BITS Pilani Hyderabad Campus
CS F214 Logic in Computer Science,
I Semester 2021-2022
Lecture Notes
Lecture 17-22

We wish to show $\phi_1, \phi_2, ..\phi_n \vDash \psi$ holds i.e. $\phi_1, \phi_2, ..\phi_n \vDash \psi_1 \wedge \psi_2$ holds.

From (1) and (2), we have that for each valuation for which $\phi_1, \phi_2, ..., \phi_n$ is true ψ_1 is true also ψ_2 is true.

$\therefore$ for each valuation $\psi_1 \wedge \psi_2$ is true.
$\therefore$ we have,

$$\phi_1, \phi_2, ..\phi_n \vDash \psi$$

Case 2: When last rule applied is $\vee_e$. We must have proved on have as a premise $\eta_1 \vee \eta_2$.
This means that the sequent $\phi_1, \phi_2, ..\phi_n \vdash \eta_1 \vee \eta_2$ is valid.

The first 'box' $\boxed{\begin{array}{c} \eta_1 \\ . \\ . \\ . \\ \psi \end{array}}$ gives us a proof of the sequent

$$\phi_1, \phi_2, ..\phi_n, \eta_1 \vdash \psi \qquad 4$$

likewise, the second 'boxes' gives us

$$\phi_1, \phi_2, ..\phi_n, \eta_2 \vdash \psi \qquad 5$$

By induction hypothesis, the corresponding semantic entailment relations for (3),(4) and (5) holds. We shall call semantic entailment relations as (6),(7) and (8).
Consider an arbitrary evaluation for which $\phi_1, ...\phi_n$ are true.
By (6), we have that $\eta_1 \vee \eta_2$ is true for this valuation. This means that for this valuation at least η_1, η_2 is true.

Case(2a) η_1 is true.
By (7) we have that ψ is true.

Case (2b) η_2 is true.
By (8), we have that ψ is true.
Therefore in both the cases ψ is true.
$\therefore we have \phi_1, \phi_2, ...\phi_n \vDash \psi$ holds.

Rest of the cases can be shown similarly.

13.2 Contrapositive of The Soundness Theorem

If $\phi_1, ..\phi_n \vDash \psi$ does not hold then,$\phi_1, ..\phi_n \vdash \psi$ is not valid.

This implies that in order to show that a sequent is not valid, it suffices to find a valuation for which $\phi_1, ..\phi_n$ are true but ψ is false.

14 Completeness of Propositional Logic

Whenever $\phi_1, .., \phi_n \vDash \psi(1)$ holds then there exists a natural deduction proof for the sequent $\phi_1, .., \phi_n \vdash \psi$.

Proof Sketch

1. Assuming (1), we show that $\vDash \phi_1 \rightarrow (\phi_2 \rightarrow (..(\phi_n \rightarrow \psi))$ holds.
2. We show that $\vdash \phi_1 \rightarrow (\phi_2 \rightarrow (..(\phi_n \rightarrow \psi))$ is valid.
3. $\phi_1, .., \phi_n \vdash \psi$ is valid.

Please refer the textbook for the proof.

14.1 Corollary[Soundness and Completeness of Propositional Logic]

Let $\phi_1, ..., \phi_n$ and ψ be formulae of propositional logic. Then, $\phi_1, ..\phi_n \vdash \psi$ is valid.

14.2 Semantic Equivalence

Let ϕ and ψ be formulas in propositional logic. We say that, ϕ an ψ are semantically equivalent iff $\psi \vDash \psi$ holds and $\psi \vDash \phi$ holds as well.
We write $\phi \equiv \psi$
We call ϕ valid iff $\vDash$ holds. Semantic equivalence is identical to provable equivalence.
e.g. $p \rightarrow \equiv \neg q \rightarrow \neg p$
$p \rightarrow q \equiv \neg p \vee q$

We want to transform formulaes into forms in which validity checks are easy.
$\phi \rightarrow \psi \equiv \neg\phi \vee \psi$.

