Logic

Logic deals with the formalization of natural language and reasoning methods.

A variety of logical systems have been developed, including

propositional logic,
predicate logic,
temporal logics, and
modal logics.

Typical applications of logic in computing include

logic programming,
automated verification, and
reasoning about knowledge

We will begin with an introduction to propositional logic.

From the Files of Inspector Craig

The following facts are known about a robbery:

1. If A is guilty and B is innocent, then C is guilty.
2. C never works alone.
3. A never works with C.
4. No one other than A, B, or C was involved, and at least one of them is guilty.

Can one infer from these facts who is guilty and who is innocent?

The Case of McGregor's Shop

Mr. McGregor phoned Scotland Yard that his shop had been robbed. Three suspects A, B, C were rounded up for questioning and the following facts were established:

1. Each of A, B, C had been in the shop on the day of the robbery, and no one else had been in the shop that day.
2. If A was guilty, then he had exactly one accomplice.
3. If B is innocent, so is C.
4. If exactly two are guilty, then A is one of them.
5. If C is innocent, so is B.

Whom did Inspector Craig indict?

For other cases see *What Is the Name of This Book?* by Raymond Smullyan.

Fundamental Notions in Logic

The above examples can be formalized in *propositional logic*, a system based on well-known logical connectives, such as negation, conjunction, disjunction, and implication. (Most applications of logic to computing require richer logical languages with additional, more specialized logical operators.)

Some of the key questions in the study of logical systems are:

When is a given logical formula true? (*Validity*)

Do given assumptions logically imply a given formula? (*Logical consequence*)

How can we deduce a desired conclusion from given axioms? (*Provability*)

The relationship between the concepts of *truth* and *proof* within specified a logical system often plays a central role.

Propositional Logic

Propositional logic is a formal system in which the basic units are *propositions*. These represent statements and can be combined via *logical connectives* into more complex propositions.

The basic assumption is that

> *each proposition is either true or false* (but not both).

Simple propositions are denoted by *(propositional) variables* or by constants representing true and false.

The connectives used to form more complex propositions include negation (¬, read "not"), conjunction (∧, read "and"), disjunction (∨, read "or"), and implication (→, read "implies").

Syntax of Propositional Logic

The syntax of propositional formulas is specified by the following rules:

$$
\begin{array}{lcl}
\langle\text{proposition}\rangle & ::= & \top \mid \bot \mid \langle\text{variable}\rangle \\
& & \mid (\neg\langle\text{proposition}\rangle) \\
& & \mid (\langle\text{proposition}\rangle \wedge \langle\text{proposition}\rangle) \\
& & \mid (\langle\text{proposition}\rangle \vee \langle\text{proposition}\rangle) \\
& & \mid (\langle\text{proposition}\rangle \rightarrow \langle\text{proposition}\rangle) \\
\langle\text{variable}\rangle & ::= & P \mid Q \mid R \mid \ldots
\end{array}
$$

We use the letters α and β to denote propositional formulas.

Other common connectives are exclusive disjunction (⊕, read "either-or") and biconditional (↔, read "if and only if").

Parentheses are often omitted to increase readability (provided the intended expression remains unambiguous).

The notion of a *subformula* can be defined in the expected way. For example, P and $(P \rightarrow Q)$ are both subformulas of $(\neg(P \rightarrow Q))$.

Semantics of Propositional Logic

The constants ⊤ and ⊥ are also called *truth values* and represent *truth* and *falsity*, respectively.

The semantics of propositional logic formulas rests on so-called *truth functions* for the logical connectives.

