Date: _____ Day: M T W T F S

Normal Form:

↳ structured representation of logical expression

- literals (variables) — A, ¬A (negation)
- logical operators (AND, OR), NOT)

First, you all know some important rules/laws:

→ Indempotent law ($P \wedge P \leftrightarrow P$ and $P \vee P \leftrightarrow P$)

→ commutative law ($P \wedge q \leftrightarrow q \wedge P$ and $P \vee q \leftrightarrow q \vee P$)

→ Associative law $(P \vee q) \vee r = P \vee (q \vee r)$

→ De-Morgen law $\sim(P \wedge q) \leftrightarrow \sim P \vee \sim q$ (AND)

$\sim(P \vee q) \leftrightarrow \sim P \wedge \sim q$

Also,

$P \rightarrow q \leftrightarrow \sim P \vee q$

→ Disjunction Normal Form:

↳ logical Formula consisting of a ~~disjun~~ disjunction (OR, V) of conjunctions (AND, ∧) b/w literals.

Structure

$(A \wedge \neg B) \vee (\neg A \wedge C) \vee (B \wedge C)$

A ↑ AND ↓ OR

Date: _____ Day: M T W T F S

Examples:

→ $(A \wedge B) \vee (A \wedge \neg C)$ ✓

→ $A \wedge (B \vee C)$ ✗

→ $A \wedge (B \vee C) = (A \wedge B) \vee (A \wedge C)$ ✓

Suppose

convert it into DNF

$(P \rightarrow q) \wedge (\neg p \wedge q)$

As: $P \rightarrow q \leftrightarrow \neg p \vee q$

So; $(\neg p \vee q) \wedge (\neg p \wedge q)$

Apply distributive law;

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

So; $(\neg p \wedge q) \vee (q \wedge \neg p)$

→ Conjunction Normal Form:

Opposite of DNF

↳ AND of OR

Conjuction of Disjunction

Structure:

$(A \vee \neg B) \wedge (\neg A \vee C) \wedge (B \vee C)$

Date: _____ Day: M T W T F S

used in

↳ SAT solver

→ Automated theorem proving

→ Boolean Algebra.

Examples

→ $(\neg P \vee q) \wedge (\neg p \vee r)$ ✓

convert it into CNF

$(P \wedge q) \vee (\neg P \wedge q \wedge r)$

using distributive law:

$(P \vee (\neg P \wedge q \wedge r)) \wedge (q \vee (\neg P \wedge q \wedge r))$

$[(P \vee \neg P) \wedge (P \vee q) \wedge (P \vee r)] \wedge [(q \vee \neg P) \wedge (q \vee q) \wedge (q \vee r)]$

$[(P \vee q) \wedge (P \vee r)] \wedge [(q \vee \neg P) \wedge q \wedge (q \vee r)]$

Now: Complex Example of DNF

$P \vee (\neg P \rightarrow (q \vee (q \rightarrow \neg r)))$

$P \vee (\neg P \rightarrow (q \vee (\neg q \vee \neg r)))$

$P \vee (P \vee (q \vee (\neg q \vee \neg r)))$

$P \vee (P \vee ((q \vee \neg q) \vee (q \vee \neg r)))$

$P \vee (P \vee (q \vee \neg r))$

$P \vee (P \vee q \vee P \vee \neg r)$

$P \vee (P \vee q \vee \neg r)$

$P \vee P \vee P \vee q \vee P \vee \neg r \Rightarrow P \vee q \vee \neg r$

Date: _____ Day: M T W T F S

Horn clause:

clause

↳ expression that has a finite collection of literals (variables)

Horn clause:

↳ disjunction (OR, ∨) of literal in which at most one positive literal; and

Example:

$\neg A_1 \vee \neg A_2 \neg A_3 \dots \neg A_n \vee B$

Form of Horn clauses:

→ clauses

↳ exact one positive and rest are negative literals

$\neg A_1 \vee \neg A_2 \dots \neg A_n \vee B$

→ Fact:

Single positive, no negative literal

B

(Just a Fact)

Date: _____ Day: M T W T F S

→ Goal clause (Query)

only negative literal

$\neg A \vee \neg B$

(make a query)

Negative is always left
Positive is right.

Examples

Forms	Horn clause	Meaning
Implication	$\neg A \vee B$	$A \Rightarrow B$
Fact	Rain (B)	"It is raining"
Rule	$\neg Hot \vee \neg summer \vee Tired$	$(Hot \wedge summer) \Rightarrow Tired$
Query	$\neg Rain$	is it raining

Forward chaining; & Backward chaining

(Facts → Rules → Goals)		(Goals → Rules → Facts)
→ Data Driven		→ Goal Driven
→ Data is available		→ Goal state is given.
↓ (Data)		↓ (Goal)
Data	Rules	Conclusion
$x = 1$	If ($x == 1$ & $y == 2$)	$a = 4$
$y = 2$	Then $z = 3$;	
	if $z == 3$:	
	Then $a = 4$;	

Date: Day: M T W T F S

Here, I

Inference Engine: ↳ used it
↳ Forward & Backward
→ and it's core component of expert system and rule based AI models, to drive data to reached at a goal.

(Example of FC)

→ Fact → patient has fever
→ Fact → patient has a sore throat
→ Rule → If patient has fever & sore throat, they might have flu.

↓

Conversion of Facts into (FOL)

Fact → Fever(x)

Rule → Fever(x) ∧ soreThroat(x) → Flu(x).

Date: _____ Day: M T W T F S

Example of Backward chaining

① → Goal → Does patient have flu?

② → Rule → If patient have fever & sore throat they might have flue.

③ → sub goals:

→ verify patient has fever

→ " " sore throat

→ **Knowledge Engineering in (FOL)**

↳ building a knowledge based system, where a system can reason, Infer and solve problem based on formal logic

→ **First order logic (FOL)**

↳ extend propositional logic by including

→ constants (like, Shamzy etc)

→ variables (x, y, z, --)

→ Predicates (loves (Shamzy, x))

→ Functions (Father of (x))

Date: Day: M T W T F S

→ Quantifiers:
- → Universal ∀ (for all)
- → Existential ∃ (there exist)

Steps
- ↳ Identify Domain / Task:
  - → (what do you want the system to do)
- ↳ Assemble Relevant Knowledge:
  - → (Gather all info about Domain)
- ↳ Represent Knowledge in FOL:
  - → (convert real word facts into FOL)

  Like: "All humans are mortal"

  $\forall x$ (Human(x) → Mortal(x))
- → Use Inference to Drive New Knowledge:
  - use rules of Inference.
- ↳ Implement in Reasoning Engine:
  - use logic Programming language
  - build a reasoning Engine.
- → Accept FOL (KB)
- → Applies Inference Rules
- ↳ Answer Query