Knowledge Engineering & Expert Systems

1

Agenda

- What is knowledge?
- Knowledge Engineering
- Knowledge Representation Techniques
- Expert Systems
- Expert System Structure
- Expert System Characteristics
- Inference Chaining
- Forward chaining and backward chaining
- Conflict resolution
- Uncertainty Management in Expert System

2

What is Knowledge?

- **Knowledge** is a **theoretical or practical understanding of a subject or a domain.** Knowledge is also the sum of what is currently known, and apparently knowledge is power. Those who possess knowledge are called experts.
  - Static knowledge
  - Dynamic Knowledge
- **Domain Expert** is **anyone has deep knowledge** (of both facts and rules) and strong practical experience in a particular domain. The area of the domain may be limited. In general, an **expert is a skillful person who can do things other people cannot.**

3

Knowledge Engineering

- The disciplines of Knowledge Engineering(KE) and Knowledge Management (KM) are closely tied.
- **Knowledge Engineering** deals with the **development of intelligent information systems in which knowledge and reasoning play pivotal role**
- **Knowledge Management** is a developed filed at the intersection of computer science and management science **deals with knowledge as a key source** in modern organizations

4

Knowledge Representation Techniques

- Logical Representation
- Semantic Network Representation
- Frame Representation
- Production Rules

5

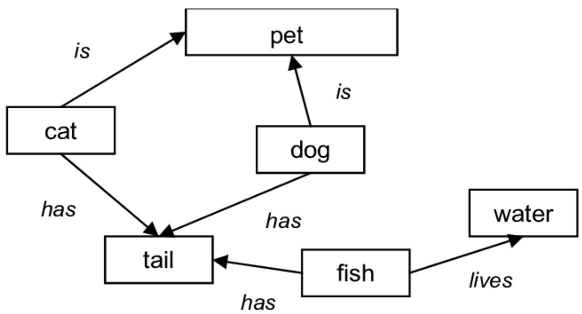
Logical Representation

- Language with concrete communication rules that deals with propositions
- No ambiguity in representation
- Example: All birds fly
  - $\forall x \; bird(x) \rightarrow fly(x)$

6

Semantic Network Representation

- Semantic network is graphical representation that represent semantic relations between concepts.
  - Vertices: represent concepts
  - Edges: represent semantic relation/mapping/connection between concepts



https://www.researchgate.net/figure/Example-of-a-semantic-network-describing-animals-Source-own-work-A-new-vision-of-the_fig23_283328263

7

Frame Representation

- Knowledge is represented as set of related frames.
- Frame representation is based on the inheritance concept.
- Frame is set of attributes that defines the object condition.
- Every attribute has Name and value.

8

Frame Representation(cont.)

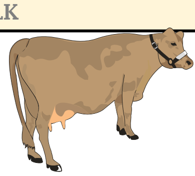
Frame Representation of the " animal kingdom"

MAMMAL

A-KIND-OF ANIMAL

SKIN COVER HAIR

ACTIVITY: PRODUCES MILK



CARNIVORE

A-KIND-OF ANIMAL

APPEARANCE : FORWARD EYES

POINTED TEETH

APPENDGES : CLAWS

ACTIVITY : EATS Meat

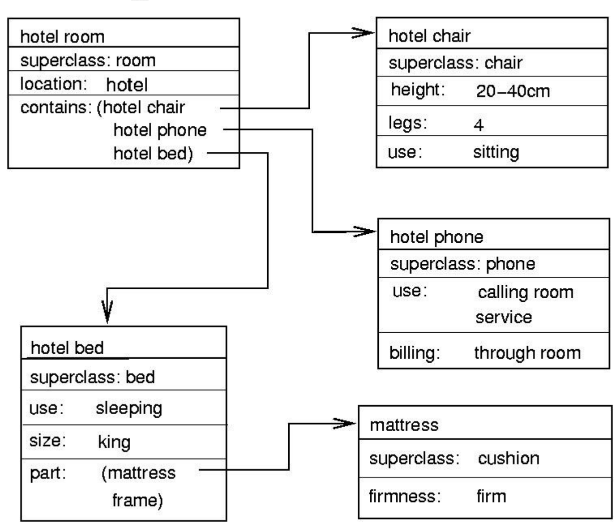
BIRD

A-KIND-OF ANIMAL

SKIN COVER : FEATHER

ACTIVITY : FLY LAYS EGGS

Frame Representation(cont.)



https://transblog.grieve-smith.com/wp-content/uploads/2017/08/hotel.jpg

Production Rules

- The human mental process is internal, and it is too complex to be represented as an algorithm. However, **most experts are capable of expressing their knowledge in the form of Production rules for problem solving.**
- Production rules (rules) is defined as **IF-THEN structure** that relates given information or facts in the IF part to some action in the THEN part.
  - IF part called the antecedent (premise or condition)
  - Then part called the consequent (conclusion or action).
- A rule provides some description of how to solve a problem.