Traditionally truth functions are given by way of truth tables, though they can also be defined by suitable identities:

$$
\begin{array}{lcl}
\neg\top & \approx & \bot \\
\neg\bot & \approx & \top
\end{array}
\qquad
\begin{array}{lcl}
\top\rightarrow\top & \approx & \top \\
\top\rightarrow\bot & \approx & \bot \\
\bot\rightarrow\top & \approx & \top \\
\bot\rightarrow\bot & \approx & \top
\end{array}
$$

$$
\begin{array}{lcl}
\top\wedge\top & \approx & \top \\
\top\wedge\bot & \approx & \bot \\
\bot\wedge\top & \approx & \bot \\
\bot\wedge\bot & \approx & \bot
\end{array}
\qquad
\begin{array}{lcl}
\top\vee\top & \approx & \top \\
\top\vee\bot & \approx & \top \\
\bot\vee\top & \approx & \top \\
\bot\vee\bot & \approx & \bot
\end{array}
$$

Formally, these identities define an equivalence relation on propositional formulas. We define:

$$\alpha \approx \beta$$

if, and only if, β can be obtained from α by repeatedly using the above identities to replace a subformula matching one side by the other side.

For example,

$$(\top \vee \bot) \rightarrow \bot \approx \top \rightarrow \bot \approx \bot.$$

If a propositional formula contains no variables it is equivalent either to ⊤ or ⊥ (but not both).

One of Inspector Craig's Cases

The known facts can be represented by the formulas

1. $P \wedge \neg Q \rightarrow R$
2. $R \rightarrow P \vee Q$
3. $P \rightarrow \neg R$
4. $P \vee Q \vee R$

where P represents the statement "A is guilty," Q the statement "B is guilty," and R the statement "C is guilty."

Let α be the conjunction of the above four formulas. We get the following truth table for α:

P	Q	R	α
$\bot$	$\bot$	$\bot$	$\bot$
$\bot$	$\bot$	$\top$	$\bot$
$\bot$	$\top$	$\bot$	$\top$
$\bot$	$\top$	$\top$	$\top$
$\top$	$\bot$	$\bot$	$\bot$
$\top$	$\bot$	$\top$	$\bot$
$\top$	$\top$	$\bot$	$\top$
$\top$	$\top$	$\top$	$\bot$

Truth Valuations

A (truth) *valuation* is a mapping from propositiona) variables to truth values.

It is usually sufficient to consider truth valuations with a *finite* domain; various notations are used to denote such mappings, e.g.,

$$[P \mapsto \top, Q \mapsto \top, R \mapsto \bot]$$

or

$$[\top/P, \top/Q, \bot/R].$$

If $\alpha = \alpha(P_1, \ldots, P_n)$ is a formula containing variables $P_1, \ldots, P_n$, and σ is a truth valuation, then by $\alpha\sigma$ we denote the result of replacing in α each occurrence of a variable P_i by its truth value, as specified by σ.

Propositional Equivalence

Two propositional formulas α and β are said to be *(propositionally) equivalent*, written $\alpha \sim \beta$, if and only if $\alpha\sigma \approx \beta\sigma$, for all truth valuations σ whose domain includes all variables occurring in α or β.

Basically, one may check equivalence of propositional formulas by inspecting truth tables.

Examples.

$$\begin{array}{rcl} \neg P \rightarrow Q & \sim & P \vee Q \\ P \wedge \neg P & \sim & \neg(P \rightarrow P) \\ P \rightarrow P & \sim & \top \end{array}$$

Note that $\sim$ extends $\approx$ in the sense that for all variable-free propositional formulas α and β, we have $\alpha \sim \beta$ if, and only if, $\alpha \approx \beta$.

Some Basic Equivalences

$$\begin{array}{rcl} P \vee P & \sim & P \\ P \wedge P & \sim & P \\ P \vee Q & \sim & Q \vee P \\ P \wedge Q & \sim & Q \wedge P \\ P \wedge (P \vee Q) & \sim & P \\ P \wedge (Q \vee R) & \sim & (P \wedge Q) \vee (P \wedge R) \\ P \vee \neg P & \sim & \top \\ P \wedge \neg P & \sim & \bot \\ \neg\neg P & \sim & P \\ P \vee \top & \sim & \top \\ P \wedge \top & \sim & P \\ P \vee \bot & \sim & P \\ P \wedge \bot & \sim & \bot \\ \neg(P \vee Q) & \sim & \neg P \wedge \neg Q \\ \neg(P \wedge Q) & \sim & \neg P \vee \neg Q \\ P \rightarrow Q & \sim & \neg P \vee Q \end{array}$$

Substitution

Valuations are a special kind of substitutions. In general, by a *(propositional) substitution* we mean a mapping from (propositional) variables to (propositional) formulas.

