Lecture 1 and 2

Logic formalises valid methods of reasoning.

Ex.

1. if the train arrives late (p) and there is no taxi ($\neg q$) then John is late for his meeting (r).

1. p and $\neg q$ implies r.

2. John is not late and train arrived late

1 and 2 implies there is a taxi

2. $\neg r$ and p ; 1 and 2 implies q.

3. If it is raining (p) and John does not have his umbrella ($\neg q$) then John will get wet (r)

4. it rained (p) and John did not get wet ($\neg r$)

3 and 4 implies John had his umbrella.

p : train arrives late.
q : There is a taxi
r : John is late for his meeting.

$p \wedge \neg q$

> p and $\neg q$ implies r
> p and $\neg r$
> Deduce : q

Propositional Logic

Syntax.

Countably infinite set of atomic propositions.

$$P = \{p_0, p_1, \ldots\}.$$

Logical Connectives : $\neg$ (not), $\vee$ (or) disjunction, $\wedge$ (and) Conjunction

The set Φ of formulas of propositional logic is the smallest set satisfying the following conditions.

- Every atomic proposition $p \in \Phi$
- if $\alpha \in \Phi$ then $\neg \alpha \in \Phi$
- if $\alpha, \beta \in \Phi$ then $\alpha \vee \beta \in \Phi$
- if $\alpha, \beta \in \Phi$ then $\alpha \wedge \beta \in \Phi$.

Binary : $\neg$ followed by $\vee, \wedge$

Ex. $\neg \alpha \vee \beta \triangleq ((\neg \alpha) \vee \beta)$

$\underline{p \wedge \neg q}$; $\wedge q$; $p \neg q$

T - True , $\perp$ - False.

Semantics.

Valuation v is a function $v : P \to \{T, \perp\}$.

$$v \subseteq P = \{p \mid v(p) = T\}.$$

Extend the function $v : P \to \{T, \perp\}$, to $\hat{v} : \Phi \to \{T, \perp\}$.

For $p \in P$, $\hat{v}(p) = v(p)$

For α of the form $\neg\beta$ $\hat{v}(\alpha) = \begin{cases} T & \text{if } \hat{v}(\beta) = \perp \\ \perp & \text{otherwise.} \end{cases}$

For α of the form $\beta \vee \gamma$ $\hat{v}(\alpha) = \begin{cases} \perp & \text{if } \hat{v}(\beta) = \hat{v}(\gamma) = \perp \\ T & \text{otherwise.} \end{cases}$

For α of the form $\beta \wedge \gamma$ $\hat{v}(\alpha) = \begin{cases} T & \text{if } \hat{v}(\beta) = \hat{v}(\gamma) = T \\ \perp & \text{otherwise.} \end{cases}$

$$\hat{v} = \{\alpha \mid \hat{v}(\alpha) = T\}.$$

$v \vDash \alpha$

- $v \vDash p$ iff $v(p) = T$
- $v \vDash \neg\alpha$ iff $v \nvDash \alpha$
- $v \vDash \beta \vee \gamma$ iff $v \vDash \beta$ or $v \vDash \gamma$
- $v \vDash \beta \wedge \gamma$ iff $v \vDash \beta$ and $v \vDash \gamma$.

Satisfiability

A formula α is satisfiable if there is valuation v such that $v(\alpha) = T$. Notation $v \models \alpha$

α is satisfiable.
v satisfies α.

Validity

α is valid if $v \models \alpha$ for every valuation v.

Notation $\models \alpha$ – α is valid

Tautologies – valid formulas.

Proposition. Let $\alpha \in \Phi$ α is valid iff $\neg\alpha$ is not satisfiable.

Proof. α is not valid iff $\exists v$ s.t $v(\alpha) = \bot$
$v \not\models \alpha$
iff $v(\neg\alpha) = T$ iff $\neg\alpha$ is satisfiable.

