Propositional logic

Dr. Constantinos Constantinides, P.Eng.

Department of Computer Science and Software Engineering
Concordia University Montreal, Canada

7 January, 2020

Introduction

- Logic was established as a discipline by Aristotle who set down rules dealing with statements called propositions.
- A *proposition* is a statement (or *declarative sentence*) that is either true or false (but not both).
- Propositions are called *atomic* if they cannot be broken down into other sentences, otherwise they are called *compound*.

Introduction /cont.

Examples of atomic propositions are

1. Ottawa is the capital of Canada.
2. $5 < 3$.

Introduction /cont.

Examples of compound propositions are

1. London is the capital of the UK and Washington DC is the capital of the USA.
2. $(3 > 7)$ *or* $(11 < 15)$.

Introduction /cont.

Examples of sentences which are not propositions are

1. What time is it?
2. Come here!
3. What a nice day!
4. $x > y$.

Introduction /cont.

Examples of sentences which are not propositions are

1. What time is it? Not declarative: *interrogative sentence* (or *question*).
2. Come here! Not declarative: *imperative sentence* (or *command*).
3. What a nice day! Not declarative: *exclamative sentence* (or *exclamation*).
4. $x > y$ Why is this not a proposition?

Introduction /cont.

- $x > y$ is not a proposition because the values of the variables x and y have not been assigned any value.
- As is, the sentence is a *predicate* and it can be assigned a truth value (and thus become a proposition) once the values of its variables are defined.

Producing new propositions: Negation

- Much like operators in arithmetic allow us to construct expressions (e.g. $(-3) \times (2 + 5)$) new propositions can be constructed by combining one or more existing propositions using *logical operators*.
- Let p and q be propositions:
- **Definition**: The statement $\neg p$ is a new proposition called the *negation* of p that is true when p is false, and is false otherwise.

Producing new propositions: Conjunction and disjunction

- Logical operators that are used to form new propositions from two or more existing propositions are called *connectives*.
- **Definition**: The statement $p \wedge q$ ("p and q") is a new proposition called the *conjunction* of p and q that is true when both p and q are true and is false otherwise.
- **Definition**: The statement $p \vee q$ ("p or q") is a new proposition called the *disjunction* of p and q that is false when both p and q are false and is true otherwise.
- Disjunction is called an *inclusive operation* because it includes the possibility that both p and q are true.

Producing new propositions: Conjunction and disjunction

- **Definition**: The statement $p \oplus q$ is a new proposition called the *exclusive or* of p and q that is true when exactly one of p and q is true and is false otherwise.
- Note that contrary to disjunction, this operation excludes the possibility that both p and q are true.
- In a natural language ambiguity may arise as the word "or" can be used as inclusive or exclusive. Normally, its intention can be deduced from context.

Producing new propositions: Implication

- **Definition**: For propositions p and q, the implication $p \to q$ is a new proposition that is false when p is true and q is false and is true otherwise.
- In this implication, p is called the *premise* (also: *antecedent*, *assumption* or *hypothesis*) and q is called the *conclusion* (or *consequent*).

Producing new propositions: Implication /cont.

- Note that $p \rightarrow q$ represents that q is true whenever p is true.
- Implication does not necessarily entail causation, i.e. the two propositions do not necessarily constitute cause and effect.
- Ways to express implication include the following:
  - "if p then q."
  - "q if p."
  - "q when p."
  - "p implies q."

Producing new propositions: Biconditional

- **Definition**: For propositions p and q, the *biconditional* $p \leftrightarrow q$ is a new proposition that is true when p and q have the same truth value and false otherwise.
- The biconditional of p and q can also be expressed as *"p is true if and only if (iff) q is true."*
- Note that the biconditional is the exact opposite of the exclusive-or, i.e. $p \leftrightarrow q$ means $\neg(p \oplus q)$.
- Note that the biconditional does not imply that both p and q are true, or that either of them causes the other, or that they have a common cause.

Truth tables

We can display the relationship between the truth values of these fundamental propositions using a *truth table* which is a visual depiction of the relationships between the truth values of propositions when logical operators are applied.

