Lecture 1 - 5th Sept 2024

Formal Verification (FV): viewing a program as a logical argument and determining whether it behaves correctly using symbolic reasoning.

What is logical reasoning?

- **Premises:** facts
- **Conclusions:** what we deduce from the premises. The conclusion logically follows from the facts.
  - ↳ **Example:** if the train arrives late and there are no taxis at the station, then John is late for his meeting. John is not late for his meeting. The train did arrive late. Were there taxis at the station?
    - ↳ Yes!

What is formal logic?

- **Syntax:** acceptable sentence in the logic
- **Semantics:** what the symbols and sentences in the logic mean.
- **Proof theory:** how the valid proofs in the logic are constructed.
- **Logic provides:**
  - ↳ A language for expressing knowledge precisely (modelling / specification)
  - ↳ A way to reason about the consequences of that knowledge rigorously (proof / verification)
- A logic is formal if there is only one possible interpretation of a formula in the language.

- formal logic is concerned with the structure of the argument.
- we use symbols in the syntax to represent phrases in the sentences. It's also called symbolic logic.
  ↳ using symbols allows us to conquer complexity and eliminate non-logical aspects of argument.

- Propositional logic: a declarative sentence that is either true or false, but not both.

- Predicate logic: a means of describing relationships between objects, and a quantification over objects.
  ↳ eg: "every course has an instructor."

- specification: ways of describing what a system is required to do.

- program correctness: a progam is correct iff the output is correct for every input.

- the syntax of a logic defines how to arrange symbols into a well-formed formula (wff)

Semantics:

- $\models$: logical implication ← NOT a part of wff, but a meta-level symbol.
- $\Leftrightarrow$: logical equivalence
- $\vdash$: proves → $P_1, P_2, \ldots, P_n \vdash Q$ means that from $P_1, P_2, \ldots, P_n$, we can prove Q using a proof theory.

Proof Theories: methods that perform **mechanical manipulations** on strings of symbols.
↳ based on pattern matching!

On $P_1, P_2, \ldots, P_n \vdash Q$ and $P_1, P_2, \ldots, P_n \models Q$,

goals: if we **aren't yet sure they are valid** / proven
theorems: if they've been **determined to be valid** / proven

Soundness: a proof theory is sound **if whenever** $P_1, P_2, \ldots, P_n \vdash Q$ (we have a proof), **then** $P_1, P_2, \ldots, P_n \models Q$ (valid)

Completeness: a proof theory is complete **if whenever** $P_1, P_2, \ldots, P_n \models Q$ (valid) **then** $P_1, P_2, \ldots, P_n \vdash Q$ (there is a proof).

Propositional Connectives:

symbol	informal meaning	george
$\neg$	negation (not)	!
$\wedge$	conjunction (and / both)	&
$\vee$	disjunction (or / at least one)	|
$\Rightarrow$	implication (if / implies)	=>
$\Leftrightarrow$	equivalent (biconditional, iff)	<=>

Brackets and Precedence:

- Brackets around the outermost formula are usually omitted.

- The order of precedence for the operators is shown in the above table, with ¬ having the highest precedence and <=> having the lowest.

- All binary logical connectives are right-associative.
  - ↳ Eg: $a \vee b \vee c \rightarrow a \vee (b \vee c)$
    $a \Rightarrow b \Rightarrow c \rightarrow a \Rightarrow (b \Rightarrow c)$

Terminology:

- For $P \wedge Q$, P and Q are conjucts
- For $P \vee Q$, P and Q are disjuncts
- For $P \Rightarrow Q$, P is the premise/antecedent/hypothesis, and Q is the consequent/conclusion

- The contrapositive of $p \Rightarrow q$ is $\neg q \Rightarrow \neg p$.

Prime Proposition: a proposition that cannot be broken down further (indecomposable).
↳ Eg: "the snow is red"

Compound Proposition: a proposition that contains multiple prime propositions joined by connectives.
↳ Eg: "the snow is red and the grass is green".

Formalizing Natural Language:

$\neg P$: not P, P does not hold, it is not the case

that P, P is false

$P \wedge Q$: P and Q, P but Q, P despite Q, not only P but Q, P while Q, P yet Q, P although Q.

$P \vee Q$: P or Q, P or Q or both, P "and/or" Q, P unless Q

$P \Rightarrow Q$: if P then Q, Q if P, P only if Q, Q when P, P is sufficient for Q, Q is necessary for P, P implies Q.

