COL703: Logic for Computer Science (Jul-Nov 2023)

Lectures 1–5 (Propositional Logic, Natural Deduction, Resolution)

Kumar Madhukar

madhukar@cse.iitd.ac.in

July 24th and 31st, Aug 3rd, 7th and 10th, 2023

Motivation

- The aim of logic in computer science is to develop languages to model the situations we encounter.
- Why? So that we reason about the situations formally.
- Why do we want to do this formally? So that we make <u>valid</u> arguments while reasoning, so that the arguments can be defended rigorously, and can even be executed on a machine.

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. *Therefore*, there were taxis at the station.
- Intuitively, the argument seems valid. Why?
- The sentence after the 'therefore' logically <u>follows from</u> the sentences before it.

Another example

- If it is raining and Mark does not have his umbrella with him, then he will get wet. Mark is not wet. It is raining. *Therefore*, Mark has his umbrella with him.

- Seems a valid argument as well. In fact, has the same structure as the previous example.

Another example

- If it is raining and Mark does not have his umbrella with him, then he will get wet. Mark is not wet. It is raining. *Therefore*, Mark has his umbrella with him.

- Seems a valid argument as well. In fact, has the same structure as the previous example.

- If p and not q, then r. Not r. p. *Therefore*, q.

Declarative sentences

- In order to make arguments rigorous, we need to develop a language.
- To express sentences in a way that brings out their logical structure.
- Propositional logic – based on propositions (or declarative sentences) which one can argue as being true or false.

Declarative sentences

- The sum of numbers 3 and 5 equals 8.
- All students registered for COL703 are present in the class today.
- We won't consider non-declarative sentences: e.g. Could you please pass me the salt?
- We are interested in precise declarative sentences, or statements about the behaviour of computer systems, or programs.
- And a calculus of reasoning (so that we can argue whether certain inferences can be made correctly or not).

Propositional logic

- Consider atomic, or indecomposable, declarative sentences, e.g., "The number 5 is even."
- We assign distinct symbols to these atomic sentences: p, q, r, ...
- Code up complex sentences in a compositional way, using symbols $\neg$, $\wedge$, $\vee$, $\rightarrow$.

Example

- p: The number 5 is even.
- q: The number 5 is prime.
- r: The number 5 is bigger than the number 2.

- $\neg p$: The number 5 is **not** even. (Or, equivalently, it is **not** true that the number 5 is even.)

- $p \vee q$: (at least one of these is true) The number 5 is even **or** the number 5 is prime.

- $p \wedge q$: The number 5 is even **and** the number 5 is prime.

- $q \rightarrow p$: **If** the number 5 is prime, **then** the number 5 is even.

Binding priorities

- $\neg$ binds more tightly than $\wedge$ and $\vee$
- $\wedge$ and $\vee$ bind more tightly than $\rightarrow$
- $\rightarrow$ is right-associative: $p \rightarrow q \rightarrow r$ denotes $p \rightarrow (q \rightarrow r)$
- $p \wedge q \rightarrow \neg r \vee q \quad$ is $\quad (p \wedge q) \rightarrow ((\neg r) \vee q)$

Natural deduction

- Calculus for reasoning about propositions.
- Proof rules that can allow us to infer formulas from other formulas.
- $\phi_1, \phi_2, \ldots, \phi_n \vdash \psi$

  (e.g., $p \wedge \neg q \rightarrow r$, $\neg r$, $p \vdash q$)
- This sequent is valid if a proof for it can be found using the proof rules.
- The rules should allow valid arguments and disallow invalid ones.

Rules for natural deduction

1. Rules for conjunction

Examples

Prove that $p \land q, r \vdash q \land r$ is valid.

Examples

Prove that $p \wedge q, r \vdash q \wedge r$ is valid.

Prove that $(p \wedge q) \wedge r, s \wedge t \vdash q \wedge s$ is valid.

Rules for natural deduction

1. Rules for conjunction
2. Rules for double negation

Examples

Prove that $p, \neg\neg(q \wedge r) \vdash \neg\neg p \wedge r$ is valid.

Rules for natural deduction

1. Rules for conjunction
2. Rules for double negation
3. Rules for implication

Examples

Prove that $p, p \to q, p \to (q \to r) \vdash r$ is valid. (elimination)

Prove that $p \to (q \to r), p, \neg r \vdash \neg q$ is valid. (elimination)

Prove that $p \to q \vdash \neg q \to \neg p$ is valid. (introduction)

Prove that $\neg q \to \neg p \vdash p \to \neg\neg q$ is valid. (introduction)

Examples (Theorems)

Prove that $\vdash p \to p$ is valid.

Examples (Theorems)

Prove that $\vdash p \to p$ is valid.

