Introduction to Logic Programming

Subodh Sharma, 2021

Study of Logic!

What is logic? Why is it important?

- Logic is about reasoning
  - One has to show validity of arguments
  - Reasoning is sound only if it is impossible to draw false conclusions from true facts.
    - Sound reasoning can however produce true/false conclusions from false premises
- Logic is a tool to study computation systems: example law, PL, math etc.
  - For instance, one can cleanly identify circular reasoning in a mathematical proof!
- Formal logic can also be viewed as a formal language with syntax and semantics.

Example of Reasoning

- Forms of reasoning:
  - Deductive: the truth of premises guarantees the truth of the conclusion
    - Humans are mortal, Socrates is a human => Socrates is mortal
  - Inductive: Conclusions follow from premises with some probability; in other words we make broad generalisations from specific observations.
    - First note pulled is one rupee, second note pulled is again one rupee note => all notes in the purse are one rupee notes
  - Abductive: From some partial observations, create the most likely explanation
    - Eg: I saw milk spilled from a packet, I also saw a cat nearby => Cat spilled the milk
  - **We will restrict ourselves to deductive reasoning in this course.**

Propositional Logic

- Study of statements expressed in natural languages
  - Eg: Ostriches cannot fly, It rained today. In general assertions and denials can be considered as propositions.
- Syntax:
  - **A**: A countable set of proposition atoms
    - Assign a variable, say c, to Ostriches cannot fly. c is then a propositional atom.
  - $\Omega = \{ \perp, \top, \neg, \wedge, \vee, \rightarrow, \leftrightarrow \}$ as set of operators/connectives with well defined arities
  - Parenthesis (and)
  - $\mathscr{P} = \mathscr{T}_{\Omega}(A)$ is the smallest set generated from A and Ω

Propositional Logic

- Such that:
    - $\bot, \top \in \mathscr{P}$
    - If $p \in A$ then $p \in \mathscr{P}$
    - If $f \in \mathscr{P}$ then $\neg f \in \mathscr{P}$
    - If $f, g \in \mathscr{P}$ then $f \circ g \in \mathscr{P}$ where $\circ \in \Omega$
- In BNF: $\phi, \psi ::= \bot \mid \top \mid p \in A \mid (\neg \phi) \mid \phi \circ \psi$
    - The first three are atomic formulae (ie. $\bot, \top, v \in A$)
- Each expression of this language is callee a **well-formed formula** of $\mathscr{P}$