Question. Given α, an algorithm to check if $\models \alpha$

Ex. p – satisfiable $\quad p \vee \neg p$ – valid

$p \wedge \neg p$ – not satisfiable.

Vocabulary $\quad Voc(\alpha)$.

- For $p \in P$ $\quad Voc(p) = \{p\}$.
- If $\alpha \equiv \neg\beta$, $Voc(\alpha) = Voc(\beta)$
- if $\alpha \equiv \beta \vee \gamma$, $Voc(\alpha) = Voc(\beta) \cup Voc(\gamma)$.
- if $\alpha \equiv \beta \wedge \gamma$, $Voc(\alpha) = Voc(\beta) \cup Voc(\gamma)$

Proposition. Let $\alpha \in \Phi$ v_1, v_2 be valuations.

If v_1 and v_2 agree on $Voc(\alpha)$ then $\underline{v_1(\alpha) = v_2(\alpha)}$

$v_1 \models \alpha$ iff $v_2 \models \alpha$.

Derived Connectives.

$$\alpha \supset \beta \triangleq \neg \alpha \vee \beta$$

$$\alpha \equiv \beta \triangleq (\alpha \supset \beta) \wedge (\beta \supset \alpha)$$

$$\alpha \wedge \beta \triangleq \neg(\neg \alpha \vee \neg \beta)$$

Logical Connectives. $\neg$ (not), $\vee$ (or)

Lecture 3

Syntax

$$\Phi(P) ::= p \in P \mid \neg\alpha \mid \alpha \vee \beta$$

negation Disjunction

Semantics

Valuation $v : P \to \{T, \bot\}$ $\quad v \subseteq P$

$\hat{v} : \Phi \to \{T, \bot\}$ $\quad \hat{v} \subseteq \Phi$

v satisfies α ($\in \Phi$) if $\hat{v}(\alpha) = T$

$v \models \alpha$ [v satisfies α
v models α].

Derived Connectives

$\alpha \wedge \beta$

$\alpha \supset \beta \triangleq \neg\alpha \vee \beta$
$\alpha \to \beta$

$\alpha \equiv \beta \triangleq (\alpha \to \beta) \wedge (\beta \to \alpha)$
$\alpha \leftrightarrow \beta$

Implication.
$v \models \alpha \to \beta$
if $v \not\models \alpha$ then $v \models \alpha \to \beta$.
if $v \models \alpha$ and $v \models \beta$ then $v \models \alpha \to \beta$
$v \models \alpha$ and $v \not\models \beta$ then $v \not\models \alpha \to \beta$.

α is satisfiable if there exist $v \in V$ s.t $v \models \alpha$

$v(\alpha) = T$

α is valid if for all $v \in V$, $\left.\begin{matrix} v \models \alpha \\ v(\alpha) = T \end{matrix}\right\} \models \alpha$

$P \vee \neg P$ - valid $\quad$ P - satisfiable but not valid

$P \wedge \neg P$ - not satisfiable.

Example.

1. $(p \to \neg q) \to (q \vee \neg p) = \alpha_1$ Satisfiable / Not valid

$v = \{p, q\}$ $\quad v \not\models p \to \neg q \quad v \models (p \to \neg q) \to (q \vee \neg p)$

$v = \{q\}$ $\quad v \models \alpha$.

$v = \phi$ $\quad v \models \alpha$.

$v = \{p\}$ $\quad v \models p \to \neg q \; ; v \not\models (q \vee \neg p)$

$v \not\models \alpha$.

2. $(\neg p \vee q) \to r = \alpha_2$ Satisfiable / Not Valid

$v = \{p, q, r\}$ $\quad v \models \alpha_2$

$v = \{q, r\}$ $\quad v \models \alpha_2$

$v = \{p\}$ $\quad v \not\models \neg p \vee q \; ; v \models \alpha_2$

$v = \{q\}$ $\quad v \models \neg p \vee q \; ; v \not\models r$, Thus $v \not\models \alpha_2$

3. $\alpha_3 = (p \to (q \to r)) \vee (p \to q)$

Question: Is α_3 valid? $\quad T \to \bot$

For all $v \in V$, does $v \models \alpha$

Suppose α_3 is not valid.