We will use the letters σ and τ to denote substitutions, and write $\alpha\sigma$ to denote the result of applying the substitution σ to the formula α.

Note that applying a substitution means to *simultaneously* replace all occurrences of variables by the indicated formulas.

For example, if

$$\sigma = [P \mapsto P \wedge Q, Q \mapsto \neg R]$$

then $((P \vee Q) \rightarrow P)\sigma$ is $((P \wedge Q) \vee \neg R) \rightarrow (P \wedge Q)$.

Substitution Theorem.

> For all propositional formulas α and β and propositional substitutions σ, if $\alpha \sim \beta$, then $\alpha\sigma \sim \beta\sigma$.

Replacement

We write $\alpha[\beta]$ to indicate that β occurs as a subformula of α, and (ambiguously) denote by $\alpha[\beta']$ the result of *replacing* a particular occurrence of β in α by β'.

If necessary, one indicates the occurrence by writing $\alpha[\beta]_p$, where p specifies the position of the subformula, e.g., in Dewey decimal notation. (The subformula of α at position p is often denoted by $\alpha|_p$.)

For example, if α is $(P \wedge Q) \vee R$, then $\alpha_{1.2}$ is Q and $\alpha[P]_{1.2}$ is $(P \wedge P) \vee R$.

Replacement Theorem.

> If α, β and β' are propositional formulas with $\beta \sim \beta'$, and p is a position in α, then $\alpha[\beta]_p \sim \alpha[\beta']_p$.

Tautologies and Contradictions

A propositional formula α is said to be *satisfiable* if $\alpha\sigma \approx \top$, for some truth valuation σ.

A propositional formula α is called a *tautology* if it always evaluates to true, i.e., if $\alpha\sigma \approx \top$ for every truth valuation σ whose domain contains all variables occurring in α.

Similarly, α is called a *contradiction* (or *unsatisfiable*) if it always evaluates to false.

A formula that is neither a tautology nor a contradiction is also called a *contingency*.

For example, $P \vee \neg P$ is a tautology, whereas $P \wedge \neg P$ is a contradiction.

Theorem. [Tautology and contradiction]

> A propositional formula α is a tautology if and only if its negation $\neg\alpha$ is a contradiction.

Theorem. [Tautology and equivalence]

> Two propositional formulas α and β are logically equivalent if and only if the formula $(\alpha \rightarrow \beta| \wedge (\beta \rightarrow \alpha)$ is a tautology.

Logical Consequence

A (propositional) formula α is called a *logical consequence* of a set of formulas N, written

$$N \models \alpha$$

if α is true for every valuation σ under which each formula in N is true.

For instance, α is a logical consequence of a finite set $N = \{\alpha_1, \ldots, \alpha_n\}$ if

$$\alpha\sigma \approx \top \text{ whenever } \alpha_1\sigma \approx \cdots \approx \alpha_n\sigma \approx \top.$$

Theorem [Tautology and logical consequence]

> A formula α is a logical consequence of $\alpha_1, \ldots, \alpha_n$ if, and only if, the implication
>
> $$\alpha_1 \wedge \cdots \wedge \alpha_n \rightarrow \alpha$$
>
> is a tautology.

We call a set of formulas N *satisfiable* if there is a valuation σ that makes each formula α in N true. A set of formulas is *unsatisfiable* if it is not satisfiable.

Theorem [Logical consequence and unsatisfiability]

> A propositional formula α is a logical consequence of a set of propositional formulas N if, and only if the set $N \cup \{\neg\alpha\}$ is unsatisfiable.