Definition: *A literal L is either an atom or negation of atom.*

A formula C in Conjunctive Normal Form (CNF) if it is a conjunction of the

clauses where each clause is a disjunction of literals.
e.g.

$$(1)\ (p \vee q) \wedge (\neg p \vee r)(2)(\neg(q \vee \neg p) \vee r) \wedge (p \vee q)$$

(2) is not in CNF form as it has negation of clause.

Definition of CNF in Backus Norm Form (BNF)

$$\begin{array}{rr} \text{Literal} & L ::= p|\neg p \\ \text{Clause} & D ::\neq L|L \vee D \\ \text{CNF Formula} & C ::= D|D \wedge C \end{array}$$

Observations:

1. A CNF is a conjunction of clauses $C_1, C_2, ..., C_n$
   i.e. $C \equiv C_1 \wedge C_2 \wedge ... \wedge C_n$.
   For C to be true it must be the case that each one of $C_1, C_2, ..., C_n$ are true.
   Suppose C_i is not a valid formula then C is not valid. Now there may be a single clause featuring all n atoms.

Lemma: *A disjunction of literals $L_1 \vee L_2 \vee .... \vee L_m$ is valid iff there are i,j such that $1 \leq i, j \leq m$, so that L_i is $\neg L_j$*

Proof: Consider i,j so that L_i is $\neg L_j$.

Now,

$$L_1 \vee L_2 \vee .. \vee L_m \equiv (L_1 \vee ...) \vee (L_i \vee \neg L_i) \quad (1)$$

Now, $(L_i \vee \neg L_i)$ is always true.
Suppose L_i is an atom P. In any valuations, p is either true or false.

Case 1: p is true
Then L_i is true.
$\therefore (L_i \vee \neg L_i)$ is true.
$\therefore$ (1) is true.

Case 2: p is false.
L_i is false, $\neg L_i$ is true.
$\therefore (L_i \vee \neg L_i)$ is true.
$\therefore$ (1) is true.
Suppose L_i is $\neg p$, the formula is valid for similar reasons.

To prove the converse, suppose for all i,j $1 \leq i, j \leq m$. L_i is not $\neg L_j$ then

The formula is $L_1 \vee L_2 \vee ... \vee L_m$.

Now, consider a valuation where each literal is made false. Suppose a literal L_i is P_k. Then set P_k false in the valuation.

Suppose L_i is $\neg P_l$. Then set P_l to true in the valuation.

This procedure will not make any literal evaluate to false because that would imply that such a literal is a negation of a literal that had previously been made false.

Definition: *Given a formula ϕ in propositional logic, we say that ϕ is satisfiable if it has a validation in which it evaluates to true.*

e.g.

$$(1) p \vee q \rightarrow p \text{ is satisfiable}$$
$$(2) (p \vee q) \wedge (\neg p) \wedge (\neg q)$$

Proposition: Let ϕ be a formula of propositional logic. Then ϕ is satisfiable iff $\neg\phi$ is not valid.

Proof: Suppose ϕ is satisfiable.

Then there exist a valuation of ϕ, in which ϕ evaluates to true. In this valuation $\neg\phi$ evaluates to false.

$\therefore \neg\phi$ is not valid.

To prove the converse, suppose $\neg\phi$ is not valid. Then, there exists a valuation for which $\neg\phi$ is false.

For the same valuation, we have ϕ evaluates to True.

Since $\phi \equiv \neg\neg\phi$.

$\therefore \phi$ is satisfiable.

Please read sec 1.5.2 from textbook regarding the conversion of any formula in CNF (Page 57).

The logical constants ('bottom') $\perp$ and ('top') $\top$ denote respectively unsatisfiable formula and tautology.

