Elective Course
Course Code: CS4103
Autumn 2025-26

Lecture #28

# Artificial Intelligence for Data Science

**Week-8:**
Introduction to Knowledge Representation and Logic [Part-VI]
Revisiting Homework
Rule-based Systems

Course Instructor:
**Dr. Monidipa Das**
Assistant Professor
Department of Computational and Data Sciences
Indian Institute of Science Education and Research Kolkata, India 741246

## Propositional Logic: Homework

2

- Given the following knowledge base (KB)

$$\textbf{KB: } P \wedge Q \quad P \Rightarrow R \quad (Q \wedge R) \Rightarrow S$$

- Prove the following theorem
  a. **using inference rules (inference by proof approach)**
  b. **using resolution technique**

$$\textbf{Theorem: } S$$

Solved during today's lecture session

# FOL: Homework

3

- **[Homework]** Convert the following sentence to clause form
  $$(\forall x)(P(x) \;=>\; ((\forall y)(P(y) \;=>\; P(f(x,y))) \wedge \sim(\forall y)(Q(x,y) \;=>\; P(y))))$$
  Ans.: $\{\sim P(x) \vee \sim P(y) \vee P\big(f(x,y)\big),\ \sim P(x) \vee Q\big(x, g(x)\big),\ \sim P(x) \vee \sim P(g(x))\}$

- **[Homework]** Consider the following axioms:
  1. All hounds howl at night. ∀x(Hound(x) → Howl(x))
  2. Anyone who has any cat will not have any mice.
     ∀x∀y(Has(x, y) ∧ Cat(y) → ¬∃z(Has(x, z) ∧ Mouse(z)))
  3. Light sleepers do not have anything which howls at night. ∀x(LS(x) → ¬∃y(Has(x, y) ∧ Howl(y)))
  4. John has either a cat or a hound. ∃x(Has(John, x) ∧ (Cat(x) ∨ Hound(x)))
  5. (Conclusion) If John is a light sleeper, then John does not have any mice.
     LS(John) → ¬∃z(Has(John, z) ∧ Mouse(z))

  Now, prove the conclusion using Resolution



4

# Expert Systems

# Expert System

- Expert Systems are computer programs that exhibit intelligent behavior.
- The term “expert system” is reserved for programs whose knowledge base contains the knowledge used by human experts.
- Expert systems and knowledge-based systems are used synonymously.
- Building an expert system is known as *knowledge engineering* and its practitioners are called *knowledge engineers*.



# Building Blocks of Expert Systems

- ***Two*** principal parts:
  - the knowledge base; and
  - the reasoning or inference engine
- Knowledge Base
  - Factual and Heuristic knowledge
  - Knowledge Representation
    - Unit representation (also known as frame, schema, or list structure)
    - Production rule, or simply, rule: consists of an IF part and a THEN part (also called a *condition* and an *action*).
  - Expert systems whose knowledge is represented in rule form are called ***rule-based systems*.**

# Building Blocks of Expert Systems

- Inference engines or Inference procedures
  - problem-solving methods, built into program modules that manipulate and use knowledge in the knowledge base to form a line of reasoning
    - Forward Chaining
    - Backward Chaining etc.



# Early Expert Systems Success Stories

**DENDRAL** (Feigenbaum, Lederberg, Djerassi and Buchanan, 1965)
- deduce the likely molecular structure of organic chemical compounds from known chemical analyses and mass spectrometry data

**MYCIN** (Buchanan, Cohen, and Shortliffe, 1972-1980)
- diagnosis of infectious blood diseases and recommendation for use of antibiotics

**PROSPECTOR** (SRI International's Artificial Intelligence Center, 1970)
- analysis of geological data for minerals
- discovered a mineral deposit worth $100 million

**XCON/R1** (McDermott, 1978)
- configuration of DEC VAX computer systems
- 2500 rules; processed 80,000 orders by 1986; saved DEC $25M a year

# When not to use an Expert System

9

Expert systems are not suitable for all types of domains and tasks

They are not useful or preferable, when …

- efficient conventional algorithms are known
- the main challenge is computation, not knowledge
- knowledge cannot be captured efficiently or used effectively
- users are reluctant to apply an expert system, e.g. due to criticality of task, high risk or high security demands