An Example

Consider the following propositional formulas:

$$\begin{array}{rcll} A\wedge B\wedge C & \rightarrow & D & (1)\\ \neg A\wedge M & \rightarrow & L & (2)\\ F\wedge E & \rightarrow & \neg D & (3)\\ G\wedge M & \rightarrow & C & (4)\\ \neg B\wedge F & \rightarrow & \neg H & (5)\\ \neg D\wedge B\wedge E & \rightarrow & G & (6)\\ M\wedge\neg I & \rightarrow & J & (7)\\ H\wedge M & \rightarrow & K & (8)\\ K\wedge J\wedge\neg L & \rightarrow & E & (9)\\ \neg H\wedge F & \rightarrow & L & (10)\\ M\wedge L & \rightarrow & \neg F & (11)\\ K\wedge I\wedge A & \rightarrow & E & (12) \end{array}$$

Is the implication

$$M \rightarrow \neg F \qquad (13)$$

a logical consequence of these formulas?

Checking validity via the corresponding truth table is possible, but rather time consuming, considering that the table has $2^{13} = 8192$ rows.

Example (cont.)

Let us apply Quine's method to the formula

$$\alpha = (1)\wedge\cdots\wedge(12)\rightarrow(13)$$

where $(1),\ldots,(13)$ refer to the formulas above.

Let us select the variable M. It can easily be seen that the formula $\alpha[M\mapsto\bot]$ is equivalent to $\top$ (as $\bot\rightarrow\neg F$ is equivalent to $\top$).

The formula $\alpha[M\mapsto\top]$ can be simplified by eliminating all occurrences of the constant $\top$. Specifically, $\alpha[M\mapsto\top]$ is equivalent to

$$\alpha_1 = (14)\wedge\cdots\wedge(25)\rightarrow(26)$$

where the subformulas are:

$$\begin{array}{rcll} A\wedge B\wedge C & \rightarrow & D & (14)\\ \neg A & \rightarrow & L & (15)\\ F\wedge E & \rightarrow & \neg D & (16)\\ G & \rightarrow & C & (17)\\ \neg B\wedge F & \rightarrow & \neg H & (18)\\ \neg D\wedge B\wedge E & \rightarrow & G & (19)\\ \neg I & \rightarrow & J & (20)\\ H & \rightarrow & K & (21)\\ K\wedge J\wedge\neg L & \rightarrow & E & (22)\\ \neg H\wedge F & \rightarrow & L & (23)\\ L & \rightarrow & \neg F & (24)\\ K\wedge I\wedge A & \rightarrow & E & (25)\\ & & \neg F & (26) \end{array}$$

Quine's Method

The following method can be used to determine whether a given propositional formula is a tautology, a contradiction, or a contingency.

Let α be a propositional formula.

- If α contains no variables, it can be simplified to $\top$ or $\bot$, and hence is either a tautology or a contradiction.

- If α contains a variable, then (i) select a variable, say P, (ii) simplify both $\alpha[P\mapsto\top]$ and $\alpha[P\mapsto\bot]$, denoting the simplified formulas by α_1 and α_2, respectively, and (iii) apply the method recursively to α_1 and α_2.

  If α_1 and α_2 are both tautologies, so is α. If α_1 and α_2 are both contradictions, so is α. Otherwise α is a contingency.

Example (cont.)

Next we select the variable F. The formula $\alpha_1[F\mapsto\bot]$ is obviously equivalent to $\top$.

