UNIT 4
Knowledge Representation and Reasoning

(7 Hours) — Knowledge Engineering (CACS458)

> ⚠ **Please Note**
> *These notes are generated to help with revision and **may contain errors or omissions**. Please verify important definitions, formulas and examples against your textbook, lecture notes and the official TU syllabus before relying on them in an exam.*

4.1 Proposition Logic, Predicate Logic and Reasoning

Knowledge representation (KR) is the study of how knowledge can be written down in a form a computer can store and reason with. A good representation should be *representationally adequate* (can express the needed knowledge), *inferentially adequate* (supports new inferences), *inferentially efficient* and *acquisitionally efficient* (easy to add new knowledge).

Propositional Logic

A **proposition** is a declarative statement that is either true or false. Propositions are combined using connectives:

Connective	Symbol	Read as
Negation	¬P	not P
Conjunction	P ∧ Q	P and Q
Disjunction	P ∨ Q	P or Q
Implication	P → Q	if P then Q
Biconditional	P ↔ Q	P if and only if Q

P	Q	¬P	P ∧ Q	P ∨ Q	P → Q	P ↔ 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

bcalink.com

- **Tautology**: always true. **Contradiction**: always false. **Satisfiable**: true under at least one assignment.
- Useful equivalences: $P \rightarrow Q \equiv \neg P \vee Q$; De Morgan: $\neg(P \wedge Q) \equiv \neg P \vee \neg Q$ and $\neg(P \vee Q) \equiv \neg P \wedge \neg Q$.
- **Modus Ponens**: from P and $P \rightarrow Q$ infer Q. **Modus Tollens**: from $\neg Q$ and $P \rightarrow Q$ infer $\neg P$. **Resolution**: from $(A \vee B)$ and $(\neg B \vee C)$ infer $(A \vee C)$.

Note: Propositional logic cannot express statements about all or some objects, nor relations between objects. Predicate logic removes this limitation.

Predicate (First-Order) Logic

First-order logic (FOL) extends propositional logic with *objects (constants)*, *variables*, *predicates* (properties/relations), *functions* and *quantifiers*: universal $\forall$ (for all) and existential $\exists$ (there exists).

English	FOL
Ram is a student.	Student(Ram)
All students are human.	$\forall x$ (Student(x) $\rightarrow$ Human(x))
Some students like AI.	$\exists x$ (Student(x) $\wedge$ Likes(x, AI))
Every student has a teacher.	$\forall x$ (Student(x) $\rightarrow \exists y$ Teacher(y, x))
No student is perfect.	$\forall x$ (Student(x) $\rightarrow \neg$Perfect(x))

Note: Typical mistake: use $\rightarrow$ with $\forall$ and $\wedge$ with $\exists$. "$\forall x$ Student(x) $\wedge$ Human(x)" wrongly says everything is a student.

bcalink.com

Reasoning in Logic

- **Universal instantiation**: from $\forall x$ P(x) infer P(a) for any constant a.
- **Unification**: finding a substitution that makes two predicates identical, e.g. Likes(x, AI) and Likes(Ram, y) unify with {x/Ram, y/AI}.
- **Forward chaining**: start from known facts and apply rules until the goal is reached (data-driven).
- **Backward chaining**: start from the goal and look for rules/facts that prove it (goal-driven; used by Prolog).
- **Resolution refutation**: negate the goal, convert everything to clauses and derive the empty clause.

Example 4.1 — Classic syllogism in FOL

Premises: (1) All humans are mortal. (2) Socrates is a human. Prove: Socrates is mortal.

Solution

(1) ∀x (Human(x) → Mortal(x)); (2) Human(Socrates).

Instantiate (1) with x = Socrates: Human(Socrates) → Mortal(Socrates).

Modus Ponens with (2): **Mortal(Socrates)**.

Example 4.2 — Resolution

Given: (a) P → Q, (b) Q → R, (c) P. Prove R.

Solution

Clauses: (¬P ∨ Q), (¬Q ∨ R), (P). Negated goal: (¬R).

Resolve (P) with (¬P ∨ Q) → (Q). Resolve (Q) with (¬Q ∨ R) → (R). Resolve (R) with (¬R) → empty clause ∎. Hence R is proved.

bcalink.com

4.2 Knowledge Representation Languages

Besides logic, several schemes are used to represent knowledge:

Scheme	Idea	Example
Production rules	IF condition THEN action; used in expert systems	IF temp > 38 THEN fever
Semantic networks	Graph of nodes (concepts) and labelled links (relations, is-a)	Canary —is-a→ Bird —has→ Wings
Frames	Structured records with slots, default values and inheritance	Frame Car: wheels = 4, engine = ?
Scripts	Stereotyped sequence of events	Restaurant script: enter, order, eat, pay
Conceptual graphs	Graph form of logic	[Person]→(Eats)→[Food]
Description logics / ontologies	Formal concepts, roles and individuals	Student ⊑ Person

Logic-Based and Rule Languages

- **Prolog**: programming in logic with Horn clauses (facts and rules), backward chaining and unification. Example: *mortal(X) :- human(X).*
- **CLIPS / Jess**: production-rule (forward-chaining) languages for expert systems.
- **KIF**: Knowledge Interchange Format — a logic-based language for exchanging knowledge between systems.
- **RDF, RDFS, OWL**: web-based knowledge representation languages (Unit 6).
- **Frame languages and description logics** underlie OWL.