$\exists v \in V$ s.t $v \not\models \alpha_3$.

$v \not\models p \to (q \to r)$ and $v \not\models p \to q$

$v \models p$ and $v \not\models q \to r$ $\qquad \underbrace{v \models p \text{ and } v \not\models q}_{v(p) = T \,;\; v(q) = \bot}$

$\underbrace{v \models q \text{ and } v \not\models r}_{V(q) = T;\; v(r) = \bot}$

$V(p) = T$; $V(q) = T$; $v(r) = \bot$

Contradicts the assumption that α_3 is not valid.

Therefore, α_3 is valid.

$$\alpha_4 = (p \leftrightarrow ((\neg q) \vee r)) \rightarrow (\neg p \rightarrow q)$$

Question. Is α_4 valid?

Suppose α_4 is not valid. $\exists v \in V$ s.t $v \not\models \alpha_4$.

$v \models p \leftrightarrow ((\neg q) \vee r)$ and $v \not\models \neg p \rightarrow q$

$v \not\models \neg q \vee r$ $\qquad$ $v \models \neg p$ and $v \not\models q$.

$v(p) = \bot$ and $v(q) = \bot$

Contradiction to the assumption that $\exists v \in V$ s.t $v \not\models \alpha_4$.

Thus α_4 is valid.

Question. Given α, can you check if $\models \alpha$?

$Voc(\alpha)$

Enumerate all valuations v over $Voc(\alpha)$

Check if $\underbrace{v \models \alpha}$.

Polynomial time algorithm.

$|\alpha| = n$ $\underline{2^n. - \text{number}}$ of valuation.

Not an efficient algorithm.

BNF

Normal Forms.

$$\Phi ::= p \in P \mid \neg\alpha \mid \alpha \vee \beta$$

Conjunctive Normal Form. (CNF)

Literal - either a proposition p or its negation $\neg p$

Clause - is a disjunction of literals.

$$l_1 \vee l_2 \vee \cdots \vee l_m.$$

Example

$p, \neg p$

$p \vee q$, $p \vee \neg q \vee r$

α is in CNF if α is a conjunction of clauses.

Ex. $(\neg p \vee q) \wedge (r \vee \neg p \vee \neg q)$ is a CNF formula.

$$\alpha = C_1 \wedge C_2 \wedge \cdots \wedge C_n.$$

$$L := p \mid \neg p$$

$$D := L \mid L \vee D$$

$$C := D \mid D \wedge C$$

Lemma. A clause (disjunction of literals) $l_1 \vee l_2 \cdots \vee l_m$ is valid iff $\exists i,j : 1 \leq i,j \leq m$ s.t l_i is $\neg l_j$.

Proof. $\Leftarrow$ From the definition

$\Rightarrow$ Suppose no literal l_k has a matching negation in $\{l_1 \cdots l_m\}$.

For each $k : 1 \leq k \leq m$ assign $\perp$ to p if $l_k = p$.
assign T to p if $l_k = \neg p$.

Eg. $p \vee \neg q \vee r$ $\quad v(p) = \perp \;\; v(r) = \perp, \; v(q) = T$

$v \not\models p \vee \neg q \vee r$.

Theorem. A CNF $\alpha = C_1 \wedge C_2 \wedge \cdots \wedge C_n$ is valid iff $\forall i : 1 \leq i \leq n$, C_i is valid

Question. Given α, how to check if $\models \alpha$?

if α is in CNF

Question. For $\alpha \in \Phi$ does there exists β in CNF s.t α and β are "semantically equivalent".

$\models \alpha \leftrightarrow \beta$ is valid.