The formula $\alpha[F\mapsto\top]$ can be simplified as follows:

$$\alpha_2 = (27)\wedge\cdots\wedge(38)\rightarrow\bot$$

where the subformulas are:

$$\begin{array}{rcll} A\wedge B\wedge C & \rightarrow & D & (27)\\ \neg A & \rightarrow & L & (28)\\ E & \rightarrow & \neg D & (29)\\ G & \rightarrow & C & (30)\\ \neg B & \rightarrow & \neg H & (31)\\ \neg D\wedge B\wedge E & \rightarrow & G & (32)\\ \neg I & \rightarrow & J & (33)\\ H & \rightarrow & K & (34)\\ K\wedge J\wedge\neg L & \rightarrow & E & (35)\\ \neg H & \rightarrow & L & (36)\\ L & \rightarrow & \bot & (37)\\ K\wedge I\wedge A & \rightarrow & E & (38) \end{array}$$

Example (cont.)

We continue with selecting the variable L. Note that the formula $\alpha_2[L \mapsto \top]$ is equivalent to $\top$, whereas $\alpha_2[L \mapsto \bot]$ can be simplified as follows:

$$\alpha_3 = (39) \wedge \cdots \wedge (49) \rightarrow \bot$$

where the subformulas are:

$$A \wedge B \wedge C \rightarrow D \quad (39)$$
$$\neg A \rightarrow \bot \quad (40)$$
$$E \rightarrow \neg D \quad (41)$$
$$G \rightarrow C \quad (42)$$
$$\neg B \rightarrow \neg H \quad (43)$$
$$\neg D \wedge B \wedge E \rightarrow G \quad (44)$$
$$\neg I \rightarrow J \quad (45)$$
$$H \rightarrow K \quad (46)$$
$$K \wedge J \rightarrow E \quad (47)$$
$$\neg H \rightarrow \bot \quad (48)$$
$$K \wedge I \wedge A \rightarrow E \quad (49)$$

Example (cont.)

We pick variable A next. The formula $\alpha_3[A \mapsto \bot]$ is equivalent to $\top$, whereas $\alpha_3[A \mapsto \top]$ can be simplified to

$$\alpha_4 = (50) \wedge \cdots \wedge (59) \rightarrow \bot$$

where the subformulas are:

$$B \wedge C \rightarrow D \quad (50)$$
$$E \rightarrow \neg D \quad (51)$$
$$G \rightarrow C \quad (52)$$
$$\neg B \rightarrow \neg H \quad (53)$$
$$\neg D \wedge B \wedge E \rightarrow G \quad (54)$$
$$\neg I \rightarrow J \quad (55)$$
$$H \rightarrow K \quad (56)$$
$$K \wedge J \rightarrow E \quad (57)$$
$$\neg H \rightarrow \bot \quad (58)$$
$$K \wedge I \rightarrow E \quad (59)$$

Example (cont.)

The variable H is a suitable next choice. The formula $\alpha_4[H \mapsto \bot]$ is equivalent to $\top$, whereas $\alpha_4[H \mapsto \top]$ can be simplified to

$$\alpha_5 = (60) \wedge \cdots \wedge (68) \rightarrow \bot$$

where the subformulas are:

$$B \wedge C \rightarrow D \quad (60)$$
$$E \rightarrow \neg D \quad (61)$$
$$G \rightarrow C \quad (62)$$
$$\neg B \rightarrow \bot \quad (63)$$
$$\neg D \wedge B \wedge E \rightarrow G \quad (64)$$
$$\neg I \rightarrow J \quad (65)$$
$$K \quad (66)$$
$$K \wedge J \rightarrow E \quad (67)$$
$$K \wedge I \rightarrow E \quad (68)$$

Example (cont.)

The next variable we select is K. The formula $\alpha_5[K \mapsto \bot]$ is equivalent to $\top$, whereas $\alpha_5[K \mapsto \top]$ can be simplified to

$$\alpha_6 = (69) \wedge \cdots \wedge (76) \rightarrow \bot$$

where the subformulas are:

$$B \wedge C \rightarrow D \quad (69)$$
$$E \rightarrow \neg D \quad (70)$$
$$G \rightarrow C \quad (71)$$
$$\neg B \rightarrow \bot \quad (72)$$
$$\neg D \wedge B \wedge E \rightarrow G \quad (73)$$
$$\neg I \rightarrow J \quad (74)$$
$$J \rightarrow E \quad (75)$$
$$I \rightarrow E \quad (76)$$

Example (cont.)

