Jiangxi University of Science and Technology

Discrete Mathematics and Its Applications

Logic Module (Part 1)

Acknowledgement

- Most of these slides are adapted from ones created by Professor Bart Selman at Cornell University and Dr Johnnie Baker

Chapter Summary

- Propositional Logic
  - The Language of Propositions
  - Applications
  - Logical Equivalences
- Predicate Logic
  - The Language of Quantifiers
  - Logical Equivalences
  - Nested Quantifiers
- Proofs
  - Rules of Inference
  - Proof Methods
  - Proof Strategy

Propositional Logic Summary

- The Language of Propositions
  - Connectives
  - Truth Values
  - Truth Tables
- Applications
  - Translating English Sentences
  - System Specifications
  - Logic Puzzles
  - Logic Circuits
- Logical Equivalences
  - Important Equivalences
  - Showing Equivalence
  - Satisfiability

<u>Note</u>: The order that these slides cover the material in the textbook is not always exactly the same as the textbook order, although the order is roughly the same.

Logic in general

- Logics are formal languages for formalizing reasoning, in particular for representing information such that conclusions can be drawn
- A logic involves:
  - A language with a syntax for specifying what is a legal expression in the language;
  - syntax defines well formed sentences in the language
  - Semantics for associating elements of the language with elements of some subject matter.
    - Semantics defines the "meaning" of sentences (link to the world); i.e., semantics defines the truth of a sentence with respect to each possible world
  - Inference rules for manipulating sentences in the language

Original motivation: Early Greeks settled arguments based on purely rigorous (symbolic/syntactic) reasoning starting from a given set of premises.

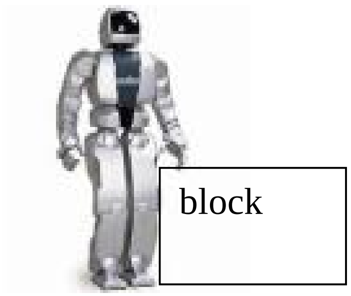
Example of a formal language: Arithmetic

- E.g., the language of arithmetic
  - $x+2 \geq y$ is a sentence;
  - $2x+y > \{\}$ is not a sentence
  - $x+2 \geq y$ is true iff the number $x+2$ is no less than the number y
  - $x+2 \geq y$ is true in a world where $x = 7, y = 1$
  - $x+2 \geq y$ is false in a world where $x = 0, y = 6$

Simple Robot Domain

- Consider a robot that is able to lift a block,
  - if that block is liftable (i.e., not too heavy), and
  - if the robot’s battery power is adequate.

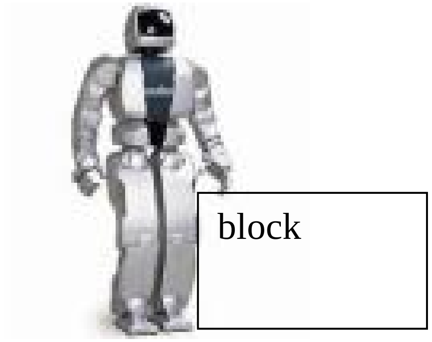
•If both of these conditions are satisfied, then when the robot tries to lift a block it is holding, its arm n



Feature 1: BatIsOk (True or False)
Feature 2: BlockLiftable (True or False)
Feature 3: RobotMoves (True or False)

Simple Robot Domain

We need a *language* to express the *features/properties/assertions* and *constraints among them*; also *inference mechanisms*, *i.e*, principled ways of performing reasoning.



Example - logical statement about the robot:

(BatIsOk and BlockLiftable) implies RobotMoves

Binary valued featured descriptions

- Consider the following description:
  - The router can send packets to the edge system only if it supports the new address space. For the router to support the new address space it is necessary that the latest software release be installed. The router can send packets to the edge system if the latest software release is installed. The router does not support the new address space.
  - Features:
    - Router
      - Feature 1 – router can send packets to the edge of system
      - Feature 2 – router supports the new address space
    - Latest software release
      - Feature 3 – latest software release is installed

Binary valued featured descriptions

- Constraints:
  - The router can send packets to the edge system only if it supports the new address space. (constraint between feature 1 and feature 2);
  - It is necessary that the latest software release be installed for the router to support the new address space . (constraint between feature 2 and feature 3);
  - The router can send packets to the edge system if the latest software release is installed. (constraint between feature 1 and feature 3);

- How can we write these specifications in a formal language and reason about the system?