Truth table of fundamental compound propositions

p	q	$p \wedge q$	$p \vee q$	$p \oplus q$	$p \rightarrow q$	$p \leftrightarrow q$
T	T	T	T	F	T	T
T	F	F	T	T	F	F
F	T	F	T	T	T	F
F	F	F	F	F	T	T

Arity

- The term *arity* is used to describe the number of arguments or operands that an operator takes.
- A *unary* operator (arity 1) takes one argument. A *binary* operator (arity 2) takes two arguments.
- The negation operator is a unary operator. The rest of the operators discussed are binary operators.

Alternative notations

Note that in the literature, alternative notations exist for logical operators:

- Negation: $\sim$!
- Conjunction: &
- Disjunction: $+$ $||$
- Implication: $\Rightarrow$ $\supset$
- Biconditional: $\Leftrightarrow$ $\equiv$

Formulas

- Propositional logic is also called *symbolic logic* since it uses symbols called *propositional variables* (also: *propositional letters*, *sentenial variables* or *sentenial letters*) to represent knowledge in the form of propositions.
- The set of propositional variables, together with the symbols for logical connectives and parentheses, constitute an alphabet that allows us to build a *propositional formula* (also: *well-formed formula (wff)*, *sentential formula*, *propositional expression*, *propositional form*, or *sentence*).

Formulas /cont.

We are now ready to inductively define a formula by using a grammar as follows:

- Each propositional variable on its own is a formula (called *atomic formula*).
- If ϕ is a formula, then $\neg\phi$ is a formula.
- If ϕ and ψ are formulas, and $\bullet$ is any binary connective, then $(\phi \bullet \psi)$ is a formula, where $\bullet$ could be any of (but is not limited to) the usual operators $\wedge, \vee, \rightarrow, \leftrightarrow$.

Formulas /cont.

- The set of structural rules is referred to as a *grammar* and can help us to determine the correctness of formulas.
- For example, $(p \wedge q) \vee (r \rightarrow s)$ is a formula as it is gramatically correct, whereas $p \rightarrow qq$ is not a formula as it is not grammatically correct.

Compound propositions and precedence of logical operators

- As the term suggests, a *compound proposition* is one that combines individual propositions.
- Unless parentheses are used, negation takes the highest precedence in a compound proposition, followed by conjunction and disjunction.
- Conjunction takes higher precedence over disjunction.
- Implication and biconditional take the lowest precedence.
- Implication takes a higher precedence over biconditional.

Compound propositions and precedence of logical operators /cont.

Operator	Precedence
$\neg$	1: Highest.
$\wedge$	2
$\vee$	3
$\rightarrow$	4
$\leftrightarrow$	5: Lowest.

Example 1: Compound propositions

- $\neg p \vee q$ is the disjunction of $\neg p$ and q.
- $p \vee \neg q \wedge r$ means $p \vee (\neg q \wedge r)$ where the parenthesis is the conjunction of $\neg q$ and r.
- $\neg p \wedge q \wedge r \rightarrow s$ means $[(\neg p \wedge q) \wedge r] \rightarrow s$.

Visualizing compound propositions: Truth tables

- To construct a truth table for a compound proposition we have to produce rows that contain all possible combinations (called *interpretations*) of the truth values of all individual propositions together with the truth value of the compound proposition.
- In doing this, and knowing that each individual proposition can only have two values, we will have 2^n row entries for n individual propositions.
- The columns depicting the values of all individual propositions are called *operand columns* whereas the one depicting the values of the compound proposition is called the *result column*.

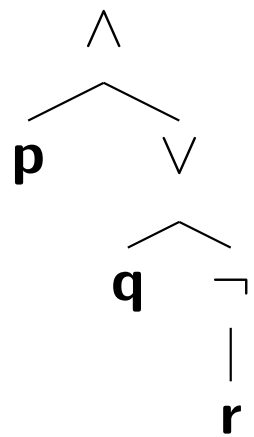
Example 2: Truth table

Consider the proposition $p \wedge (q \vee \neg r)$.