$P \Leftrightarrow Q$: P if and only if Q, P is necessary and sufficient for Q, P exactly if Q, P is exactly Q.

Example: formalize the following. Let C = "it is cold" and S = "it is snowing".

1) It is cold but it is not snowing
   ↳ $C \wedge \neg S$

2) It is cold only if it is snowing
   ↳ $S \Rightarrow C$

3) It is snowing only if it is cold
   ↳ $C \Rightarrow S$

Example: "it is not the case that I pass the course if I fail the exam". Let F be "I fail the exam" and C be "I pass the course".

↳ $F \Rightarrow \neg C$; "failing the exam implies not passing", or $\neg (F \Rightarrow C)$?

both seem viable, so let's use a truth table!

↓

F	C	$F \Rightarrow \neg C$	$\neg (F \Rightarrow C)$	
0	0	1	0	
0	1	1	1	→ same!
1	0	1	0	
1	1	0	0	→ same!

↓ more correct.

∴, $F \Rightarrow \neg C$.

- Sometimes, logical connectives don't exactly match their meanings in English.
  - ↳ Eg: "the driver hit the cyclist and drove on" is very different to "the driver drove on and hit the cyclist", even though $P \wedge Q == Q \wedge P$. ↳ bc of the temporal (time) element!

Exclusive OR: $(P \vee Q) \wedge \neg (P \wedge Q)$

↳ Eg: "You may take Thursday off or you may take Friday off".

- We have to be careful of the meaning of implication.

↳ especially the "false implies anything" problem – if the antecedent is false, the implication is true.

- **Classical logic is two-valued:** the two possible truth values are T and F.
  - ↳ T denotes the property of a formula being true.
  - ↳ F denotes the property of a formula being false.

↳ the range of the semantic function for propositional logic is the set of truth values: $Tr = \{T, F\}$.

Boolean Valuation (BV): a function from the set of wffs in propositional logic to the set Tr.

↳ given a formula $p \wedge q$, we write $[p \wedge q]$ to mean "the meaning of the formula" in a certain boolean valuation. The [] is a function mapping syntax to its value.

In all boolean valuations:

- $[false] = F$, $[true] = T$
- $[\neg P] = NOT([P])$
- For the connectives,
  - $[P \wedge Q] = [P]$ AND $[Q]$
  - $[P \vee Q] = [P]$ OR $[Q]$
  - $[P \Rightarrow Q] = [P]$ IMP $[Q]$
  - $[P \Leftrightarrow Q] = [P]$ IFF $[Q]$

↳ when describing a boolean valuation, we only need to describe the association of truth values with the proposition symbols.

Example: show the truth value associated with the formula $(p \Rightarrow q) \wedge r$ in the boolean valuation where $p = [T]$, $q = [F]$, and $r = [F]$.

$$
\begin{aligned}
[(P \Rightarrow q) \wedge r] &= [p \Rightarrow q] \text{ AND } [r] \\
&= ([P] \text{ IMP } [q]) \text{ AND } [r] \\
&= (T \text{ IMP } F) \text{ AND } F \\
&= F \text{ AND } F \\
&= F.
\end{aligned}
$$

↳ "can it be true?"

Satisfiability: a formula P is satisfiable if there is a boolean valuation such that $[P] = T$.

↳ "is it always true?"

Tautology: a formula P is a tautology (or valid) if $[P] = T$ for all Boolean valuations.

↳ Eg: $P \vee \neg P$

P	¬P	P v ¬P
0	1	1
1	0	1

↳ since the expression is true for all boolean valuations, it is a tautology (and is also obviously satisfiable).

↳ Eg: $\neg(p \wedge \neg q)$

↳ when $[P] = T$ and $[Q] = T$:

NOT([P] AND NOT([Q]))

NOT(T AND NOT(T))

NOT(T AND F)

NOT(F)

T

since there is a boolean valuation for which the statement is true, it is satisfiable.

when $[P] = T$ and $[Q] = F$:

NOT([P] AND NOT([Q]))

NOT(T AND NOT([F]))

NOT(T AND T)

NOT(T)

F

since there is a boolean valuation for which the statement is false, it is not a tautology.

∴, the statement is satisfiable, but not a tautology!

- When a formula Q is a tautology, we write:

$\models Q$, which is pronounced "entails Q".