1.1 Propositional Logic
Syntax: Elements of the language

Primitive propositions --- statements like:

Bob loves Alice $\longrightarrow$ P

Alice loves Bob $\longrightarrow$ Q

Propositional Symbols
(atomic propositions)

Compound propositions

Bob loves Alice *and* Alice loves Bob

$\longrightarrow$ $P \wedge Q$ ($\wedge$ - stands for and)

1.1 Propositional Logic
Connectives

- $\neg$ - not
- $\wedge$ - and
- $\vee$ - or
- $\rightarrow$ - implies
- $\leftrightarrow$ - equivalent (if and only if)

Compound Propositions: Negation

– The *negation* of a proposition p is denoted by $\neg p$ and has this truth table:

p	$\neg p$
T	F
F	T

- **Example**: If p denotes "The earth is round.", then $\neg p$ denotes "It is not the case that the earth is round," or more simply "The earth is not round."

Conjunction

- The *conjunction* of propositions p and q is denoted by $p \wedge q$ and has this truth table:

p	q	$p \wedge q$
T	T	T
T	F	F
F	T	F
F	F	F

- **Example**: If p denotes "I am at home." and q denotes "It is raining." then $p \wedge q$ denotes "I am at home and it is raining."

Disjunction

- The *disjunction* of propositions p and q is denoted by $p \vee q$ and has this truth table:

p	q	$p \vee q$
T	T	T
T	F	T
F	T	T
F	F	F

- **Example**: If p denotes "I am at home." and q denotes "It is raining." then $p \vee q$ denotes "I am at home or it is raining."

1.1 Propositional Logic
Syntax

- **Syntax of Well Formed Formulas (wffs) or sentences**
  - Atomic sentences are wffs:
    - Examples: P, Q, R, BlockIsRed; SeasonIsWinter;
  - Complex or compound wffs examples, assuming that w1 and w2 are wwfs:
    - $\neg$ w1 (negation)
    - (w1 $\wedge$ w2) (conjunction)
    - (w1 $\vee$ w2) (disjunction)
    - (w1 $\rightarrow$ w2) (implication; w1 is the antecedent; w2 is the consequent)
    - (w1 $\leftrightarrow$ w2) (biconditional)

1.1 Propositional Logic
Propositional logic: Examples

Additional Examples of wffs

- $P \land Q$
- $(P \lor Q) \to R$
- $P \lor Q \to P$
- $(P \to Q) \to (\neg Q \to \neg P)$
- $\neg\neg P$

Comments:

- Atoms or negated atoms are called ***literals***;
  - Examples: p and $\neg$p are literals.
- $P \land Q$ is a compound statement or compound proposition.
- Parentheses are important to ensure that the syntax is unambiguous. Quite often parentheses are omitted;
- The order of precedence in propositional logic is (from highest to lowest): $\neg$,$\land$, $\lor$, $\to$, $\leftrightarrow$

1.1 Propositional Logic

TABLE 8
Precedence of Logical Operators.

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

- $\neg$ - not
- $\wedge$ - and
- $\vee$ - or
- $\rightarrow$ - implies
- $\leftrightarrow$ - equivalent (if and only if)

Implication

- If p and q are propositions, then $p \rightarrow q$ is a *conditional statement* or *implication* which is read as "if p, then q" and has this truth table:

p	q	$p \rightarrow q$
T	T	T
T	F	F
F	T	T
F	F	T

- **Example**: If p denotes "I am at home." and q denotes "It is raining." then $p \rightarrow q$ denotes "If I am at home then it is raining."
- In $p \rightarrow q$, p is the *hypothesis* (*antecedent* or *premise*) and q is the *conclusion* (or *consequence*).

Understanding Implication

- In $p \rightarrow q$ there does not need to be any connection between the antecedent or the consequent. The "meaning" of $p \rightarrow q$ depends only on the truth values of p and q.

- These implications are perfectly fine, but would not be used in ordinary English.
  - "If the moon is made of green cheese(新鲜乳酪), then I have more money than Bill Gates. "
  - "If the moon is made of green cheese then I'm on welfare(接收救济)."
  - "If 1 + 1 = 3, then your grandma wears combat boots(短筒靴)."

