4CCS1ELA – Elementary Logic with Applications

1 – Introduction to Propositional Logic

Dr. Odinaldo Rodrigues

odinaldo.rodrigues@kcl.ac.uk

Room S1.26, +44 (0)20 7848 2087

Department of Informatics

King's College London

Outline

BUILDING BLOCKS

Defining a formal language

Building block 1: propositions

A *proposition* is a declarative statement that is either **true** or **false**, but **not both**.

- Each one of these states will be called the *truth-value* of the proposition.
- The truth-value 'truth' can be written as T (Rosen); as *true* (Hein); or as 1 (we will oscillate between T and 1)
- The truth-value 'falsehood' can be written F (Rosen); as *false* (Hein); or as 0 (we will oscillate between F and 0)

Examples

1. Can John come with you?
   This is a question, and hence not a proposition.
2. Bachelors are unmarried men.
   This is a proposition: it can be true or false.
3. Tom shut the door.
   This is a proposition: it can be true or false.
4. Tom, shut the door!
   This is a command, and hence not a proposition.
5. The sky is blue.
   This is a proposition: it can be true or false.

Defining a formal language

Building block 2: language connectives

A *connective* is a symbol that is used to modify a statement or to combine two statements to make more complex statements.

The simplest statement is a proposition.
A unary connective applies to one statement.
A binary connective applies to two statements.

Connectives:

not (unary)	if . . . , then . . . (binary)
and (binary)	if and only if . . . (binary)
or (binary)	

Examples

1. Bachelors are unmarried men **and** Tom shut the door.

   This is *not* a proposition, but it is a well-formed complex statement.

2. Tom shut the door **not** the sky is blue.

   This is not a proposition, nor a well-formed complex statement, since "not" is unary and there is no binary connective connecting the first proposition with the second complex statement.

Example of a well-formed version of the last statement

We can make a complex statement with the propositions in the last example, by using a binary connective to bind them. With the connective **and** this gives us:

Tom shut the door **and not** the sky is blue.

In English, we use the connective **not** after the verb, so the above is actually uttered as:

Tom shut the door **and** the sky is **not** blue.

Replacing symbols for propositions

Proposition statements quickly become difficult to write, so instead we associate them with *propositional symbols* of a pre-defined *alphabet*.

We could, for instance, associate propositions and symbols as follows:

Proposition	Alphabet 1	Alphabet 2	Alphabet 3
Bachelors are unmarried men	*U*	α	$\clubsuit$
Tom shut the door	*S*	β	$\spadesuit$
The sky is blue	*B*	γ	$\heartsuit$

The choice of symbols is irrelevant, but it must remain the same throughout. We use "convenient" symbols.

Defining propositional logical connectives

As done for the propositions, we associate special symbols with the English connectives.

The usual association is as follows

not:	$\neg$	if X, then Y:	$X \rightarrow Y$
and:	$\wedge$	X if and only if Y:	$X \leftrightarrow Y$
or:	$\vee$		

The propositional symbols and connectives define a *formal language*.

The collection of all correctly constructed statements of the language constitutes the language's *well-formed formulae*.

FORMAL LANGUAGE

Propositional language

- An infinite set of *propositional symbols* (or *propositional variables*) represented by (possibly subscripted) letters such as

$$P, \; Q, \; R, \; \ldots, P_1, \; Q_1, \; \ldots$$

- The logical connectives:
  $\neg$ (negation), $\wedge$ (conjunction), $\vee$ (disjunction),
  $\rightarrow$ (implication), $\leftrightarrow$ (equivalence)
- The auxiliary symbols: '(' and ')'.

Informally, $\neg$ is read as "not", $\wedge$ as "and", $\vee$ as "or", $\rightarrow$ as "if . . ., then . . . ", $\leftrightarrow$ as "if and only if", as before.

Propositional language: formulae

The set of all *well-formed formulae* (wff) is defined inductively as follows:

- All propositional symbols are (*atomic*) formulae;
- If A is a formula, then $\neg A$ is a formula;
- If A and B are formulae, then
  $(A \wedge B)$, $(A \vee B)$, $(A \rightarrow B)$, $(A \leftrightarrow B)$
  are formulae.
- Nothing else is a formula.

Formulae are just (well-formed) strings over a certain alphabet.
We will always omit the outer brackets of a formula.