↳ $\models$ is a meta-symbol, it's not a part of the syntax of propositional logic - it tells us something about a proposational logic formula.

Logical Implication: a formula P logically implies a formula Q if and only if, for all boolean valuations, if $[P] = T$, then $[Q] = T$.

↳ $P \models Q$, pronounced "P entails Q" or "P logically implies Q"

this can be generalized: a set of formulas $P_1, P_2, \ldots, P_n$ logically imply a formula Q if and only if, for all boolean valuations, if $[P_1] = T$ and $[P_2] = T$ and $\ldots [P_n] = T$ then $[Q] = T$.

↳ $P_1, P_2, \ldots, P_n \models Q$ → also called a valid argument.

↳ equivalent to saying $P_1 \wedge P_2 \wedge \ldots P_n \models Q$, or also $\models P_1 \wedge P_2 \wedge \ldots \wedge P_n \Rightarrow Q$!

Contradiction: a propositional formula A is a contradiction (or a falsehood) if $[A] = F$ for all boolean valuations.

↳ Eg: $P \wedge \neg P$.

Contingent: A contingent formula is one that is neither a tautology nor a contradiction.

↳ ie - has a mixture of Ts and Fs.

Logical Equivalence: two formulas, P and Q, are logically

equivalent if and only if $[P] = [Q]$ in all boolean valuations.

↳ $P \Longleftrightarrow Q$, pronounced "P is logically equivalent to Q"

↳ $P \Longleftrightarrow Q$ iff $\models P \Leftrightarrow Q$.

Consistency: a collection of formulas is consistent if there is a boolean valuation in which all the formulas are T.

↳ Example: are the following sentences consistent?

1) Sales of houses fall off if interest rates rise
2) Auctioneers aren't happy if sales of houses fall off.
3) Interest rates are rising.
4) Auctioneers are happy.

First, we must formalize these sentences:

- let s be "sales of houses fall off"
- let r be "interest rates rise"
- let h be "auctioneers are happy".

∴ the formulas are:

1) $r \Rightarrow s$
2) $s \Rightarrow \neg h$
3) r
4) h

→

			1	2	3	4
s	r	h	r⇒s	s⇒¬h	r	h
0	0	0	1	1	0	0
0	0	1	1	1	0	1
0	1	0	0	1	1	0
0	1	1	0	1	1	1
1	0	0	1	1	0	0
1	0	1	1	0	0	1
1	1	0	1	1	1	0
1	1	1	1	0	1	1

↳ since there's no boolean valuation for which each sentence is true, the set of formulas is inconsistent!

Example: is this formula satisfiable?:

$$\neg(p \Rightarrow q) \wedge \neg(\neg r \Rightarrow \neg p)$$

since the "terms" are conjuncts, for the formula to be satisfiable, there needs to be some boolean valuation of p, q, and r, for which each term = T.

↳ for $\neg(p \Rightarrow q)$ to be T, $p \Rightarrow q$ must be F. this only occurs if $[P] = 1$ and $[Q] = 0$.

since $[P]$ must be 1, $\neg(\neg r \Rightarrow \neg p)$ can be written as $\neg(\neg r \Rightarrow NOT([T]))$.

↳ $\neg(\neg r \Rightarrow F)$ ∴, $[r] = F$.

we see that when $[p] = T$, $[q] = F$, and $[r] = F$, the formula evaluates to true. ∴, the formula is satisfiable!

Example: is $p \Rightarrow q, q \Rightarrow p \models q \wedge p$ a valid argument?

↳

p	q	p⇒q	q⇒p	q ∧ p
0	0	1	1	0
0	1	1	0	0
1	0	0	1	0
1	1	1	1	1

this row is a counterexample!

no, because $q \wedge p$ is not always 1 when $p \Rightarrow q$ and $q \Rightarrow p$ are both 1.

- We can use truth tables to check these properties, but since a truth table needs 2^n rows, they can be very tedious. Therefore, we can use proof theories to determine these properties.
  - ↳ As long as the proof theory is sound, we can use it to determine tautologies and valid arguments.

Proof Theory: Transformational Proof

Transformational Proof: an algebraic manipulation of formulas in propositional logic following rules.

Formal definition: a transitional proof is a means of determining if two well-formed formulas, P and Q, are logically equivalent by the (repeated) exchange of subformulas of P for logically equivalent subformulas that result in P being transformed into Q.