Understanding Implication (cont)

- One way to view the logical conditional is to think of an obligation or contract.
  - “If I am elected, then I will lower taxes.”
  - “If you get 100% on the final, then you will get an A.”
- If the politician is elected and does not lower taxes, then the voters can say that he or she has broken the campaign pledge. Something similar holds for the professor. This corresponds to the case where p is true and q is false.

Different Ways of Expressing $p \to q$

- if p, then q
- if p, q
- q unless $\neg p$
- q if p
- q whenever p
- q follows from p
- p implies q
- p only if q
- q when p
- q when p
- p is sufficient for q
- q is necessary for p

- a necessary condition for p is q
- a sufficient condition for q is p

1.1 Propositional Logic

Propositional Logic:Syntax vs. Semantics

- Syntax involves whether notation is correctly formed
- Semantics has to do with “meaning”:
  - it associates the elements of a logical language with the elements of a domain of discourse.

- Propositional Logic – involves associating *atoms* with *propositions* or *assertions* about the world (therefore called “propositional logic”).

1.1 Propositional Logic
Truth Assignment to Propositions

- Interpretation or Truth Assignment:
- In an application, a truth assignment (True or False) must be made to each proposition.
- So if for **n** atomic propositions, there are $\mathbf{2^n}$ truth assignments or interpretations.
- This makes the representation powerful: the propositions implicitly capture $\mathbf{2^n}$ possible states of the world.

1.1 Propositional Logic
Sematics Example

- We might associate the atom (just a symbol!) BlockIsRed with the proposition: “The block is Red”,
- However, we could also associate it with the proposition “The block is Black” even though this would be quite confusing…
- BlockIsRed has value True just in the case the block is red; otherwise BlockIsRed is False.
  - Computers manipulate symbols. The string “BlockIsRed” does not “mean” anything to the computer.
  - Meaning has to come from how to come from relations to other symbols and the “external world”. Hmm

1.1 Propositional Logic
Sematics Example (cont.)

- How can a computer / robot obtain the meaning ``The block is Red’’?
- The fact that computers only “push around symbols” led to quite a bit of confusion in the early days or Artificial Intelligence, Robotics, and natural language understanding.

1.1 Propositional Logic
Propositions Review

- Which ones are propositions?
  - Cornell University is in Ithaca NY
  - 1 + 1 = 2
  - what time is it?
  - 2 + 3 = 10
  - watch your step!

1.1 Propositional Logic
Propositions Review

- What is the negation of the proposition “At least ten inches of rain fell today in Miami”?

1.1 Propositional Logic
Propositions Review

- What is the negation of the proposition “At least 1o inches of rain fell today in Miami”?
  - It is not the case that at least 10 inches of rain fell today in Miami
  - (Simpler) Less than 10 inches of rain fell today in Miami.

Biconditional

- If p and q are propositions, then we can form the *biconditional* proposition $p \leftrightarrow q$, read as "p if and only if q." The biconditional $p \leftrightarrow q$ denotes the proposition with this truth table:

p	q	$p \leftrightarrow q$
T	T	T
T	F	F
F	T	F
F	F	T

- If p denotes "I am at home." and q denotes "It is raining." then $p \leftrightarrow q$ denotes "I am at home if and only if it is raining."

Expressing the Biconditional

- Some alternative ways "p if and only if q" is expressed in English:
  - p **is necessary and sufficient for** q
  - **if** p **then** q **, and conversely**
  - p **iff** q

Truth Tables For Compound Propositions

Construction of a truth table:

Rows

- Need a row for every possible combination of values for the atomic propositions.

Columns

- Need a column for the compound proposition (usually at far right)
- Need a column for the truth value of each expression that occurs in the compound proposition as it is built up.
  - This includes the atomic propositions

1.1 Propositional Logic
Propositional Logic:Semantics

Truth table for connectives

Given the values of atoms under some interpretation, we can use a truth table to compute the value
for any wff under that same interpretation; the truth table establishes the semantics (meaning) of the propositional connectives.

P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \Rightarrow Q$ $\rightarrow$	$P \Leftrightarrow Q$ $\leftrightarrow$
$false$	$false$	$true$	$false$	$false$	$true$	$true$
$false$	$true$	$true$	$false$	$true$	$true$	$false$
$true$	$false$	$false$	$false$	$true$	$false$	$false$
$true$	$true$	$false$	$true$	$true$	$true$	$true$

We can use the truth table to compute the value of any wff given the values of the constituent atom in the wff. Note: In table, P and Q can be compound propositions themselves.
<u>Note</u>: Implication is not necessarily aligned with English usage.

