Knowledge representation

- Intelligent agents need knowledge about the world in order to reach good decisions.
- Declarative knowledge is represented in a form of sentences in a knowledge representation language and stored in a knowledge base.
- Knowledge base is used by an inference engine to infer a new sentence which will be used for the agent to decide what action to take next.
- Formal languages are defined by its grammar and semantic rules
    - **grammar**: defines the syntax of legal sentences
    - **semantic rules**: defines the meaning.
- Common knowledge representation formalisms:
    - **Propositional logic**
    - **First-order logic**
    - Fuzzy logic
    - Semantic networks
    - Ontologies

Propositional logic

- a declarative language in which can handle propositions that are known true, known false, or completely (unknown true or false)
- a BNK (Backus-Naur Form) grammer of sentences in propositional logic.
    - $\neg$ = NOT
    - $\wedge$ = AND
    - $\vee$ = OR
    - $\implies$ = IMPLIES
    - $\iff$ = IF AND ONLY IF
- Negation: a sentence using $\neg$ is called negation
- Literal: either an atomic sentence or a negated atomic sentence
- Conjunction: two sentences connected by $\wedge$. Eac hof them is called **conjunct**
- Disjunction: two sentences connected by $\vee$. Each of them is called **disjunct**
- Implication: two sentences connected by $\implies$. $P \implies Q$. P is called premise or antecedent and Q is the conclusion or consequent.
    - An implication is `if-then` statement.

Truth tables

P	Q	¬P	P ∧ Q	P ∨ Q	P $\implies$ Q	P $\iff$ Q
T	**T**	F	T	T	T	T
T	**F**	F	F	T	F	F
F	**T**	T	F	T	T	F
F	**F**	T	F	F	T	T

FOL, First-order logic

- a declarative language
- Syntax of FOL builds on that of propositional logic
- terms to represent objects, universal quantifier, and existential quantifier
- a model in FOL must provide the information required to determine the truth value of every atomic sentence in the language.
- $\forall x P(x)$ means for all x, P(x) is true
- $\exists x P(x)$ means there exists an x such that P(x) is true

Logical equivalence

logical equivalence	meaning
$(\alpha \wedge \beta) \equiv (\beta \wedge \alpha)$	Commutativity of $\wedge$
$(\alpha \vee \beta) \equiv (\beta \vee \alpha)$	Commutativity of $\vee$
$(\alpha \wedge (\beta \wedge \gamma)) \equiv ((\alpha \wedge \beta) \wedge \gamma)$	Associativity of $\wedge$
$(\alpha \vee (\beta \vee \gamma)) \equiv ((\alpha \vee \beta) \vee \gamma)$	Associativity of $\vee$
$\neg(\neg\alpha) \equiv \alpha$	Double negation elimination
$(\alpha \implies \beta) \equiv (\neg\alpha \vee \beta)$	Implication elimination
$(\alpha \implies \beta) \equiv (\neg\beta \implies \neg\alpha)$	Contraposition
$(\alpha \iff \beta) \equiv ((\alpha \implies \beta) \wedge (\beta \implies \alpha))$	Biconditional elimination
$\neg(\alpha \wedge \beta) \equiv (\neg\alpha \vee \neg\beta)$	De Morgan's law for $\wedge$
$\neg(\alpha \vee \beta) \equiv (\neg\alpha \wedge \neg\beta)$	De Morgan's law for $\vee$
$(\alpha \wedge (\beta \vee \gamma)) \equiv ((\alpha \wedge \beta) \vee (\alpha \wedge \gamma))$	Distributivity of $\wedge$ over $\vee$

logical equivalence	meaning
$(\alpha \lor (\beta \land \gamma)) \equiv ((\alpha \lor \beta) \land (\alpha \lor \gamma))$	Distributivity of $\lor$ over $\land$

$$\forall x \neg P(x) \equiv \neg \exists x P(x)$$

- For all x, not P(x) is logically equivalent to "it is not the case that there exists an x such that P(x) is true."

Reasoning

Deductive reasoning

- a process of reasoning from one or more statements (premises) to reach a logical conclusion.
- first premise, second premise, therefore conclusion.

Inductive reasoning

- the process of resoning from specific observations to borader generalizations and theories.
- also described as a method where one's experiences and observations are synthesized to cmoe up with a general truth.
- premises and then conclusion

Inference

- steps in reasoning
- moves from premises to logical consequences
- In AI Context, inference is to derive new logical sentences (as the conclusion) from existing logical sentences (as premises).
- researchers develop automated inference systems to emulate human inference.

Inference Problem

$$KB \models \gamma$$

- KB, Knowledge Base, is a set of propositions that represent what is known about the world.
- γ, Query sentence, is the target conclusion which needs to be confirmed based on the given KB.
- where $\models$ denotes the relation of logical entailment between KB and the sentence γ, reading as "KB entails γ" or "if KB is true, then γ must also be true".