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

Visualizing compound propositions: Expression trees

- Another way to visualize a compound proposition is with an *expression tree* as the one shown below for $p \wedge (q \vee \neg r)$. Note that leaf nodes contain individual propositions while non-leaf nodes contain logical operators.
- For clarity, we have drawn leaf nodes in bold even though this is not necessary.



Tautologies, contradictions and contingencies

- **Definition**: A *tautology* is a (compound) proposition that is always true regardless of the values of its variables.
- **Definition**: A *contradiction* is a proposition that is always false regardless of the values of its variables.
- For example, for a proposition p, the statement $p \vee \neg p$ is a tautology, whereas $p \wedge \neg p$ is a contradiction.
- When the statement can assume any of the two values then it is called a *contingency*.

System specifications

- Sentences expressed in a natural language normally arise in system specifications.
- The advantage of such specification is readability.
- The disadvantage is that it can be ambiguous and verbose.
- We can translate such sentences into logical expressions in order to remove any ambiguity as well as to determine if the set of sentences is consistent or to infer conclusions (see later on *inference*).

Logical equivalence

- **Definition**: A set of sentences is *consistent* (or *satisfiable*) if and only if it is possible that every sentence in the set is true.
- **Definition**: Two propositions p and q are *logically equivalent*, denoted by $p \equiv q$, if they have the same truth values in all possible cases (i.e. for every combination of the truth values of their variables).

The inverse, converse and contrapositive of the implication statement

- **Definition**: The *inverse* of $p \to q$ is $\neg p \to \neg q$.
- **Definition**: The *converse* of $p \to q$ is $q \to p$.
- **Definition**: The *contrapositive* of $p \to q$ is $\neg q \to \neg p$.

Logical equivalence /cont.

On a truth table, we can determine if two propositions are logically equivalent if and only if the columns giving their truth values are identical.

p	q	$\neg p$	$\neg q$	$p \to q$	$\neg p \to \neg q$	$q \to p$	$\neg q \to \neg p$
T	T	F	F	T	T	T	T
T	F	F	T	F	T	T	F
F	T	T	F	T	F	F	T
F	F	T	T	T	T	T	T

Logical equivalence /cont.

From the table we can make the following observations:

1. The inverse and the converse of the implication proposition are logically equivalent, i.e.$(\neg p \rightarrow \neg q) \equiv (q \rightarrow p)$.
2. The implication proposition is logically equivalent to its contrapositive, i.e. $(p \rightarrow q) \equiv (\neg q \rightarrow \neg p)$.

Basic equivalence rules

Double negation	$\neg \neg p \equiv p$
Disjunction	$p \vee \mathit{true} \equiv \mathit{true}$ $p \vee \mathit{false} \equiv p$ $p \vee p \equiv p$ $p \vee \neg p \equiv \mathit{true}$
Conjunction	$p \wedge \mathit{true} \equiv p$ $p \wedge \mathit{false} \equiv \mathit{false}$ $p \wedge p \equiv p$ $p \wedge \neg p \equiv \mathit{false}$

Basic equivalence rules /cont.

Implication	$p \rightarrow \mathit{true} \equiv \mathit{true}$ $p \rightarrow \mathit{false} \equiv \neg p$ $\mathit{true} \rightarrow p \equiv p$ $\mathit{false} \rightarrow p \equiv \mathit{true}$ $p \rightarrow p \equiv \mathit{true}$
Absorption laws	$p \wedge (p \vee q) \equiv p$ $p \vee (p \wedge q) \equiv p$ $p \wedge (\neg p \vee q) \equiv p \wedge q$ $p \vee (\neg p \wedge q) \equiv p \vee q$
De Morgan's laws	$\neg(p \wedge q) \equiv \neg p \vee \neg q$ $\neg(p \vee q) \equiv \neg p \wedge \neg q$

Basic equivalence rules /cont.