1.1 Propositional Logic

Implication ($p \rightarrow q$)

- This is only False (violated) when q is False and p is True.
- Related implications:
  - Converse: $q \rightarrow p$;
  - Contra-positive: $\neg q \rightarrow \neg p$;
  - Inverse $\neg p \rightarrow \neg q$;

Important: only the contra-positive of $p \rightarrow q$ is equivalent to $p \rightarrow q$ (i.e., has the same truth values in all models); the converse and the inverse are equivalent;

1.1 Propositional Logic

Implication ($p \rightarrow q$)

- Implication plays an important role in reasoning. A variety of terminologies are used to refer to implication:

•conditional statement
•if p then q
•if p, q
•p is sufficient for q
•q if p
•q when p
•a necessary condition for p is q (*)

•p implies q
•p only if q (*)
•a sufficient condition for q is p
•q whenever p
•q is necessary for p (*)
•q follows from p

Note: the mathematical concept of implication is independent of a cause and effect relationship between the hypothesis (p) and the conclusion (q), that is normally present when we use implication in English.

Note: Focus on the case, when is the statement False. That is, p is True and q is False, should be the only case that makes the statement false.

(*) assuming the statement true, for p to be true, q has to be true

1.1 Propositional Logic
Implication Questions

- Let p be the statement "Maria learns discrete mathematics" and q the statement "Maria will find a good job". Express $p \rightarrow q$ as a statement in English.
- You can access the internet from campus only if you are a computer science major or you are not a freshman

1.1 Propositional Logic
Implication Question (cont.)

- Question:
- Let p be the statement "Maria learns discrete mathematics" and q the statement "Maria will find a good job". Express $p \to q$ as a statement in English.
- Solution: Any of the following.
- If Maria learns discrete mathematics, then she will find a good job.
- Maria will find a good job when she learns discrete mathematics
- For Maria to get a good job, it is sufficient for her to learn discrete mathematics.

1.1 Propositional Logic
Second Conditional Question

- You can access the internet from campus only if you are a computer science major or you are not a freshman.
- Solution:
- Let a, c and f represent “you can access the Internet from campus” , “you are a computer science major”, and “you are a freshman”.
- Then above statement can be stated more simply as “You can access the internet implies that you are a computer science major major or you are not a freshman
- $a \rightarrow (c \vee \neg f)$

1.1 Propositional Logic
Bi-Conditionals ($p \leftrightarrow q$)

- Variety of terminology :
  - p is necessary and sufficient for q
  - if p then q, and conversely
  - p if and only if q
  - p iff q

$p \leftrightarrow q$ is equivalent to $(p \rightarrow q) \wedge (q \rightarrow p)$

Note: the if and only if construction used in biconditionals is rarely used in common language;

Example: "if you finish your meal, then you can play;" really means: "If you finish your meal, then you can play" and "You can play, only if you finish your meal".

1.1 Propositional Logic

TABLE 6 The Truth Table for the Biconditional $p \leftrightarrow q$.

p	q	$p \leftrightarrow q$
T	T	T
T	F	F
F	T	F
F	F	T

江西理工大学
JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY

1.1 Propositional Logic
Exclusive Or

- Truth Table

P	Q	$P \oplus Q$
T	T	F
T	F	T
F	T	T
F	F	F

$P \oplus Q$ is equivalent to $(P \wedge \neg Q) \vee (\neg P \wedge Q)$
and also equivalent to $\neg (P \leftrightarrow Q)$

Use a truth table to check these equivalences.

1.1 Propositional Logic

Propositional Logic:Satisfiability and Models

Satisfiability and Models

- **An interpretation or truth assignment** ***satisfies*** **a wff, if the wff is assigned the value** ***True, under that interpretation*****.**

- **An interpretation that satisfies a wff is called a** ***model*** **of that wff.**

Given an interpretation (i.e., the truth values for the n atoms) then one can use the truth table to find the value of any wff.