- $\alpha \models \beta \iff M(\alpha) \subseteq M(\beta)$
  - $M(\alpha)$ is the set of all models that satisfy α.
  - $M(\beta)$ is the set of all models that satisfy β.
  - The statement $\alpha \models \beta$ means that in every model where α is true, β is also true. In other words, if α holds, then β must also hold.
- Model Checking: enumerates all possible models and checks if the entailment holds in each model.
  - Knowledge base will be used to draw inferences.
  - Query sentence, γ, is needed to be checked whether it is entailed by the KB.
  - Symbols, a list of all symbols (or atomic propositions) used in the problem context.
  - Models, assignments of truth and false values to those identified symbols.
- Model Checking Procedure
  i. Identify the propositional symbols involved in the KB sentences and query sentence
  ii. Enumerate all possible models by assigning truth values to the identified symbols.
  iii. Evaluate the KB sentence in each model and fined the models in which KB is true.
  iv. Evaluate the query sentence in the models from step 3 and check if query sentence is true in these models.
  v. Conclude that the KB entails the query sentence γ if and only if the query sentence is true in all models where the KB is true.

Inference Problem Example

- Obersvation **P**: "It is raining"
- Query sentence **Q**: "The ground is wet"
- Knowledge: $P \implies Q$ (If it is raining, then the ground is wet)
- Knowledge Base **KB**: $P \land (P \implies Q)$
- Inference Problem: $KB \models Q$ from KB=T, get Q=T
- Proof:
  i. From the 4 possible models, only Model 1 makes KB true.
  ii. M(KB) = model 1
  iii. Model 1 also makes Q true.
  iv. M(Q) = model 1
  v. $M(KB) \subseteq M(Q)$, therefore $KB \models Q$

Model	P	Q	P $\implies$ Q	P $\land$ (P $\implies$ Q)
1	**T**	**T**	T	**T**
2	**T**	F	F	F

Model	P	Q	$P \implies Q$	$P \land (P \implies Q)$
3	F	**T**	T	F
4	F	F	T	F

Inference by theorem proving

> to apply rules of inference directly to the sentences in the KB to construct a proof of the desired sentence without consulting models.

- Proof: a chain of consequences that leads to the desired goal.
- KB will be used to draw inferences.
- Desired sentence is needed to be checked whether it is entailed by the KB.
- The rules of inference are the approved logical equivalences and rules.

Aspect	Model Checking	Theorem Proving
방식	참/거짓으로 실제 계산	논리 규칙을 사용해 증명
예시	진리표	Modus Ponens, Resolution
장점	단순함	복잡한 문장도 처리 가능
단점	계산 많음	규칙 익혀야 함

Modus Ponens Rule

$$\frac{\alpha \implies \beta, \alpha}{\therefore \beta}$$

- whenever any sentences of the form $\alpha \implies \beta$ and α are given, then the sentence β can be inferred.

Add-Elimination Rule

$$\frac{\alpha \land \beta}{\therefore \alpha}$$

- from a conjunction, one of the conjuncts can be inferred.

Terms to contradiction and resolution

```
CNF
┌──────────────────────────────────────────────┐
│            (P ∨ Q)         ∧     (¬Q ∨ R)    │
│   ┌──────────────────────┐   ┌─────────────┐ │
│   │  Clause 1            │   │  Clause 2   │ │
│   │  (P ∨ Q)             │   │  (¬Q ∨ R)   │ │
│   │  ┌───────┬───────┐   │   │  ┌─────┬───┐ │
│   │  │  P    │   Q   │   │   │  │ ¬Q  │ R │ │
│   │  └───────┴───────┘   │   │  └─────┴───┘ │
│   └──────────────────────┘   └─────────────┘ │
└──────────────────────────────────────────────┘
```

- Literal: an atomic sentence or its negation
- Complementary literals: a literal and its negation are complementary literals
- Clause: an expression formed from a collection of finite literals
  - In most $l_1 \vee l_2 \vee \ldots \vee l_n$ cases, a clause is a disjunction of finite literals.
  - written as the symbol l_i.
- Conjunctive Normal Form: a sentence expressed as a conjunction of clauses.
- Satisfiability: a sentence is satisfiable if it is true in, or satisfied by, some models.

Propositional satisfiability (SAT) problem

- to determine the satisfiability of sentences in propositional logic.
- if there exists a model that satisfies a given logical sentence, then the sentence is satisfiable.
- Satisfiability can be checked by enumerating the possible models until one is found that satisfies the sentence, or by resolving complementary literals until an empty clause is derived.
- many problems in CS are really SAT problems.

Resolution rule

Unit resolution rule

$$\frac{(l_1 \vee l_2 \vee \ldots \vee l_{i-1} \vee l_i \vee l_{i+1} \vee \ldots \vee l_k,\ m)}{l_1 \vee \ldots \vee l_{i-1} \vee l_{i+1} \vee \ldots \vee l_k}$$

- where l is a literal and l_i and m are complementary literals.
- Unit resolution rule takes a clause which is a disjunction of literals, and a literal and produces a new clause as the resolvent.