Propositional Logic

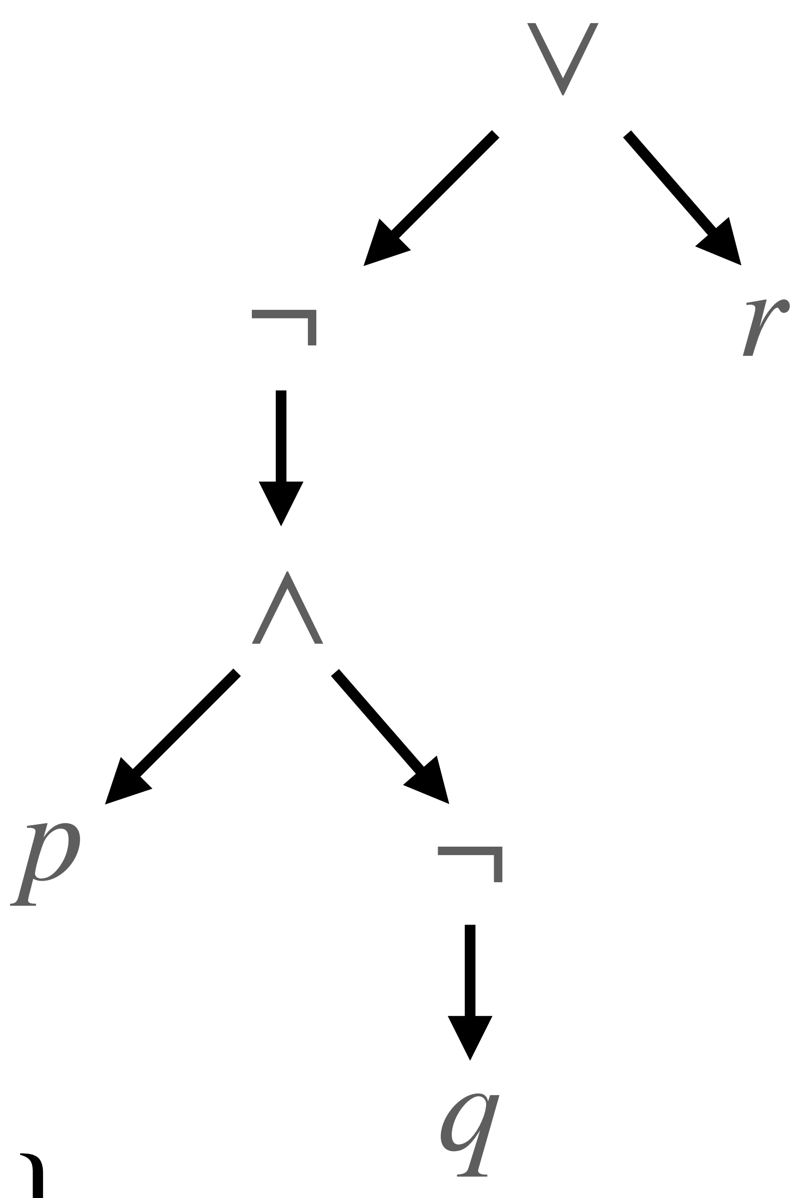
Associativity and Precedence

- Why do we need it?
  - So that we can translate an expression unambiguously into its AST (for parsing).
- Assuming the language to be fully parenthesised
  - Operators $\wedge$, $\vee$ are left associative
  - Operators $\rightarrow$, $\leftrightarrow$ are right associative
  - Precedence convention: $\leftrightarrow \prec \rightarrow \prec \vee \prec \wedge \prec \neg$

Propositional Logic

Identity, Subformula, Parsing

- Any two propositions ϕ and ψ which have the same AST are syntactically identical, $\phi \equiv \psi$
  - They may differ only in redundant parenthesis
- Consider $(\neg(p \land \neg q) \lor r)$:

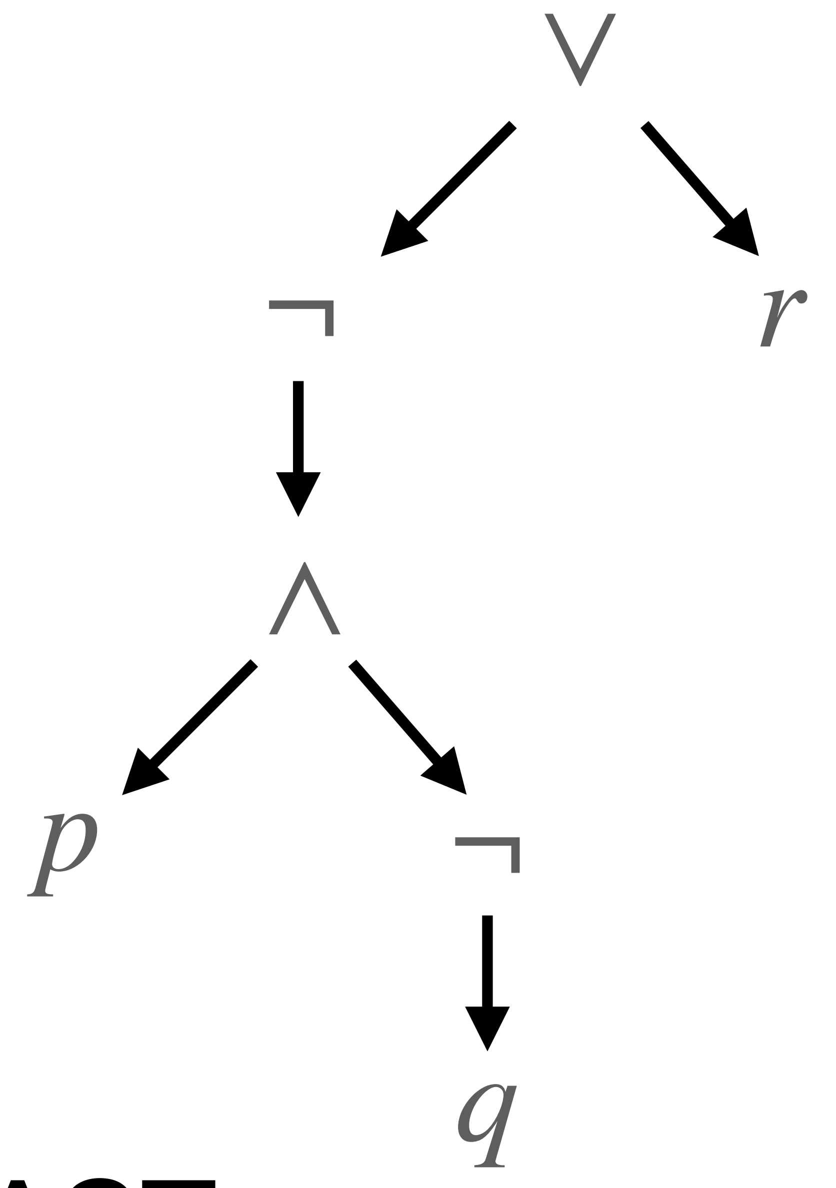


- Subformula: defined by induction on the structure of the formula
  - $SF(\bot) = \{\bot\}, SF(\top) = \{\top\},$
  - $SF(p) = \{p\}, SF(\neg\phi) = SF(\phi) \cup SF(\neg\phi)$
  - $SF(\phi \circ \psi) = SF(\phi) \cup SF(\psi) \cup \{\phi \circ \psi\}$, where $\circ \in \{\land, \lor, \rightarrow, \leftrightarrow\}$

Propositional Logic

Identity, Subformula, Parsing

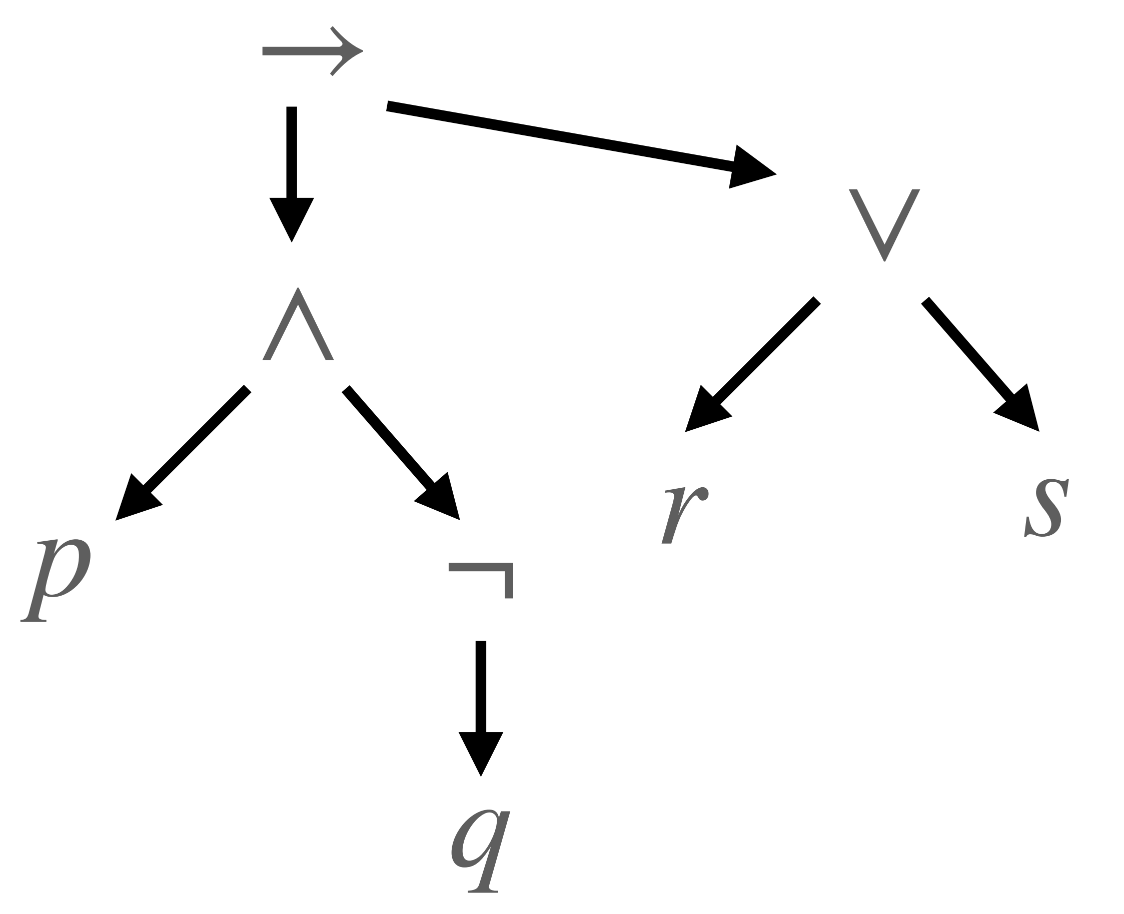
- Atoms of a formula are leaves in the AST of the formula. They can also be defined by induction on the structure of the formula.
  - $A(\perp) = \{\perp\}, A(\top) = \{\top\},$
  - $A(p) = \{p\}, A(\neg\phi) = A(\phi)$
  - $A(\phi \circ \psi) = A(\phi) \cup A(\psi)$, where $\circ \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$
- Atoms in the example?
- Immediate subformulas are the children of the formula in the AST
- Leading connective is the connective that is used to join the children