Example 4.3 — Frame with inheritance

Frame **Bird**: can_fly = yes, covering = feathers. Frame **Penguin** (is-a Bird): can_fly = no, habitat = Antarctica. Frame **Tweety** (instance of Penguin). Does Tweety fly?

Solution

Tweety inherits from Penguin first; the local value *can_fly = no* overrides Bird's default. Answer: **No**; covering = feathers is inherited from Bird.

bcalink.com

4.3 Non-Monotonic Reasoning

In **monotonic reasoning** (classical logic), adding new knowledge never removes earlier conclusions. In **non-monotonic reasoning**, conclusions drawn from incomplete information may be *retracted* when new information arrives. This matches human commonsense reasoning.

Example 4.4 — The Tweety problem

Initially: Tweety is a bird; birds normally fly ⇒ conclude Tweety flies. Later we learn: Tweety is a penguin ⇒ conclusion is withdrawn.

Main Approaches

- **Default reasoning** (Reiter): uses default rules "if A holds and it is consistent to assume B, then conclude B". Example: Bird(x) : Flies(x) / Flies(x).
- **Closed-World Assumption (CWA)**: anything not known to be true is assumed false (used in databases and Prolog's negation as failure).
- **Circumscription** (McCarthy): minimises the extension of abnormal things — assume only the abnormal cases that are forced.
- **Truth Maintenance Systems (TMS)**: record justifications for beliefs so that dependent beliefs are withdrawn when an assumption is retracted.

- **Autoepistemic logic**: reasoning about one's own knowledge ("I would know if it were false").
- **Belief revision**: rules for changing a knowledge base consistently when new, conflicting information arrives.

Monotonic	Non-monotonic
Conclusions are never retracted	Conclusions may be retracted
Needs complete information	Works with incomplete information
Example: mathematics, FOL	Example: commonsense, default rules

4.4 Probabilistic Reasoning

Real-world knowledge is often **uncertain** (incomplete, noisy or vague). Probability theory provides a principled way to reason under uncertainty.

Basic Probability

- Probability of event A: $0 \le P(A) \le 1$; $P(\neg A) = 1 - P(A)$.
- **Conditional probability**: $P(A \mid B) = P(A \wedge B) / P(B)$.
- **Product rule**: $P(A \wedge B) = P(A \mid B) \cdot P(B)$. **Independence**: $P(A \wedge B) = P(A) \cdot P(B)$.
- **Bayes' theorem**: $P(H \mid E) = P(E \mid H) \cdot P(H) / P(E)$, where $P(H)$ is the prior and $P(H \mid E)$ the posterior.

bcalink.com

Example 4.5 — Bayes' theorem (medical test)

A disease affects 1% of people. A test detects it 90% of the time (sensitivity), but gives a false positive for 5% of healthy people. If a person tests positive, what is the probability they have the disease?

Solution

$P(D) = 0.01$; $P(+ \mid D) = 0.90$; $P(+ \mid \neg D) = 0.05$.

$P(+) = 0.90 \times 0.01 + 0.05 \times 0.99 = 0.009 + 0.0495 = 0.0585$.

$P(D \mid +) = 0.009 / 0.0585 \approx$ **0.154 (15.4%)** — low because the disease is rare.

Bayesian (Belief) Networks

A **Bayesian network** is a directed acyclic graph in which nodes are random variables and edges show direct influence. Each node stores a **conditional probability table (CPT)** given its parents. The joint distribution factorises as $\mathbf{P(X_1,\ldots,X_n) = \Pi\ P(X_i \mid Parents(X_i))}$.

Example 4.6 — Small network

Rain → WetGrass ← Sprinkler. P(Rain) = 0.2, P(Sprinkler) = 0.1. P(Wet | Rain, Sprinkler) = 0.99; P(Wet | Rain, ¬S) = 0.9; P(Wet | ¬R, S) = 0.8; P(Wet | ¬R, ¬S) = 0. Find P(Wet).

Solution

P(R,S) = 0.2×0.1 = 0.02 → 0.02×0.99 = 0.0198

P(R,¬S) = 0.2×0.9 = 0.18 → 0.18×0.9 = 0.162

P(¬R,S) = 0.8×0.1 = 0.08 → 0.08×0.8 = 0.064

P(¬R,¬S) = 0.72 → 0.72×0 = 0

P(Wet) = 0.0198 + 0.162 + 0.064 + 0 = **0.2458**

Other Uncertainty Methods

- **Naïve Bayes classifier**: assumes features are independent given the class.
- **Hidden Markov Models**: probabilistic reasoning over sequences (speech, POS tagging).
- **Dempster–Shafer theory**: belief functions; can represent ignorance.
- **Certainty factors** (used in MYCIN): CF ranges from −1 to +1.
- **Fuzzy logic**: handles vagueness through degrees of membership between 0 and 1 (e.g. "fairly tall" = 0.7).

bcalink.com

Key Takeaways — Unit 4

✓ Propositional logic handles whole statements; first-order logic adds objects, predicates and the quantifiers ∀ and ∃.

✓ Inference uses Modus Ponens, resolution, unification, and forward or backward chaining.

✓ Other KR schemes: rules, semantic networks, frames, scripts, description logics; languages include Prolog, CLIPS and OWL.

✓ Non-monotonic reasoning allows conclusions to be withdrawn (default reasoning, CWA, circumscription, TMS).

✓ Probabilistic reasoning uses Bayes' theorem and Bayesian networks to handle uncertainty.

— End of Unit 4 —