Full resolution rule

$$\frac{(l_1 \vee l_2 \vee ... \vee l_{i-1} \vee l_i \vee l_{i+1} \vee ... \vee l_k,\ m_1 \vee m_2 \vee ... \vee m_{j-1} \vee m_j \vee m_{j+1} \vee ... \vee m_n)}{(l_1 \vee ... \vee l_{i-1} \vee l_{i+1} \vee ... \vee l_k \vee m_1 \vee ... \vee m_{j-1} \vee m_{j+1} \vee ... \vee m_n)}$$

- where l is a literal and l_i and m_j are complementary literals.
- Full resolution rule takes two clauses which are disjuctions of literals and produces a new clause containing all the literals of the two original clauses except for the two complementary literals.

Inference via proof by contradiction through resolution

$$\alpha \models \beta$$

- to prove that $\alpha \models \beta$, we can show that the sentence $\alpha \wedge \neg\beta$ is unsatisfiable.
- by deriving an empty clause $()$ from $\alpha \wedge \neg\beta$ using the resolution rule.
- In order to derive an empty clause, we use resolution which is a process to resolve complementary literals until to find an empty clause.
    i. R1: P (observation)
    ii. R2: P $\implies$ Q (knowledge, raining implies ground is wet)
    iii. $KB = P \wedge (P \implies Q)$
    iv. Query sentence: Q
    v. Inference problem: $KB \models Q$, i.e. from $KB = T$, get $Q = T$
    vi. Proof:
    - Let $(P \wedge (P \implies Q)) \wedge \neg Q$ valid
    - Convert $(P \wedge (P \implies Q)) \wedge \neg Q$ to **CNF**
    - Apply implication elimination rule, one has $(P \wedge (\neg P \vee Q)) \wedge \neg Q$
        - $C_1 : P$
        - $C_2 : \neg P \vee Q$
        - $C_3 : \neg Q$
    - Apply unit resolution rule to C_1 and C_2, resolve P and $\neg P$, one has $C_4 : Q$
    - Apply unit resolution rule to C_3 and C_4, resolve Q and $\neg Q$, one has $C_5 : ()$

Inference in FOL

- convert the first-order inference to propositional inference using the ruls for quantifiers.
    - universal instantiation (UI)
    - existential instantiation (EI)
- do inference in propositional logic using the methods about inference in propositional logic.
- this approach to first-order logic inference via propositionalization is complete, means any entailed sentences can be proved.

    - in most cases, this approach works
    - in some cases, it is slow and only useful when the domain is small.
- **get rid of quantifiers** by instantiating them with specific constants or variables.

Universal instantiation (UI)

$$\frac{\forall v \ \alpha}{\text{Subst}(\{v/g\}, \alpha)}$$

- to convert sentences with universal quantifiers to sentences without universal quantifiers.
- it can infer any sentence obtained by substituting a ground term for the universally quantified variable.
- $Subst(\{\theta, \alpha\})$ denotes the result of applying the subsitution θ to the sentence α and g is a ground term or a constant symbol.
- universal instantiation can be applied many times to produce may different consequences.
- $\forall x Loves(x, Mary)$
    - $Loves(John, Mary)$
    - $Loves(Sue, Mary)$
    - $Loves(Bill, Mary)$
    - ...

Existential instantiation (EI)

- to convert sentences with existential quantifiers to sentences without existential quantifiers.
- the quantified variable can be replaced by a single new constant symbol.
- for any sentence α, variable v, and constant symbol k not appear elsewhere in the knowldge base, the following rule is sound:

$$\frac{\exists v \ \alpha}{\text{Subst}(\{v/k\}, \alpha)}$$

- $\text{Subst}\{\theta, \alpha\}$ denotes the result of applying the subsitiution $\theta = v/k$ in the sentence α and k is ground term or a new constant symbol.
- Existential instantiation can be applied only once, and then the existentially quantified sentence can be discarded.
- $\exists x Loves(x, Mary)$
    - $Loves(K, Mary)$
    - **where K is a new constant symbol** not appear elsewhere in the knowledge base

CNF, Conjunctive Normal Form

Step	Rule Name	Description	Example
1	제거	$\implies$, $\iff$ 없애기	$P \implies Q \rightarrow \neg P \vee Q$
2	De Morgan's	$\neg$ 분배	$\neg(P \vee Q) \rightarrow (\neg P \wedge \neg Q)$
3	Double negation	이중부정 제거	$\neg(\neg P) \rightarrow P$
4	Distribution	$\vee$ over $\wedge$ 분배	$(P \vee (Q \wedge R)) \rightarrow (P \vee Q) \wedge (P \vee R)$
5	And-Elimination	$\wedge$ 분리	$(A \wedge B) \rightarrow A, B$ 따로