Propositional Logic

Identity, Subformula, Parsing

- Again consider: $p \wedge \neg q \rightarrow r \vee s$
- Parsing by precedence order:
    - $\neg$ has the highest precedence: thus $(\neg q)$
    - $\wedge$ has higher precedence than $\rightarrow$: thus, $(p \wedge (\neg q))$
    - $\vee$ has higher precedence than $\rightarrow$: thus $(r \vee s)$
    - Leading connective: $\rightarrow$



Propositional Logic

Identity, Subformula, Parsing

- Exercise: Which of the following can be unambiguously parsed without associativity?
  - $p \vee q \wedge r$
  - $\neg p \wedge q \leftrightarrow p \vee r$
- Exercise: Define the parsing algorithm given a formula ϕ. HINT: look at the inductive definition of subformulas!

Propositional Logic
Semantics

- Truth assignment or truth valuation is a function τ which assigns to each propositional atom a truth value $\in \{0,1\}$
  - $\tau : A \rightarrow \{0,1\}$
  - The semantics of propositions via $\mathcal{T}[\![.]\!]_\tau : \mathcal{P} \rightarrow \{0,1\}$
    - $\mathcal{T}[\![\bot]\!]_\tau \triangleq 0, \quad \mathcal{T}[\![\top]\!]_\tau \triangleq 1$
    - $\mathcal{T}[\![p]\!]_\tau \triangleq \tau(p), \quad \mathcal{T}[\![\neg\phi]\!]_\tau \triangleq \neg\mathcal{T}[\![\phi]\!]_\tau$ and so on.
    - The semantics are enumerated truth assignments in the form of a **truth table.**