15 Horn Formula

A Horn formula is a formula ϕ in propositional logic, if it can be generated as instance of the following grammar.

$$P ::= \bot | \top | P$$
$$A ::= P | P \wedge A$$
$$\text{Horn Clause, } C ::= A \rightarrow P$$
$$\text{Horn Formula, } H ::= C | C \wedge H$$

Formula	Explanation
1.$(p \wedge q \wedge r \rightarrow p) \wedge (q \wedge s \rightarrow p) \wedge (\top \rightarrow s) \wedge (r \wedge s \rightarrow \bot) \wedge (\bot \wedge p \rightarrow r)$	Horn Formula
2.$(p \wedge q \wedge r \rightarrow \neg p) \wedge (q \wedge r \rightarrow q)$	Not Horn Formula due to $\neg p$
3. $(p \wedge r \wedge r \rightarrow \bot) \wedge (\neg q \wedge r \rightarrow p)$	Not Horn Formula due to $\neg q$
4. $(p_1 \wedge p_2 \wedge p_3 \rightarrow (p_4 \wedge p_5)) \wedge (\top \rightarrow p_5)$	Not Horn Formula due to $p_4 \wedge p_5$
5. $(p \wedge q \rightarrow r) \wedge (p \wedge q) \wedge (r \vee s \rightarrow p)$	Not Horn Formula due to $\vee$ and $p \wedge q$

15.1 Deciding Satisfiability of Horn Formula

- Maintain a list of all occurrences of type P in your formula.
  1. It marks $\top$, if it occurs in that list.
  2. If there is a conjunct $p_1 \wedge p_2 \wedge .. \wedge p_k \rightarrow p'$ of ϕ such that all p_j with $1 \leq j < k$ is marked, then mark p' as well and go to step 2. Otherwise if there is no such conjunct go to step 3.
  3. If $\bot$ is marked, print 'Unsatisfiable' and Stop. Otherwise go to step 4.
  4. Print 'Satisfiable'.

Example

1.$(p \wedge q \wedge w \rightarrow \bot) \wedge (t \rightarrow \bot) \wedge (r \rightarrow p) \wedge (\top \rightarrow r) \wedge (\top \rightarrow q) \wedge (u \rightarrow s)$

Solution:

- Mark all occurrences of $\top$.
- Mark r, q, u.
- Mark p.

- Mark S.
- Print Satisfiable.

Example

$$2.(p_5 \to p_{11}) \wedge (p_2 \wedge p_3 \wedge p_5 \to p_{13}) \wedge (\top \to p_5) \wedge (p_5 \wedge p_{11} \to \bot)$$

Solution:

- Mark all occurrences of $\top$.
- Mark $p_5, p_1 1, \bot$
- Print 'Unsatisfiable'.

Example: **1(a).Construct a proposition for the given truth table.**

p	q	ϕ
F	F	T
F	T	T
T	F	F
T	T	F

- The disjunctive normal form formula can be constructed as $(\neg p \wedge \neg q) \vee (\neg p \wedge q)$
- In this example, we have considered all the cases for which the proposition is true and created set of clauses by anding them.
- Later all the clauses are combined using disjunction.

Example: **1(b).Prove the converse of the previous example.**

- The DNF clause consists of set of clauses combined using disjunction.
- All these clauses have set of atoms which are combined using conjunction.
- If the truth table is true then there is at least one clause with all atoms true.
- Now, for all the cases where the proposition evaluates to false there is at least one atom which differs to the evaluation where the formula is true.
- $\therefore$ the corresponding DNF clauses also evaluates to false as at least one of the atom in the clause evaluates to false.
- Thus, DNF becomes disjunction of all clauses which evaluates to false and thus formula evaluates to false.

Example: **2.Convert given truth table to CNF formula.**

Solution:

- Consider truth table for $\neg\phi$.
- Construct proposition in DNF for $\neg\phi$
- Take negation of DNF formula and apply Demorgan's Law to convert it into CNF.

p	q	$\neg\phi$
F	F	F
F	T	F
T	F	T
T	T	T

$$\neg\phi \equiv (p \wedge \neg q) \vee (p \wedge q)$$
$$\phi \equiv \neg(p \wedge \neg q) \wedge \neg(p \wedge q)$$
$$\boxed{\phi \equiv (\neg p \vee q) \wedge (\neg p \vee \neg q)}$$