TRANSLATING

Representing English statements

Let the symbol P represent the proposition "Bob likes Mary", and Q represent the proposition "Bob likes Sue".

Then:

- "Bob likes Mary and Bob likes Sue" is represented as $P \wedge Q$.
- "Bob likes Mary or Bob likes Sue" is represented as $P \vee Q$.
- $\neg Q$ represents "Bob does not like Sue".
- "If Bob likes Mary, then Bob does not like Sue" is represented as $P \rightarrow \neg Q$.
- "Bob likes Mary if, and only if, Bob does not like Sue" is represented as $P \leftrightarrow \neg Q$.

Parentheses and the priority of connectives

How should we represent the following statement?

"If Bob is rich then Sue is happy and Jim is happy."

Natural language is ambiguous, whereas formal language is not. To disambiguate, we use parentheses.

Translation one: $B \rightarrow (S \wedge J)$
Translation two: $(B \rightarrow S) \wedge J$

We can avoid parentheses if we assume a precedence order. We will assume the following from highest to lowest: $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$

Parentheses and the priority of connectives

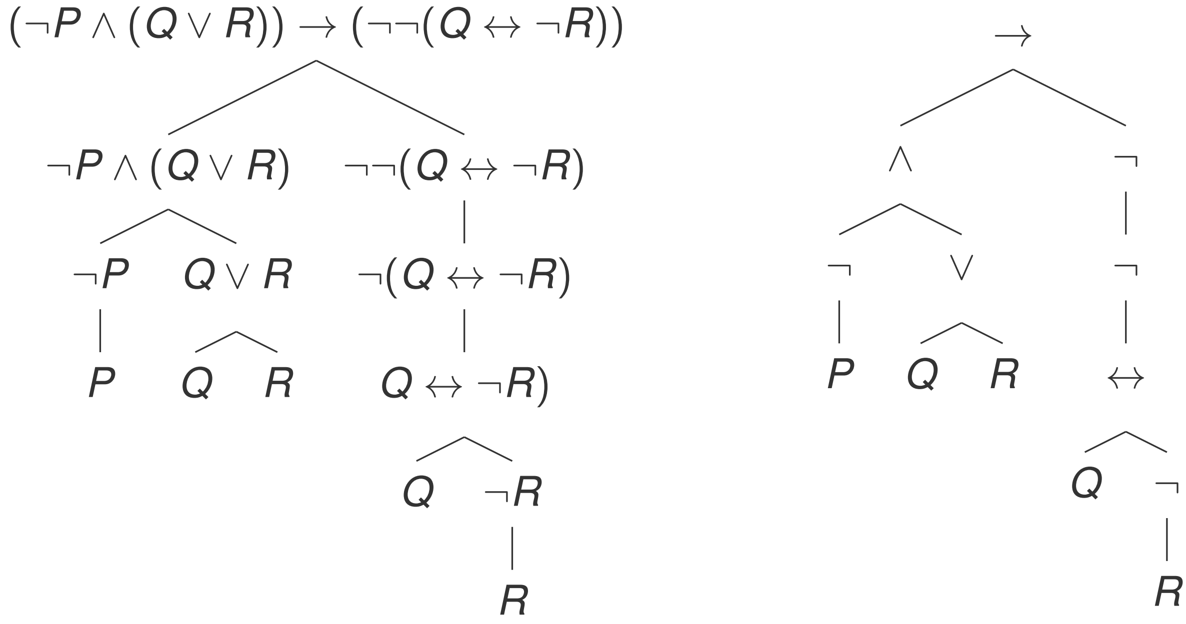
Using the given precedence order, redundant parentheses can be removed to simplify a formula:

Translation one $B \to (S \wedge J)$ can be reduced to $B \to S \wedge J$,
but translation two $(B \to S) \wedge J$ cannot be reduced.

The connectives of a formula have a fixed order of application, which are easier to see in the formula's syntactical decomposition tree.

SYNTACTICAL TREE

Syntactical decomposition tree of a formula



Each node of the tree on the left is a subformulae of the formula at its root. These subformulae correspond to subtrees of the tree on the right.

SEMANTICS

Semantics

The syntax of the language determine how statements are formed and evaluated, but it gives them no truth-values.

A wff needs to be *interpreted* to give it a truth-value and thus a semantical meaning (0 or 1).