Propositional Logic

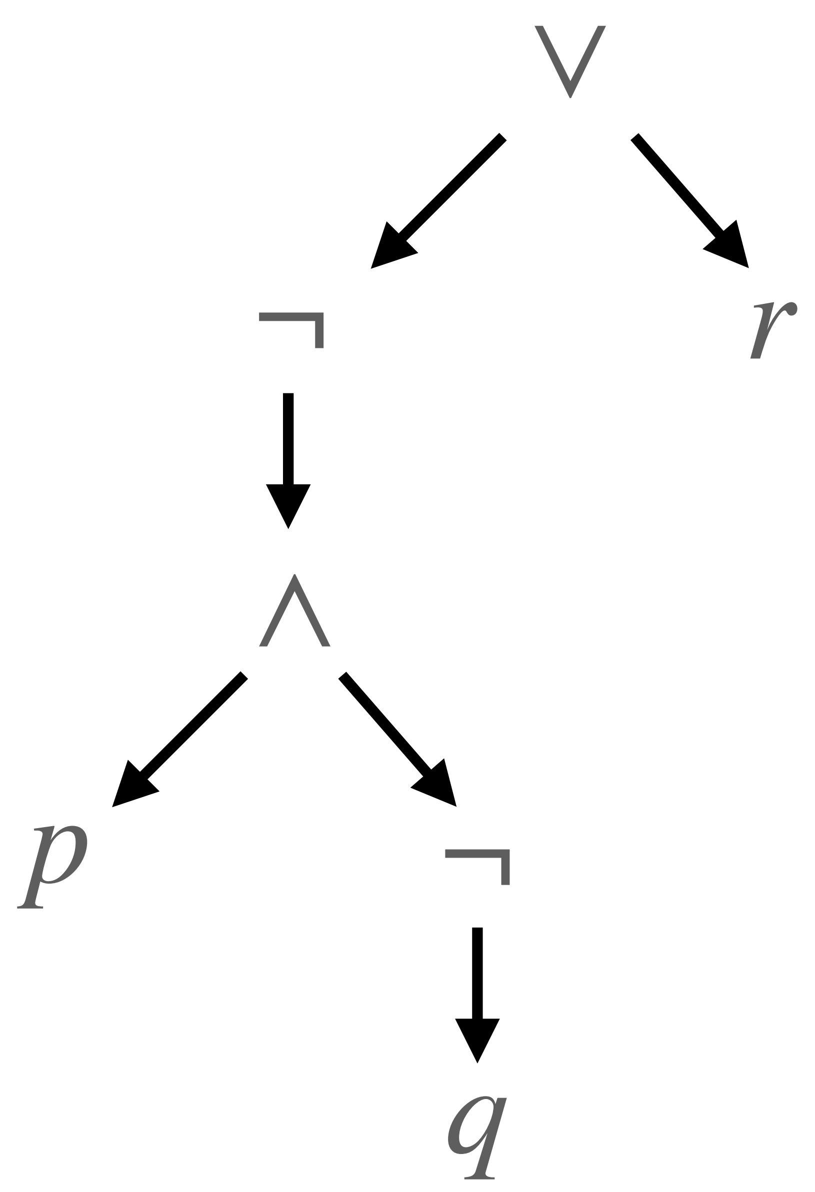
Models and Satisfiability

- A truth assignment τ is a model of a formula ϕ (denote by $\tau \vDash \phi$) iff $\mathcal{T}[\![\phi]\!]_\tau \triangleq 1$
- A formula is satisfiable if it has a model. Otherwise it is said to be unsatisfiable (or UNSAT).
- For any finite set $S = \{\phi_i \,|\, 1 \leq i \leq n\}, \tau \vDash S$ iff $\tau \vDash \phi_1 \wedge \ldots \wedge \phi_n$
- Satisfaction relation between models and formulas is the smallest relation that satisfies the following conditions:
  - $\tau \vDash p$ if $\tau(p) = 1$; $\tau \vDash \neg F$ if $\tau \nvDash F$; $\tau \vDash F_1 \vee F_2$ if $\tau \vDash F_1$ or $\tau \vDash F_2$ .. and so on.

Propositional Logic

Models and Satisfiability

- Eg: Consider $(\neg(p \wedge \neg q) \vee r)$ with model $\tau = \{p \mapsto 1, q \mapsto 1, r \mapsto 0\}$



Propositional Logic

Tautology, Contradiction, Contingent

- A proposition is said to be a :
  - Tautology if it is true under all possible truth assignments
  - Contradiction if it is false under all possible truth assignments
  - Contingent if it is both satisfiable as well as falsifiable.
  - Exercise:
    - If F is valid then $\neg F$ is ?
    - If F is a contradiction then $\neg F$ is?
    - If F is satisfiable then $\neg F$ is ?

Propositional Logic

Logical Consequence, Equivalence, Equisatisfiability

- A proposition ϕ is called a logical consequence of a set $\Gamma$of formulas (denote as $\Gamma \vDash \phi$) if any truth assignment that satisfies **all formulas** of Γ also satisfy ϕ.
  - When $\Gamma = \emptyset$, then logical consequence reduces to logical validity.
  - $\Gamma \nvDash \phi$ denotes that ϕ is not a logical consequence of Γ.
  - $\Gamma \vDash \phi$ iff $\bigwedge \psi_i \rightarrow \phi$ is **valid** where $\psi_i \in \Gamma$
- $F \equiv G$ if for each model τ, we have $\tau \vDash F$ iff $\tau \vDash G$
  - Eg: $(p \vee q) \vee r \equiv p \vee (q \vee r)$
- F and G are equisatisfiable if F is SAT iff G is SAT.

Propositional Logic

Normal Forms

- Conjunctive Normal Form (CNF)
  - Of the form $\bigwedge \delta_i$ where $\delta_i = \bigvee \gamma_j$
  - Analogously one can define DNF.
- Thm: Every formula in this logic is logically equivalent to its CNF
- Thm: Every formula in this logic is logically equivalent to its DNF

Propositional Logic

Tautology Checking

- Verifying validity of a formula by truth table can be quite tedious.
- Checking validity (or tautology) involves finding a falsifying assignment.
  - Given a formula in CNF of the form $\bigwedge \delta_i$ where $\delta_i = \bigvee \gamma_j$ where the literals in $\gamma_j = P_i \cup N_i$
  - Then γ_i can be falsified iff $P_i \cap N_i = \emptyset$
    - Complexity is linear in the size of a clause.