$p \rightarrow q \equiv \neg p \vee q$	
$\neg(p \rightarrow q) \equiv p \wedge \neg q$	
$p \rightarrow q \equiv p \wedge \neg q \rightarrow \mathit{false}$	
$\wedge$ and $\vee$ are associative.	$(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$, and $(p \vee q) \vee r \equiv p \vee (q \vee r)$
$\wedge$ and $\vee$ are commutative.	$p \wedge q \equiv q \wedge p$ and $p \vee q \equiv q \vee p$
$\wedge$ and $\vee$ distribute over each other.	$p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$, $p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$

Necessary and sufficient conditions and criteria

- For propositions p and q, the implication $p \to q$ reads "p is a sufficient condition for q." This means that the occurrence of p guarantees the occurrence of q.
- How about necessity? When we say "p is a necessary condition for q" we mean that if p does not occur, then q cannot occur either, or

$$\neg p \to \neg q$$

- The occurrence of p is necessary to achieve the occurrence of q. As the statement is logically equivalent to its contrapositive, we can express this as

$$q \to p$$

Necessary and sufficient conditions and criteria /cont.

In summary, for propositions p and q:

- **Definition**: p is called a *sufficient condition* for q iff $p \rightarrow q$.
- **Definition**: p is called a *necessary condition* for q iff $q \rightarrow p$.
- **Definition**: p is called a *criterion* for q iff $p \leftrightarrow q$.

Necessary and sufficient conditions and criteria /cont.

- *Being a Canadian citizen and of voting age is a necessary condition for being Prime Minister of Canada*.
- This can be rewritten as follows:
- **If** one is Prime Minister of Canada, **then** one is a Canadian citizen *and* of legal voting age.

Necessary and sufficient conditions and criteria /cont.

- "*Divisibility by 2 is a criterion for being an even number.*"
- This can be rewritten as a biconditional as follows: (Divisibility by 2) $\leftrightarrow$ (even number).

Arguments

- In common English usage, an *argument* refers to a dispute between two parties. In logic this is not the case. An argument is formally defined as follows:
- **Definition**: An *argument* is a set of sentences, one of which is the conclusion, supported by evidence or reasons provided by the remaining sentences which serve as premises.

Validity of arguments

- In propositional logic we are interested in the notion of *validity*. We can define validity as follows:
- **Definition**: An argument is *(deductively) valid* if and only if it is impossible that all its premises are true while its conclusion is false.
- This implies that an example of a situation in which the premises are true while the conclusion is false will guarantee invalidity of the argument.

Syllogisms

- An argument form that consists of two premises and a conclusion is called a *syllogism*, where the first and second premises are called the major and minor premises respectively.
- One such syllogism is *modus ponens*:

$$((p \to q),\ p) \vdash q$$

  where the $\vdash$ symbol reads "yields."
- For example:
  - **Major premise**: **If** December 25 is a public holiday (p), **then** school is closed. (q)
  - **Minor premise**: December 25 is a public holiday. (p)
  - **Conclusion**: **Therefore**, school is closed. (q)

Rules of inference and common validating rules

- The concept of logical necessity is important to determine validity.
- The process of deriving a conclusion from a set of premises is called *inference*.
- A *rule of inference* is a valid argument form. A number of rules exist that can provide a justification for inference.
- We have already seen *modus ponens* which is expressed in *sequent notation* as $((p \rightarrow q), p) \vdash q$.

Rules of inference and common validating rules /cont.

- The rule can also be expressed in a propositional sentence as $((p \to q) \wedge p) \to q$, or in *rule form* as

$$\frac{p \to q,\ p}{\therefore q}$$

  where the $\therefore$ symbol reads "therefore."
- *Modus ponens* is included in a collection of rules called *common validating rules* (or *inference rules*).

Rules of inference and common validating rules /cont.