1.2 Propositional Equivalences:
Inconsistency (Unsatisfiability) and Validity

- **Inconsistent or Unsatisfiable set of Wffs**
  - **It is possible that no interpretation satisifies a set of wffs**
  - **In that case we say that the set of wffs is *inconsistent* or *unsatisfiable* or a *contradiction***

Examples:

1 – $\{P \wedge \neg P\}$

2 – $\{ P \vee Q,\ P \vee \neg Q,\ \neg P \vee Q,\ \neg P \vee \neg Q\}$

(use the truth table to confirm that this set of wffs is inconsistent)

- **Validity (Tautology) of a set of Wffs**

- **If a wff is True under all the interpretations of its constituents atoms, we say that the wff is *valid or it is a tautology*.**

Examples: 1- $P \rightarrow P$; 2 - $\neg(P \wedge \neg P)$; 3 - $[P \rightarrow (Q \rightarrow P)]$; 4- $[((P \rightarrow Q) \rightarrow P) \rightarrow P]$

江西理工大学
JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY

1.2 Propositional Equivalences:
Showing a Set of wwfs are Inconsistent

Consider { $P \vee Q$, $P \vee \neg Q$, $\neg P \vee Q$, $\neg P \vee \neg Q$}

Must show that the following wwf is unsatisfiable

$$(P \vee Q) \wedge (P \vee \neg Q) \wedge (\neg P \vee Q) \wedge (\neg P \vee \neg Q)$$

List the following 11terms in your truth table in following order:

$\underline{P}$ $\underline{Q}$ $\underline{\neg P}$ $\underline{\neg Q}$ $\underline{(P \vee Q)}$ $\underline{(P \vee \neg Q)}$ $\underline{(P \vee Q) \wedge (P \vee \neg Q)}$

$\underline{(\neg P \vee Q)}$ $\underline{(\neg P \vee \neg Q)}$ $\underline{(\neg P \vee Q) \wedge (\neg P \vee \neg Q)}$

$\underline{(P \vee Q) \wedge (P \vee \neg Q) \wedge (\neg P \vee Q) \wedge (\neg P \vee \neg Q)}$

1.2 Propositional Equivalences:

Logical equivalence

- Two sentences **p** an **q** are logically equivalent ($\equiv$ or $\Leftrightarrow$) iff p $\leftrightarrow$ q is a tautology
- (and therefore p and q have the same truth value for all truth assignments)

$$
\begin{aligned}
(\alpha \wedge \beta) &\equiv (\beta \wedge \alpha) && \text{commutativity of } \wedge \\
(\alpha \vee \beta) &\equiv (\beta \vee \alpha) && \text{commutativity of } \vee \\
((\alpha \wedge \beta) \wedge \gamma) &\equiv (\alpha \wedge (\beta \wedge \gamma)) && \text{associativity of } \wedge \\
((\alpha \vee \beta) \vee \gamma) &\equiv (\alpha \vee (\beta \vee \gamma)) && \text{associativity of } \vee \\
\neg(\neg\alpha) &\equiv \alpha && \text{double-negation elimination} \\
(\alpha \Rightarrow \beta) &\equiv (\neg\beta \Rightarrow \neg\alpha) && \text{contraposition} \\
(\alpha \Rightarrow \beta) &\equiv (\neg\alpha \vee \beta) && \text{implication elimination} \\
(\alpha \Leftrightarrow \beta) &\equiv ((\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)) && \text{biconditional elimination} \\
\neg(\alpha \wedge \beta) &\equiv (\neg\alpha \vee \neg\beta) && \text{de Morgan} \\
\neg(\alpha \vee \beta) &\equiv (\neg\alpha \wedge \neg\beta) && \text{de Morgan} \\
(\alpha \wedge (\beta \vee \gamma)) &\equiv ((\alpha \wedge \beta) \vee (\alpha \wedge \gamma)) && \text{distributivity of } \wedge \text{ over } \vee \\
(\alpha \vee (\beta \wedge \gamma)) &\equiv ((\alpha \vee \beta) \wedge (\alpha \vee \gamma)) && \text{distributivity of } \vee \text{ over } \wedge
\end{aligned}
$$

Note: logical equivalence (or iff) allows us to make statements about PL, pretty much like we use = in in ordinary mathematics.