# Rule-Based Expert Systems

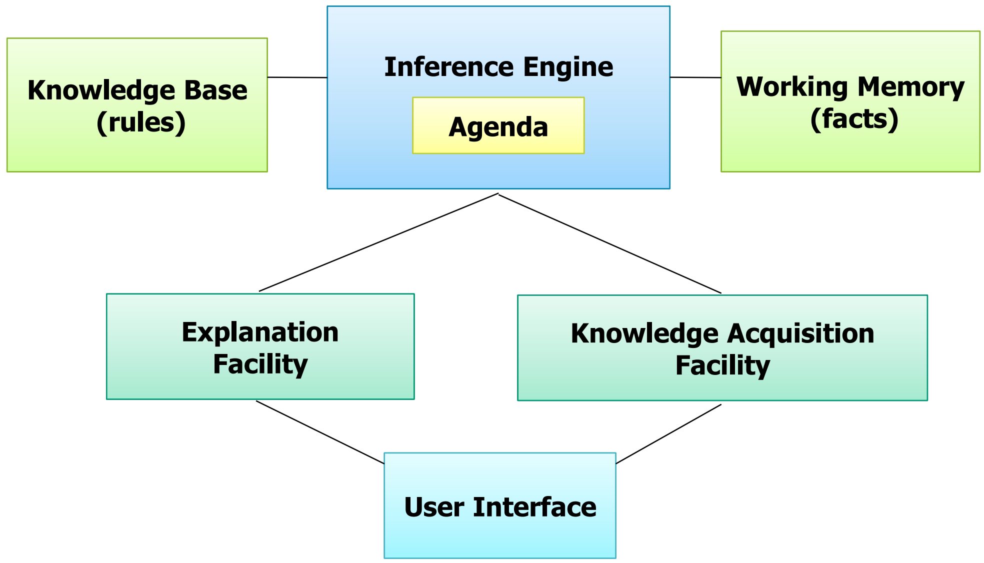
10

- **Features**
  - Knowledge base organized as a set of *if … then …* rules
  - Natural
  - Widely used
- inference engine determines which rule antecedents (condition-part) are satisfied
  - the left-hand condition-part must "match" facts in the working memory
- matching rules are "activated", i.e. placed on the agenda
- rules on the agenda can be executed ("fired")
  - an activated rule may generate new facts and/or cause actions through its right-hand side (action-part)
  - the activation of a rule may thus cause the activation of other rules through added facts based on the right-hand side of the fired rule

Dr. Monidipa Das, Department of CDS, IISER Kolkata

## Architecture of Rule-based Expert Systems








## Architecture of Rule-based Expert Systems [contd.]



- Knowledge-Base / Rule-Base
  - store expert knowledge as condition-action-rules (aka: if-then- or premise-consequence-rules)
- Working Memory
  - stores initial facts and generated facts derived by inference engine; maybe with additional parameters like the “degree of trust” into the truth of a fact ≅ certainty factor
- Inference Engine
  - matches condition-part of rules against facts stored in Working Memory (pattern matching);
  - rules with satisfied condition are active rules and are placed on the agenda;

## Architecture of Rule-based Expert Systems [contd.]

13

- Inference Engine (contd.)
  - among the active rules on the agenda, one is selected (see conflict resolution, priorities of rules) as next rule
  - execution ("firing") – consequence of rule is added as new fact(s) to Working Memory
- Explanation Facility
  - provides justification of solution to user (reasoning chain)
- Knowledge Acquisition Facility
  - helps to integrate new knowledge; also automated knowledge acquisition
- User Interface
  - allows user to interact with the XPS - insert facts, query the system, solution presentation



## Example Rules

14

IF … THEN Rules

**Rule: Red_Light**

**IF**      **the light is red (antecedent)**

**THEN**      **stop (consequent)**

**Rule: Green_Light**

**IF**      **the light is green**

**THEN**      **go**

Production Rules

**the light is red ==> stop**      **(left-hand side - antecedent)**

**(right-hand side - consequent)**

**the light is green ==> go**

# MYCIN Sample Rule

15

**MYCIN Format**

```
IF   (AND (SAME CNTEXT GRAM GRAMNEG)
     (SAME CNTEXT MORPH ROD)
     (SAME CNTEXT AIR ANAEROBIC)
THEN (CONCLUDE CNTEXT CLASS ENTEROBACTERIACEAE
     TALLY .8)
```

**Human-Readable Format**

IF the stain of the organism is gram negative
AND the morphology of the organism is rod
AND the aerobiocity of the organism is gram anaerobic
THEN there is strong evidence (0.8)
that the class of the organism is enterobacteriaceae



# Inference Engine Cycle

16

- describes the execution of rules by the inference engine
- "recognize-act cycle"
  - pattern matching
    - update the agenda (= conflict set)
      - add rules, whose antecedents are satisfied
      - remove rules with non-satisfied antecedents
  - conflict resolution
    - select the rule with the highest priority from the agenda
  - execution
    - perform the actions in the consequent part of the selected rule
    - remove the rule from the agenda

the cycle ends when no more rules are on the agenda, or when an explicit stop command is encountered

# Forward and Backward Chaining

17

- **Different methods of reasoning and rule activation**
  - **forward chaining (data-driven)**
    - reasoning from facts to the conclusion
    - as soon as facts are available, they are used to match antecedents of rules
    - a rule can be activated if all parts of the antecedent are satisfied
    - often used for real-time expert systems in monitoring and control
    - **examples:** CLIPS, OPS5
  - **backward chaining (query-driven)**
    - starting from a hypothesis (query), supporting rules and facts are sought until all parts of the antecedent of the hypothesis are satisfied
    - often used in diagnostic and consultation systems
    - **examples:** MYCIN



# Expert Systems-Advantages

18

- **economical**
  - lower cost per user
- **availability**
  - accessible anytime, almost anywhere
- **response time**
  - often faster than human experts
- **reliability**
  - can be greater than that of human experts
  - no distraction, fatigue, emotional involvement, …
- **explanation**
  - reasoning steps that lead to a particular conclusion

Dr. Monidipa Das, Department of CDS, IISER Kolkata

# Expert Systems – Problems



- **limited knowledge**
  - "shallow" knowledge
  - no "common-sense" knowledge
  - no knowledge from possibly relevant related domains
  - "closed world"
- **mechanical reasoning**
  - may not have or select the most appropriate method for a particular problem
  - some "easy" problems are computationally very expensive
- **lack of trust**
  - users may not want to leave critical decisions to machines





# Questions?

*Slide Content taken from Prof. Cesare Tinelli, Prof. Stuart Russell, and Prof. Jim Martin*