Continuing with variable B, we find that $\alpha_6[B \mapsto \bot]$ is equivalent to $\top$, whereas $\alpha_6[B \mapsto \top]$ can be simplified to

$$\alpha_7 = (77) \wedge \cdots \wedge (83) \rightarrow \bot$$

where the subformulas are:

$$\begin{array}{rcl} C & \rightarrow & D \quad (77) \\ E & \rightarrow & \neg D \quad (78) \\ G & \rightarrow & C \quad (79) \\ \neg D \wedge E & \rightarrow & G \quad (80) \\ \neg I & \rightarrow & J \quad (81) \\ J & \rightarrow & E \quad (82) \\ I & \rightarrow & E \quad (83) \end{array}$$

Example (cont.)

In the next step we select the variable E. we find that $\alpha_7[E \mapsto \bot]$ can be simplified to

$$\alpha_8 = (84) \wedge \cdots \wedge (89) \rightarrow \bot$$

where the subformulas are:

$$\begin{array}{rcl} C & \rightarrow & D \quad (84) \\ G & \rightarrow & C \quad (85) \\ \neg D \wedge E & \rightarrow & G \quad (86) \\ \neg I & \rightarrow & J \quad (87) \\ J & \rightarrow & \bot \quad (88) \\ I & \rightarrow & \bot \quad (89) \end{array}$$

The formula $\alpha_7[E \mapsto \top]$ can be simplified to

$$\alpha_9 = (90) \wedge \cdots \wedge (94) \rightarrow \bot$$

where the subformulas are:

$$\begin{array}{rcl} C & \rightarrow & D \quad (90) \\ & & \neg D \quad (91) \\ G & \rightarrow & C \quad (92) \\ \neg D & \rightarrow & G \quad (93) \\ \neg I & \rightarrow & J \quad (94) \end{array}$$

Example (cont.)

Continuing with α_8 we select the variable I and find that $\alpha_8[I \mapsto \top]$ is equivalent to $\top$, whereas $\alpha_8[I \mapsto \bot]$ can be simplified to

$$\alpha_{10} = (95) \wedge \cdots \wedge (99) \rightarrow \bot$$

where the subformulas are:

$$\begin{array}{rcl} C & \rightarrow & D \quad (95) \\ G & \rightarrow & C \quad (96) \\ \neg D \wedge E & \rightarrow & G \quad (97) \\ & & J \quad (98) \\ J & \rightarrow & \bot \quad (99) \end{array}$$

If next we select J, we find that both $\alpha_{10}[I \mapsto \top]$ and $\alpha_{10}[I \mapsto \bot]$ are equivalent to $\top$.

Example (cont.)

Coming back to α_9, we select the variable D. The formula $\alpha_9[D \mapsto \top]$ is equivalent to $\top$, whereas $\alpha_9[D \mapsto \bot]$ can be simplified to

$$\alpha_{11} = (100) \wedge \cdots \wedge (103) \rightarrow \bot$$

where the subformulas are:

$$\begin{array}{rcl} C & \rightarrow & \bot \quad (100) \\ G & \rightarrow & C \quad (101) \\ & & G \quad (102) \\ \neg I & \rightarrow & J \quad (103) \end{array}$$

The formula $\alpha_{11}[G \mapsto \bot]$ is equivalent to $\top$, whereas $\alpha_{11}[G \mapsto \top]$ can be simplified to

$$\alpha_{12} = \neg C \wedge C \wedge (\neg I \rightarrow J) \rightarrow \bot.$$

Both $\alpha_{12}[C \mapsto \bot]$ and $\alpha_{12}[C \mapsto \top]$ are equivalent to $\top$.

At this point we are done: All simplified formulas are equivalent to $\top$, which implies that the original formula α is a tautology.