Prove that $\vdash (q \to r) \to ((\neg q \to \neg p) \to (p \to r))$ is valid.

Some more examples

$p \wedge q \to r \vdash p \to (q \to r)$

Some more examples

$p \wedge q \to r \vdash p \to (q \to r)$

$p \to (q \to r) \vdash p \wedge q \to r$

Some more examples

$p \wedge q \to r \vdash p \to (q \to r)$

$p \to (q \to r) \vdash p \wedge q \to r$

$p \to q \vdash p \wedge r \to q \wedge r$

Rules for natural deduction

1. Rules for conjunction
2. Rules for double negation
3. Rules for implication
4. Rules for disjunction

Examples

$p \vee q \vdash q \vee p$

$q \rightarrow r \vdash p \vee q \rightarrow p \vee r$

$p \wedge (q \vee r) \vdash (p \wedge q) \vee (p \wedge r)$

Example

$p \to (q \to p)$

Rules for natural deduction

1. Rules for conjunction
2. Rules for double negation
3. Rules for implication
4. Rules for disjunction
5. Rules for negation

Contradiction

expressions of the form $\phi \land \neg\phi$ or $\neg\phi \land \phi$

denoted by $\perp$

they let you derive anything

bottom-elimination

Rules for natural deduction

1. Rules for conjunction
2. Rules for double negation
3. Rules for implication
4. Rules for disjunction
5. Rules for negation

Examples

$\neg p \vee q \vdash p \rightarrow q$

Examples

$\neg p \vee q \vdash p \to q$

$p \to q, p \to \neg q \vdash \neg p$

Examples

$\neg p \lor q \vdash p \to q$

$p \to q, p \to \neg q \vdash \neg p$

$p \to \neg p \vdash \neg p$

Recall our first 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. *Therefore*, there were taxis at the station.
- If p and not q, then r. Not r. p. *Therefore*, q.
- $p \wedge \neg q \rightarrow r, \neg r, p \vdash q$

Derived rules

- Modus Tollens (MT)

Derived rules

- Modus Tollens (MT)
- $\neg\neg$i

Derived rules

- Modus Tollens (MT)
- $\neg\neg$i
- proof by contradiction

Derived rules

- Modus Tollens (MT)
- $\neg\neg$i
- proof by contradiction
- law of excluded middle ($\phi \vee \neg\phi$ is true)

Examples

Use LEM to show the validity of $p \to q \vdash \neg p \vee q$

Provable equivalence

- $\phi \dashv\vdash \psi$
- $p \wedge q \rightarrow r \dashv\vdash p \rightarrow (q \rightarrow r)$
- $p \wedge q \rightarrow p \dashv\vdash r \vee \neg r$

PBC: Classical vs. Intuitionistic Logicians

What's next?

- Syntax of propositional logic
- Semantics of propositional logic
- Soundness and completeness

Syntax

- formulas are strings over propositional atoms, logical symbols and left- and right-brackets
- but not everything is allowed, of course; e.g. $(\neg)() \vee pq \rightarrow$ does not seem to make any sense
- we would like our formulas to be *well-formed*

Well-formed formulas

- propositional atoms are well-formed formulas
- if ϕ is well-formed, so is $(\neg\phi)$
- if ϕ and ψ are well-formed, so are $(\phi \wedge \psi)$, $(\phi \vee \psi)$, and $(\phi \rightarrow \psi)$
- nothing else is a well-formed formula

Grammar in BNF

$$\phi \ ::= \ p \mid (\neg\phi) \mid (\phi \wedge \psi) \mid (\phi \vee \psi) \mid (\phi \rightarrow \psi)$$

Parse-trees and subformulas

Semantics

- based on the truth value of atomic propositions, and how the logical connectives manipulate the truth values

- $\phi_1, \phi_2, \ldots, \phi_n \vDash \psi$

- truth tables

Example of $\models$ notation

Do the following hold?

- $p \wedge q \models p$
- $p \vee q \models p$
- $\neg q, p \vee q \models p$
- $p \models q \vee \neg q$

Mathematical Induction

Claim. For every well-formed propositional logic formula, the number of left-brackets is equal to the number of right-brackets.

Mathematical Induction

Claim. For every well-formed propositional logic formula, the number of left-brackets is equal to the number of right-brackets.

Induction on the height of the parse tree (*structural induction*)

Mathematical Induction

Claim. For every well-formed propositional logic formula, the number of left-brackets is equal to the number of right-brackets.

Induction on the height of the parse tree (*structural induction*)

base case: atomic formulas do not have any brackets

Mathematical Induction

Claim. For every well-formed propositional logic formula, the number of left-brackets is equal to the number of right-brackets.