The truth table method

TABLE 5 A Demonstration That $p \vee (q \wedge r)$ and $(p \vee q) \wedge (p \vee r)$ Are Logically Equivalent.

p	q	r	$q \wedge r$	$p \vee (q \wedge r)$	$p \vee q$	$p \vee r$	$(p \vee q) \wedge (p \vee r)$
T	T	T	T	T	T	T	T
T	T	F	F	T	T	T	T
T	F	T	F	T	T	T	T
T	F	F	F	T	T	T	T
F	T	T	T	T	T	T	T
F	T	F	F	F	T	F	F
F	F	T	F	F	F	T	F
F	F	F	F	F	F	F	F

(Propositional) logic has a "truth composnal semantics":
Meaning is built up from the meaning of its primitive parts (just like English text). itio

1.2 Propositional Equivalences:
Truth Tables

Truth table for connectives

We can use the truth table to compute the value of any wff given the values of the constituent atom
in the wff.

Example:
Suppose P and Q are False and R has value True.
Given this interpretation, what is the truth value of [(P $\rightarrow$ Q) $\rightarrow$ R] $\rightarrow$ P? False

If a system is described using *n features* (corresponding to *propositions*), and these features are represented by a corresponding set of *n atoms*, then there are 2^n different ways the system can be. Why? Each of the ways the system can be corresponds to an interpretation. Therefore there are 2^n interpretations.

1.2 Propositional Equivalences: Logic and Bit Operations

- Computers represent information using bits.
- A bit has only two possible values, namely 0 and 1.
- A 1 represents T (true) and 0 represents F (false)
- A variable is called a boolean variable if its value is either true or false.
- By replacing true by 1 and false by 0, a computer can perform logical operations.
- These replacements provides the following table for bit operators.

x	y	$x \vee y$	$x \wedge y$	$x \oplus y$
0	0	0	0	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	0

1.2 Propositional Equivalences:
Example: Binary valued featured descriptions

- Consider the following description:
  - The router can send packets to the edge system only if it supports the new address space. For the router to support the new address space it is necessary that the latest software release be installed. The router can send packets to the edge system if the latest software release is installed. The router does not support the new address space.
  - Features:
    - Router
      - P - router can send packets to the edge of system
      - Q - router supports the new address space
    - Latest software release
      - R – latest software release is installed

Formal:

- The router can send packets to the edge system only if it supports the new address space. (constraint between feature 1 and feature 2)
  - If Feature 1 (P) (router can send packets to the edge of system) then Feature 2 (Q) (router supports the new address space) $\mathbf{P \rightarrow Q}$

- For the router to support the new address space it is necessary that the latest software release be installed. (constraint between feature 2 and feature 3);
  - If Feature 2 (Q) (router supports the new address space) then Feature 3 (R) (latest software release is installed) $\mathbf{Q \rightarrow R}$

- The router can send packets to the edge system if the latest software release is installed. (constraint between feature 1 and feature 3);

  If Feature 3 (R) (latest software release is installed) then Feature 1 (P) (router can send packets to the edge of system) $\mathbf{R \rightarrow P}$

- The router does not support the new address space. $\mathbf{\neg Q}$

江西理工大学
JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY

The END

Jiangxi University of Science and Technology

Applications of Propositional Logic

Section 1.2

Applications of Propositional Logic: Summary

- Translating English to Propositional Logic
- System Specifications
- Boolean Searching
- Logic Puzzles
- Logic Circuits
- AI Diagnosis Method (Optional)

Translating English Sentences

- Steps to convert an English sentence to a statement in propositional logic
  - Identify atomic propositions and represent using propositional variables.
  - Determine appropriate logical connectives
- "If I go to Harry's or to the country, I will not go shopping."
  - *p*: I go to Harry's
  - q: I go to the country.
  - *r*: I will go shopping.

If *p* or *q* then not *r*.

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

Example

Problem: Translate the following sentence into propositional logic:

"You can access the Internet from campus only if you are a computer science major or you are not a freshman."

One Solution: Let *a*, *c*, and *f* represent respectively "You can access the internet from campus," "You are a computer science major," and "You are a freshman."

$$a \rightarrow (c \vee \neg f)$$

System Specifications

- System and Software engineers take requirements in English and express them in a precise specification language based on logic.

Example: Express in propositional logic:

"The automated reply cannot be sent when the file system is full"

Solution: One possible solution: Let p denote "The automated reply can be sent" and q denote "The file system is full."

$$q \rightarrow \neg p$$

Consistent System Specifications

Definition: A list of propositions is *consistent* if it is possible to assign truth values to the proposition variables so that each proposition is true.

Exercise: Are these specifications consistent?

- "The diagnostic message is stored in the buffer or it is retransmitted."
- "The diagnostic message is not stored in the buffer."
- "If the diagnostic message is stored in the buffer, then it is retransmitted."