- Each step must follow a logical law
- The logical laws are expressed using the symbol $\leftrightarrow$.
- Equivalences that we can derive using transformational proof are expressed using the symbol $\leftrightarrow$.

Transformational Proof Rules:

- **Comm - assoc:**
  - ↳ $P \wedge Q \leftrightarrow Q \wedge P \quad \rightarrow P \wedge (Q \wedge R) \leftrightarrow (P \wedge Q) \wedge R$
  - ↳ $P \vee Q \leftrightarrow Q \vee P \quad \rightarrow P \vee (Q \vee R) \leftrightarrow (P \vee Q) \vee R$
  - ↳ $P \Leftrightarrow Q \leftrightarrow Q \Leftrightarrow P$

- **lem:** $P \vee \neg P \leftrightarrow true$
- **contr:** $P \wedge \neg P \leftrightarrow false$
- **impl:** $P \Rightarrow Q \leftrightarrow \neg P \vee Q$
- **idemp:** $P \vee P \leftrightarrow P$, $P \wedge P \leftrightarrow P$
- **neg:** $\neg(\neg P) \leftrightarrow P$
- **simp1:**
  - $P \wedge true \leftrightarrow P$, $P \vee true \leftrightarrow true$
  - $P \wedge false \leftrightarrow false$, $P \vee false \leftrightarrow P$.
- **dm:** $\neg(P \wedge Q) \leftrightarrow \neg P \vee \neg Q$, $\neg(P \vee Q) = \neg P \wedge \neg Q$
- **distr:**
  - $P \vee (Q \wedge R) \leftrightarrow (P \vee Q) \wedge (P \vee R)$
  - $P \wedge (Q \vee R) \leftrightarrow (P \wedge Q) \vee (P \wedge R)$
- **contrapos:** $P \Rightarrow Q \leftrightarrow \neg Q \Rightarrow \neg P$
- **equiv:** $P \Leftrightarrow Q \leftrightarrow (P \Rightarrow Q) \wedge (Q \Rightarrow P)$
- **simp2:**
  - $P \vee (P \wedge Q) \leftrightarrow P$
  - $P \wedge (P \vee Q) \leftrightarrow P$

Example: prove $\neg(\neg p \vee \neg(r \vee s)) \leftrightarrow p \wedge r \wedge s$

1) $\neg(\neg p \vee \neg(r \vee s))$
2) $\neg\neg p \wedge \neg\neg(r \vee s)$ by dm
3) $p \wedge \neg\neg(r \vee s)$ by neg
4) $p \wedge (r \vee s)$ by neg
5) $p \wedge r \vee s$ by distr.

↱ aka prove simp2!

Example: prove that $p \vee p \wedge q \leftrightarrow p$

1) $p \vee p \wedge q$
2) $(p \vee p) \wedge (p \vee q)$ by distr
3) $p \wedge (p \vee q)$ by idemp
4) $(p \vee \text{false}) \wedge (p \vee q)$ by simp1
5) $p \vee (\text{false} \wedge q)$ by distr
6) $p \vee \text{false}$ by simp1
7) p by simp1!

Example: prove $\neg((p \wedge q) \Rightarrow p) \leftrightarrow \text{false}$ ↑ aka, prove that it's a contradiction!

1) $\neg((p \wedge q) \Rightarrow p)$
2) $\neg(\neg(p \wedge q) \vee p)$ by impl
3) $\neg\neg(p \wedge q) \wedge \neg p$ by dm
4) $p \wedge q \wedge \neg p$ by neg
5) $q \wedge \text{false}$ by simp1
6) false by simp1

Example: prove $\neg\text{true} \leftrightarrow \text{false}$

1) $\neg\text{true}$
2) $\neg(p \vee \neg p)$ by lem
3) $\neg p \wedge \neg\neg p$ by dm
4) $\neg p \wedge p$ by neg
5) false by contr!

Example: $p \wedge (\neg(\neg q \wedge \neg p) \vee p) \leftrightarrow p$

1) $p \wedge (\neg(\neg q \wedge \neg p) \vee p)$

2) $p \wedge (\neg\neg q \vee \neg\neg p \vee p)$ by dm

3) $p \wedge (q \vee \neg\neg p \vee p)$ by neg

4) $p \wedge (q \vee p \vee p)$ by neg

5) $p \wedge (q \vee p)$ by idemp

6) p by simp2.