Name	Rule
Modus ponens	$p \to q, p \vdash q$
Modus tollens	$p \to q, \neg q \vdash \neg p$
Disjunctive syllogism (Elimination)	$p \vee q, \neg p \vdash q$
Simplification	$p \wedge q \vdash p$
Hypothetical syllogism (Transitivity)	$p \to q, q \to r \vdash p \to r$

Example 3: Syllogisms

- Consider the following example:
  - **Major premise**: **If** *Socrates is human* (p), **then** *Socrates is mortal.* (q)
  - **Minor premise**: *Socrates is human.* (p)
  - **Conclusion**: **Therefore**, *Socrates is mortal.* (q)
- This is an example of *modus ponens* ($p \rightarrow q, p \vdash q$).
- Note the logical necessity of the conclusion given the two premises.
- It is impossible for the premises to be true, while the conclusion is false.

Example 4: Syllogisms

- Is the following argument valid?
  - **Major premise**: *Canada shares a land border with the USA* (p), **or** *Mexico shares a land border with the USA*. (q)
  - **Minor premise**: *Canada does not share a land border with the USA*. ($\neg p$)
  - **Conclusion**: **Therefore**, *Mexico shares a land border with the USA*. (q)
- This is indeed a valid argument, as it is an example of disjunctive syllogism ($p \vee q, \neg p \vdash q$).
- Given the two premises (i.e. *if* they were all true), the conclusion is logically necessary (and thus true by default).
- It just so happens that in this example not all premises are true despite the fact that the conclusion is true.

Truthfulness and validity

- We distinguish between truthfulness and validity:
- *Truthfulness* is a property of individual propositions, and
- *Validity* is a property of argument forms, indicating whether or not they have such a structure that it is absolutely impossible for the premises to be true without the conclusion also being true.

Example 5: Validity

- Consider the following statement: *"To serve in the Canadian Parliament, one must be a Canadian citizen and at least 18 years of age."*
- We can re-write this as "Being a Canadian citizen and at least 18 years of age (p) is necessary for serving in parliament (q)":
- Formally we can write "p is necessary for q", or

$$\neg p \rightarrow \neg q$$

meaning that if p does not occur, then q cannot occur either.

Example 5: Validity /cont.

- Taking the contrapositive (why?) we write

$$q \rightarrow p$$

or *"If one is serving in the Canadian Parliament, then one is a Canadian citizen and at least 18 years of age.*

Example 6: Validity

Consider the following statement: *"Unless an exception is thrown, this loop will repeat n times."*

- We can rephrase it as a conditional: "**If** *an exception is not thrown* ($\neg p$), **then** *this loop will repeat n times* (q)."
- Formally we can write $\neg p \rightarrow q$.
- Taking the contrapositive (why?), we have $\neg q \rightarrow p$, or "**If** *this loop does not repeat n times* ($\neg q$), **then** *an exception has been thrown* (p)."

Conditional statement in programming languages

- Most programming languages support implication through a number of different constructs.
  1. As an if statement (with an optional else statement):
     if *condition* [**then**] `expression` [[**else**] `expression2`]
     where anything in square brackets [and] denotes an optional term.
  2. As a nested if statement.

Conditional statement in programming languages /cont.

3. As a statement modifier:
   condition **?** `expression` **;** `expression2`
   Note that the question mark (?) serves a *ternary operator* in this case. If *condition* is false, then `expression2` is evaluated.
4. As an **unless** statement which serves as a negated if statement:
   unless *condition* [**then**] `expression2` [[**else**] `expression`]
   The unless-expression will only be evaluated if the condition is false.

Non-validating patterns

Some common non-validating patterns (also called *logical fallacies*) are shown below:

Affirming the consequent (or "converse error")	$p \to q, q \vdash p$
Denying the antecedent (or "inverse error")	$p \to q, \neg p \vdash \neg q$

Example 7: Syllogisms

- In the English language we can freely change the order of the antecedent and the consequent in an implication statement without affecting their (logical) roles.
- The clause with the *if* is always the antecedent. Consider the following argument:
  - **Major premise**: The prime minister will be angry (q) if the parliament votes against the proposal (p).
  - **Minor premise**: The parliament did not vote against the proposal ($\neg p$).
  - **Conclusion**: So the prime minister will not be angry ($\neg q$).
- The argument is not valid by inverse error (denying the antecedent).