Solution: Let p denote "The diagnostic message is not stored in the buffer." Let q denote "The diagnostic message is retransmitted" The specification can be written as: $p \vee q$, $p \rightarrow q$, $\neg p$. When p is false and q is true all three statements are true. So the specification is consistent.

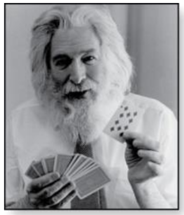
- What if "The diagnostic message is not retransmitted is added."

  Solution: Now we are adding $\neg q$ and there is no satisfying assignment. So the specification is not consistent.

江西理工大學

HANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY

Logic Puzzles



Raymond Smullyan
(Born 1919)

- An island has two kinds of inhabitants, *knights*, who always tell the truth, and *knaves*, who always lie.
- You go to the island and meet A and B.
  - A says "B is a knight."
  - B says "The two of us are of opposite types."

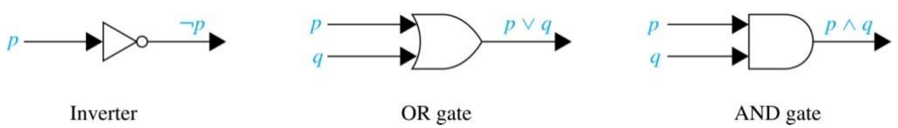
Example: What are the types of A and B?

Solution: Let p and q be the statements that A is a knight and B is a knight, respectively. So, then $\neg p$ represents the proposition that A is a knave and $\neg q$ that B is a knave.

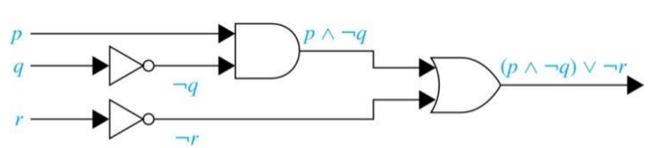
- If A is a knight, then p is true. Since knights tell the truth, q must also be true. Then $(p \land \neg q) \lor (\neg p \land q)$ would have to be true, but it is not. So, A is not a knight and therefore $\neg p$ must be true.
- If A is a knave, then B must not be a knight since knaves always lie. So, then both $\neg p$ and $\neg q$ hold since both are knaves.

Logic Circuits
(Studied in depth in Chapter 12)

- Electronic circuits; each input/output signal can be viewed as a 0 or 1.
  - 0   represents **False**
  - 1   represents **True**
- Complicated circuits are constructed from three basic circuits called gates.



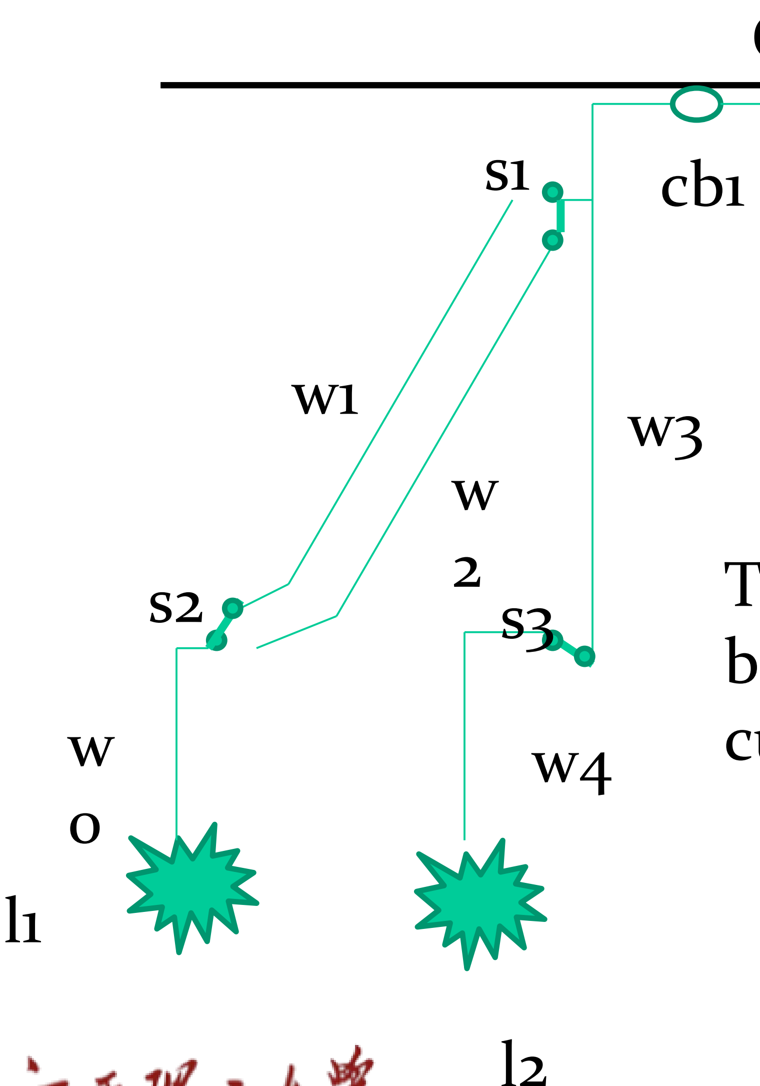
  - The inverter (**NOT gate**)takes an input bit and produces the negation of that bit.
  - The **OR gate** takes two input bits and produces the value equivalent to the disjunction of the two bits.
  - The **AND gate** takes two input bits and produces the value equivalent to the conjunction of the two bits.
- More complicated digital circuits can be constructed by combining these basic circuits to produce the desired output given the input signals by building a circuit for each piece of the output expression and then combining them. For example:



Diagnosis of Faults in an Electrical System (*Optional*)

- AI Example (from *Artificial Intelligence: Foundations of Computational Agents* by David Poole and Alan Mackworth, 2010)
- Need to represent in propositional logic the features of a piece of machinery or circuitry that are required for the operation to produce observable features. This is called the **Knowledge Base (KB)**.
- We also have observations representing the features that the system is exhibiting now.

Electrical System Diagram (optional)



Have lights (l1, l2), wires (w0, w1, w2, w3, w4), switches (s1, s2, s3), and circuit breakers (cb1)

The next page gives the knowledge base describing the circuit and the current observations.

Representing the Electrical System in Propositional Logic

- We need to represent our common-sense understanding of how the electrical system works in propositional logic.
- For example: “If l1 is a light and if l1 is receiving current, then l1 is lit.
  - $lit\_l1 \rightarrow light\_l1 \wedge live\_l1 \wedge ok\_l1$
- Also: “If w1 has current, and switch s2 is in the up position, and s2 is not broken, then w0 has current.”
  - $live\_w0 \rightarrow live\_w1 \wedge up\_s2 \wedge ok\_s2$
- This task of representing a piece of our common-sense world in logic is a common one in logic-based AI.

Knowledge Base (*opt*)

- live_outside
- light_l1
- light_l2
- live_l1 → live_w0
- live_w0 → live_w1 ∧ up_s2 ∧ ok_s2
- live_w0 → live_w2 ∧ down_s2 ∧ ok_s2
- live_w1 → live_w3 ∧ up_s1 ∧ ok_s1
- live_w2 → live_w3 ∧ down_s1 ∧ ok_s1
- live_l2 → live_w4
- live_w4 → live_w3 ∧ up_s3 ∧ ok_s3
- live_w3 → live_outside ∧ ok_cb1
- lit_l1 → light_l1 ∧ live_l1 ∧ ok_l1
- lit_l2 → light_l2 ∧ live_l2 ∧ ok_l2

We have outside power.

Both l1 and l2 are lights.

If s2 is ok and s2 is in a down position and w2 has current, then w0 has current.

Observations (*opt*)

- Observations need to be added to the KB
  - Both Switches up
    - up_s1
    - up_s2
  - Both lights are dark
    - ¬lit_l1
    - ¬ lit_l2

Diagnosis (*opt*)

- We assume that the components are working ok, unless we are forced to assume otherwise. These atoms are called *assumables*.
- The assumables (ok_cb1, ok_s1, ok_s2, ok_s3, ok_l1, ok_l2) represent the assumption that we assume that the switches, lights, and circuit breakers are ok.
- If the system is working correctly (all assumables are true), the observations and the knowledge base are consistent (i.e., satisfiable).
- The augmented knowledge base is clearly not consistent if the assumables are all true. The switches are both up, but the lights are not lit. Some of the assumables must then be false. This is the basis for the method to diagnose possible faults in the system.
- A diagnosis is a minimal set of assumables which must be false to explain the observations of the system.