An *interpretation* or valuation v is a function that assigns to every propositional symbol P a truth-value, i.e., $v(P) \in \{0, 1\}$.

- If $v(P) = 1$, then P is called *true* under the interpretation v.
- If $v(P) = 0$, then P is called *false* under the interpretation v.

An interpretation is extended to all well-formed formulae using *truth-tables*.

TRUTH-TABLES

Truth-table for Negation

The negation $\neg A$ of a formula A is true when A is false and false otherwise. Thus, if we know $v(A)$ then

$$v(\neg A) = \begin{cases} 0, & \text{if } v(A) = 1 \\ 1, & \text{if } v(A) = 0 \end{cases}$$

Corresponding truth-table:

A	$\neg A$
0	1
1	0

Think of the symbol "A" here as a placeholder for any given formula, and not necessarily as the propositional symbol "A".

Truth-table for Conjunction

The conjunction $A \wedge B$ is true if and only if both A and B are true. Thus, if we know $v(A)$ and $v(B)$ then:

$$v(A \wedge B) = \begin{cases} 1, & \text{if } v(A) = 1 \text{ and } v(B) = 1 \\ 0, & \text{if } v(A) = 0 \text{ or } v(B) = 0 \end{cases}$$

Corresponding truth-table:

A	B	$A \wedge B$
0	0	0
0	1	0
1	0	0
1	1	1

Truth-table for Disjunction

The disjunction $A \vee B$ is true if and only if at least one of A, B is true (i.e., either or both). Thus, if we know $v(A)$ and $v(B)$ then

$$v(A \vee B) = \begin{cases} 1, & \text{if } v(A) = 1 \text{ or } v(B) = 1 \\ 0, & \text{if } v(A) = 0 \text{ and } v(B) = 0 \end{cases}$$

Corresponding truth-table:

A	B	$A \vee B$
0	0	0
0	1	1
1	0	1
1	1	1

Truth-table for Implication

The implication $A \rightarrow B$ is false if and only if A is true and B is false. Thus, if we know $v(A)$ and $v(B)$ then

$$v(A \rightarrow B) = \begin{cases} 1, & \text{if } v(A) = 0 \text{ or } v(B) = 1 \\ 0, & \text{if } v(A) = 1 \text{ and } v(B) = 0 \end{cases}$$

Corresponding truth-table:

A	B	$A \rightarrow B$
0	0	1
0	1	1
1	0	0
1	1	1

More on the Implication Connective

An implication represents a conditional statement. In the statement $A \rightarrow B$, A is called the *hypothesis* (or *antecedent* or *premise*) and B is called the *conclusion* (or *consequence*).

$A \rightarrow B$ is often read as "if A, then B". However it has many different readings (Rosen, 6th edition, page 6):

"if A, then B"	"A implies B "
"A is sufficient for B"	"A only if B"
"B if A"	" a sufficient condition for B is A"
"B when A"	"B whenever A"
"a necessary condition for A is B"	"B is necessary for A"
"B unless $\neg A$"	"B follows from A"

Truth-table for Equivalence

The equivalence $A \leftrightarrow B$ is true if and only if A and B have the same truth-values. Thus, if we know $v(A)$ and $v(B)$ then

$$v(A \leftrightarrow B) = \begin{cases} 1, & \text{if } v(A) = v(B) \\ 0, & \text{if } v(A) \neq v(B) \end{cases}$$

Corresponding truth-table:

A	B	$A \leftrightarrow B$
0	0	1
0	1	0
1	0	0
1	1	1

Truth under a Valuation

The truth-value of a formula only depends on the truth-value of its propositional symbols.

Each distinct interpretation gives a different truth-value to the symbols in the language, and this may result in different truth-values for the formula under different interpretations.

Let A be a formula and v an interpretation.

- If $v(A) = 1$, then A is said to be *true* under the interpretation v.
- If $v(A) = 0$, then A is said to be *false* under the interpretation v.

Class exercise

1. Give all interpretations under which the formula $(P \vee \neg Q) \wedge R$ is true.

interpretation	P	Q	R	$\neg Q$	$P \vee \neg Q$	$(P \vee \neg Q) \wedge R$
v_0						
v_1						
v_2						
v_3						
v_4						
v_5						
v_6						
v_7						

2. How many distinct interpretations does a language with k propositional symbols have?

Logic puzzles

Using Logic to Solve Puzzles

The island of TuFa has two tribes, the Tu's who always tell the truth, and the Fa's who always lie.