Example 8: Syllogisms

- Consider the following argument:
  - **Major premise**: **If** *I live in Montreal* (p), **then** *I live in Canada.* (q)
  - **Minor premise**: *I live in Canada.* (q)
  - **Conclusion**: **Therefore**, *I live in Montreal.* (p)
- The argument is invalid by the converse error (despite the fact that the conclusion might be true).

Example 9: Wason selection task

- You are shown a set of four cards placed on a table, each of which has a number on one side and a letter on the other side.
- The visible faces of the cards show **E**, **K**, **4**, **7**.
- Which card(s) must you turn over in order to test the truth of the proposition that if a card shows an even number on one face, then its opposite face shows a vowel?

Example 9: Wason selection task /cont.

- To test the implication *even* $\rightarrow$ *vowel*, we proceed as follows:

1. Apply *modus ponens* to check all even cards in order to ensure they correspond to a vowel, i.e. check **4**, and
2. Apply *modus tollens* to check all non-vowels in order to ensure that they correspond to a non-even number, i.e. check **K**.

- Checking **E** (and subsequently making any claim based on it) would be wrong (converse error).
- Checking **7** (and subsequently making any claim based on it) would be wrong (inverse error).

Validity and soundness

- We introduce the notion of *soundness* as follows:
- **Definition**: An argument is *sound* if a) it is valid and b) all its premises are true.
- Note that it is possible to have a deductive argument that is logically valid but is not sound.
- All arguments are either valid or invalid. Furthermore, valid arguments are either sound or unsound. There is never a middle ground, such as somewhat valid or somewhat sound.

Validity and soundness /cont.

- Many times in everyday conversation, people tend to confuse validity and truth by accepting a conclusion as true in cases where an argument is valid.
- However, the notion of validity makes no claim on the truthfulness of otherwise of the premises or the conclusion of an argument, but rather only to the form of the argument.
- Similarly, people tend to reject the truthfulness of a conclusion if the argument is not valid.

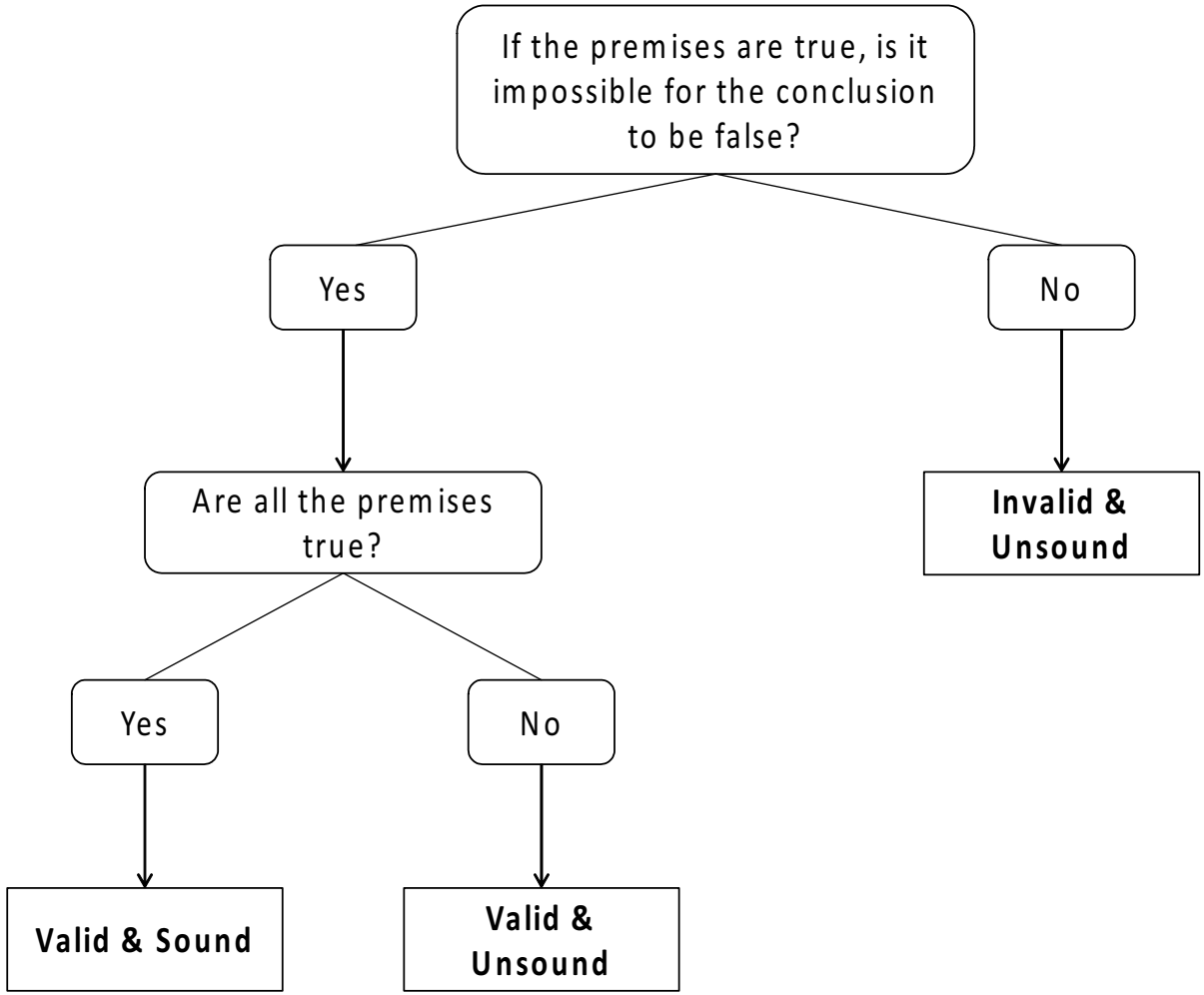
Validity and soundness /cont.

