homework."
- Hence, $A \rightarrow B \equiv A' \vee B$

A	B	A→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

Definitions for Logical Connectives

- Connective # 4: Equivalence (symbol $\leftrightarrow$)
- If A and B are statement variables, the symbolic form of "A if, and only if, B" and is denoted $A \leftrightarrow B$.
- It is true if both A and B have the same truth values.
- It is false if A and B have opposite truth values.
- The truth table is as follows:
- Note: $A \leftrightarrow B$ is a short form for $(A \rightarrow B) \wedge (B \rightarrow A)$

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

Definitions for Logical Connectives

- Connective #5: Negation (symbol ′)
- If A is a statement variable, 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.
- Example of a negation:
- A: 5 is greater than –2

  A′ : 5 is less than –2
- B: She likes butter

  B′ : She dislikes butter / She hates butter
- A: She hates butter but likes cream / She hates butter and likes cream
- A′ : She likes butter or hates cream
- Hence, in a negation, **and** becomes **or** and vice versa

Truth Tables

- A truth table is a table that displays the truth values of a statement form which correspond to the different combinations of truth values for the variables.

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	A′
T	F
F	T

A	B	A→B
T	T	T
T	F	F
F	T	T
F	F	T

A	B	A↔B
T	T	T
T	F	F
F	T	F
F	F	T

Well Formed Formula (wff)

- Combining letters, connectives, and parentheses can generate an expression which is meaningful, called a wff.
  - e.g. (A → B) V (B → A) is a wff but A)) V B (→ C) is not

- To reduce the number of parentheses, an order is stipulated in which the connectives can be applied, called the order of precedence, which is as follows:
  - Connectives within innermost parentheses first and then progress outwards
  - Negation (′)
  - Conjunction (Λ), Disjunction (V)
  - Implication (→)
  - Equivalence (↔)

  - Hence, A V B → C is the same as (A V B) → C

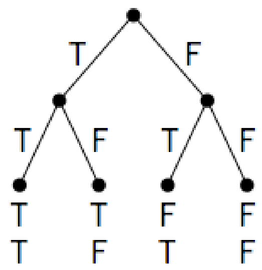
Truth Tables for some wffs

- 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	F	T	T	F	F
T	F	T	T	T	F	F
F	T	F	F	T	F	T
F	F	T	T	F	T	T

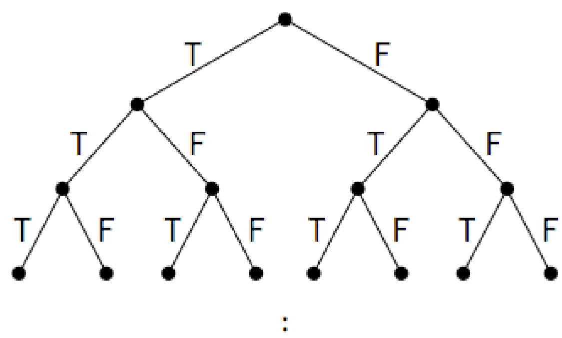
Wff with *n* statement letters

- 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

- Letters like P, Q, R, S etc. are used for representing wffs
  - $[(A \vee B) \wedge C'] \rightarrow A' \vee C$ can be represented by $P \rightarrow Q$ where
  - P is the wff $[(A \vee B) \wedge C']$ and Q represents $A' \vee C$
- <u>Definition of tautology</u>:
- 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 $A \rightarrow B$ and Q is $A' \vee B$
- <u>Definition of a contradiction</u>:
- 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 $P \Leftrightarrow Q$ or $P \equiv Q$.
- Truth table for $(A \vee B) \vee C \Leftrightarrow A \vee (B \vee C)$

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	$A \vee A' \Leftrightarrow 1$	$A \wedge A' \Leftrightarrow 0$

- Prove the distributive property using truth tables.

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 V B	(A V B)′	A′ Λ 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

- 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 a problem unambiguously.

- Algorithms are the stage 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 $j > 6$

Tautology Test Algorithm

- This algorithm applies only when the main connective is Implication ($\rightarrow$)

TautologyTest(wff P; wff Q)

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

//Assume $P \rightarrow Q$ 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 ("$P \rightarrow Q$ is a tautology.")

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

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

end if

end *TautologyTest*