Induction on the height of the parse tree (*structural induction*)

base case: atomic formulas do not have any brackets

inductive step: argue for all possible logical connectives as root

Soundness of propositional logic

If $\phi_1, \phi_2, \ldots, \phi_n \vdash \psi$ is valid

then it is inconceivable that there there is a valuation in which ψ is false, whereas $\phi_1, \phi_2, \ldots, \phi_n$ are all true.

induction on the length of the (natural deduction) proof

can be tricky though (because of the assumption boxes)

Soundness and Completeness

Soundness If $\phi_1, \phi_2, \ldots, \phi_n \vdash \psi$ is valid, then $\phi_1, \phi_2, \ldots, \phi_n \vDash \psi$ holds.

Completeness If $\phi_1, \phi_2, \ldots, \phi_n \vDash \psi$ holds, then $\phi_1, \phi_2, \ldots, \phi_n \vdash \psi$ is valid.

Semantic equivalence, Satisfiability, Validity

For propositional logic formulas ϕ and ψ, we say that they are semantically equivalent (denoted as $\phi \equiv \psi$) iff $\phi \vDash \psi$ and $\psi \vDash \phi$ hold.

ϕ is said to be valid if $\vDash \phi$ (tautologies are exactly the valid formulas)

ϕ is said to be satisfiable if it has a valuation in which it evaluates to true.

Semantic equivalence, Satisfiability, Validity

For propositional logic formulas ϕ and ψ, we say that they are semantically equivalent (denoted as $\phi \equiv \psi$) iff $\phi \vDash \psi$ and $\psi \vDash \phi$ hold.

ϕ is said to be valid if $\vDash \phi$ (tautologies are exactly the valid formulas)

ϕ is said to be satisfiable if it has a valuation in which it evaluates to true.

ϕ is satisfiable iff $\neg\phi$ is not valid.

Distributivity and De Morgan's Laws

Normal Forms

Negation Normal Form (NNF)

A well-formed formula (wff) is in NNF if it uses only $\vee$, $\wedge$, and *literals*.

- Every wff is logically equivalent to a wff in NNF.
- Exercise: convert $\neg(p \to (p \wedge q))$ into NNF.

Normal Forms

Disjunctive Normal Form (DNF)

A well-formed formula (wff) is in DNF if it is a disjunction of one or more terms, where each term is a conjunction of one or more literals.

Note: p, $(p \wedge q \wedge \neg r)$, and $(p \vee q)$ are all in DNF.

- Every wff is logically equivalent to a wff in DNF.
  How? From the truth-table, or using logical equivalences.

- Exercise: convert $\neg(p \to (p \wedge q))$ into DNF.

Normal Forms

Conjunctive Normal Form (CNF)

A well-formed formula (wff) is in CNF if it is a conjunction of one or more terms, where each term is a disjunction of one or more literals.

- Every wff is logically equivalent to a wff in CNF.
  How? From the truth-table, or using logical equivalences.
- Exercise: convert $\neg(p \rightarrow (p \wedge q))$ into CNF.

From truth tables

A	B	C	F
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Figure 1: Truth table for F

Why care about CNF formulas?

validity checking is easy (it otherwise takes time exponential in the no. of atoms)

consider $(\neg q \vee p \vee r) \wedge (\neg p \vee r) \wedge q$

$\models (\neg q \vee p \vee r) \wedge (\neg p \vee r) \wedge q$ holds iff

$\models (\neg q \vee p \vee r), \quad \models (\neg p \vee r), \quad \models q \quad$ all three hold

but that is easy to check:
<u>a disjunction of literals is valid iff they have a pair of complementary literals</u>

Propositional Resolution

- set representation of CNF formulas
- proof rule: $(\neg p \vee \phi), (p \vee \psi)$ resolve to give $(\phi \vee \psi)$
- derivation of $\square$ gives a refuation
- refutation is the way proofs are done
- e.g. $(x \vee \neg y), (y \vee z), (\neg x \vee \neg y \vee z) \vdash z$

  is proved by deriving $\square$ from $\{\{x, \neg y\}, \{y, z\}, \{\neg x, \neg y, z\}, \{\neg z\}\}$ using resolution

Resolution Lemma

Let F be a CNF formula represented as a set of clauses. Suppose R is a resolvent of two clauses C_1 and C_2 in F, then $F \equiv F \cup \{R\}$.

Soundness

If there is a derivation of $\Box$ from F then F is unsatisfiable.

Completeness

If F is unsatisfiable then there is a derivation of $\square$ from F.

Lecture notes on Resolution

`https://www.cs.ox.ac.uk/people/james.worrell/lec6-2015.pdf`

Thank you!