- It is important to stress that validity and truthfulness are two different things.
- An argument can be valid, yet with a false conclusion.
- Similarly, an argument can be invalid, yet with a true conclusion.
- **An argument can only be accepted if it is sound. Otherwise it must be rejected**.

Validity and soundness /cont.

If the **form** is...	...and the **premises** are	..then the **conclusion** is...	..and thus the **argument** is
Valid.	All true.	True.	Sound.
Valid.	Not all true.	Can be true or false.	Not sound.
Invalid.	Can be true or false.	Can be true or false.	Not sound.

Validity and soundness /cont.



Example 10: Argumentation

- Consider the following set of sentences:
  - Deven is trustworthy.
  - No one trustworthy would use my laptop without asking.
  - Deven used my laptop without asking.
- Is this an argument?

Example 10: Argumentation /cont.

- The set of sentences is not an argument because there is no statement that serves as a conclusion while others serve as premises.
- What is the problem with this set of statements?
- The three sentences are logically connected in a way that makes it impossible for all of them to be true. Such a set is called inconsistent.

Example 11: Validity and soundness

- Indicate correct or incorrect *terminology*:
  - *Philip presented a valid argument.* ✓
  - *Sammy's statements are unsound.* ×
  - *Sebastien's beliefs are sound.* ×
  - *Sophia's argument is consistent.* ×

Example 11: Validity and soundness /cont.

- Indicate correct or incorrect *terminology*:
  - *Genevieve's conclusion is false.* ✓
  - *Marwa's conclusion is invalid.* ×
  - *Evgeniy presented a true argument.* ×
  - *Stewart's statement is valid.* ×
  - *Jasmine gave a valid argument.* ✓

Summary of terminology

- A declarative statement (proposition) can be **true** or **false**.
- A set of statements (or *beliefs*) may be **consistent** or **inconsistent**.
- An argument may be
  valid or **invalid**,
  sound, or **unsound**.

Summary of terminology /cont.

- In other words, and contrary to common usage of terms, in formal logic we will never say things like
  - *A person has made a valid point.*
  - *The argument has a valid conclusion.*
- To prove that an argument is invalid, it is enough to provide a counter-example describing a possible situation in which the premises are all true, and the conclusion is false.