A traveller encountered three residents of TuFa, P, Q, and R, and each made a statement to the traveller:

P said: "Q and R tell the truth iff R tells the truth."

Q said: "If P and R tell the truth, then it is not the case that if Q and R tell the truth, then P tells the truth."

R said: "Q is lying iff P or Q is telling the truth."

Determine which tribes each of P, Q, and R belongs to.

Let P be the statement "P is telling the truth" (and thus P is a Tu), etc.

Then in symbolic form:

P says: $(Q \wedge R) \leftrightarrow R$

Q says: $(P \wedge R) \rightarrow \neg((Q \wedge R) \rightarrow P)$

R says: $\neg Q \leftrightarrow (P \vee Q)$.

The following formulae are given to be true:

A: $P \leftrightarrow ((Q \wedge R) \leftrightarrow R)$,

B: $Q \leftrightarrow ((P \wedge R) \rightarrow \neg((Q \wedge R) \rightarrow P))$,

C: $R \leftrightarrow (\neg Q \leftrightarrow (P \vee Q))$.

v	P	Q	R	$A = P \leftrightarrow ((Q \wedge R) \leftrightarrow R)$	B	C	$A \wedge B \wedge C$
v_0	0	0	0	0	0	1	0
v_1	0	0	1	1	0	0	0
v_2	0	1	0	0	1	1	0
v_3	0	1	1	0	1	0	0
v_4	1	0	0	1	0	0	0
v_5	1	0	1	0	1	1	0
v_6	1	1	0	1	1	1	1
v_7	1	1	1	1	0	0	0

v_6 is the only interpretation that makes the three statements, A, B and C, true. Therefore, P and Q must belong to Tu, and R to Fa.

TAUTOLOGIES AND CONTRADICTIONS

Tautologies

Some formulae are special.

A *tautology* is a formula which is *true* under all interpretations.

Example: All formulae of the form $A \vee \neg A$ are tautologies, because $v(A \vee \neg A) = 1$, for all interpretations v:

A	$\neg A$	$A \vee \neg A$
0	1	1
1	0	1

We can define a 'new' symbol $\mathbf{1}$ by $\mathbf{1} =_{def} P \vee \neg P$. This symbol is also often written as $\top$.

Contradictions

A *contradiction* is a formula which is false under all interpretations.

Example: All formulae of the form $A \wedge \neg A$ are contradictions, because, $v(A \wedge \neg A) = 0$ for all interpretations v:

A	$\neg A$	$A \wedge \neg A$
0	1	0
1	0	0

Note that a formula A is a tautology if and only if $\neg A$ is a contradiction.

We can define a 'new' symbol $\mathbf{0}$ by $\mathbf{0} =_{def} P \wedge \neg P$. This symbol is also often written as $\perp$.

EQUIVALENCES

Logical Equivalences

Two formulae A and B are said to be *equivalent* if they have the same truth-value under *every* possible interpretation.

Therefore, A and B are equivalent if $v(A) = v(B)$ for every interpretation v. This is denoted by $A \equiv B$.

Observation: The following conditions are equivalent, for any two formulae A and B:

- $A \equiv B$;
- $A \leftrightarrow B$ is a tautology.

$A \to B$ and $\neg A \vee B$ are logically equivalent

A	B	$A \to B$	$\neg A$	$\neg A \vee B$
0	0	1	1	1
0	1	1	1	1
1	0	0	0	0
1	1	1	0	1

Thus, implication can be defined via negation and disjunction:

$$(A \to B) \equiv (\neg A \vee B).$$

Rephrasing Equivalence

A	B	$A \leftrightarrow B$	$A \rightarrow B$	$B \rightarrow A$	$(A \rightarrow B) \wedge (B \rightarrow A)$
0	0	1	1	1	1
0	1	0	1	0	0
1	0	0	0	1	0
1	1	1	1	1	1

Thus, the logical connective equivalence can be defined via implication and conjunction:

$$(A \leftrightarrow B) \equiv (A \rightarrow B) \wedge (B \rightarrow A)$$

To know more. . .

- The material in this session can be found in detail in Chapter 1 of 'Elementary Logic with Applications'.