IF <antecedent>
THEN <consequent>

11

Production Rules(cont.)

- If the ‘traffic light’ is green

  THEN the action is go
- If the ‘traffic light’ is red

  THEN the action is stop
- A rule can have multiple antecedents joined by the keywords **AND** (conjunction), **OR** (disjunction) or a combination of both.
  - IF ‘age of the customer’ < 18
    AND ‘cash withdrawal’ > 1000
    THEN ‘signature of the parent’ is required

12

Production Rules(cont.)

- The antecedent of a rule incorporates two parts: an **object** (linguistic object) and its **value**. The object and its value are linked by an **operator**.
- The operator identifies the object and assigns the value.
- Operators such as is, are, is not, are not are used to assign a **symbolic value** to a linguistic object.
- Mathematical operators can be used to assign **numerical value to a** numerical object.

IF customer is teenager
AND 'cash withdrawal' > 1000
THEN 'signature of the parent' is required

13

Production Rules(cont.)

- Rules can represent relations, recommendations, directives, strategies and heuristics
- **Relation**
  - IF the 'fuel tank' is empty
    THEN the car is dead
- **Recommendation**
  - IF the season is autumn
    AND the sky is cloudy
    AND the forecast is drizzle
    THEN the advice is 'take an umbrella'
- **Directive**
  - IF the car is dead
    AND the 'fuel tank' is empty
    THEN the action is 'refuel the car'

14

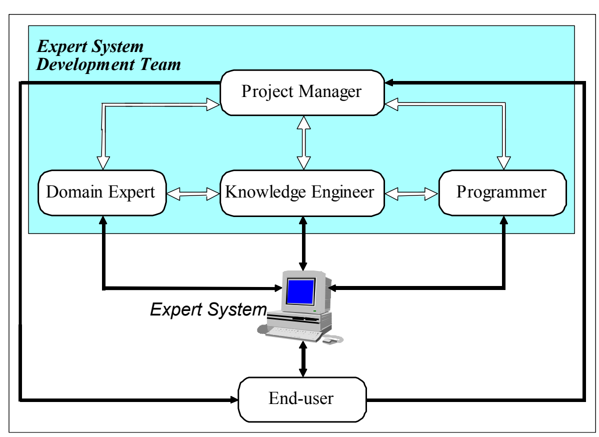
Production Rules(cont.)

- **Strategy**
  - IF the car is dead
    THEN the action is 'check the fuel tank'
    step1 is complete
  - IF step1 is complete
    AND the 'fuel tank' is full
    THEN the action is 'check the battery'
    step2 is complete
- **Heuristic**
  - IF the spill is liquid
    AND the 'spill pH' < 6
    AND the 'spill smell' is vinegar
    THEN the 'spill material' is 'acetic acid'

15

Expert System

- Expert system is computer program that emulates the decision making capability of a human expert.
- The expert system development team contains five members: the domain expert, the knowledge engineer, the programmer, the project manager and the end-user.
- The success of the expert system entirely depends on how well the members work together.



16

Expert System(cont.)

- The **domain expert** is a **knowledgeable and skilled person capable of solving problems in a specific area or domain.** This person has the greatest expertise in a given domain. This expertise is to be captured in the expert system. The domain expert is the most important member in the expert system development team.
- The **knowledge engineer** is **someone who is capable of designing, building and testing an expert system**.
- Knowledge engineer tasks
  - Interview the domain expert to find out how a particular problem is solved.
  - Establishing what reasoning methods the expert uses to handle facts and rules and decides how to represent them in the expert system.
  - Choosing the development software or an expert system shell, or looks at programming languages for encoding the knowledge.
  - Testing, revising and integrating the expert system into the workplace.

17

Expert System (cont.)

- The **programmer** is responsible for the actual programming, describing the domain knowledge in terms that a computer can understand. The programmer needs to have skills in
  - conventional programming languages such as C
  - symbolic programming in such AI languages as LISP, Prolog and OPS5
  - the application of different types of expert system shells such as CLIPS and JESS
- The **project manager** is the leader of the development team, responsible for keeping the project on track. He makes sure that all deliverables and milestones are met.
- The **end-user**, is a person who uses the expert system when it is developed. The design of the user interface of the expert system is vital for the project's success.

18

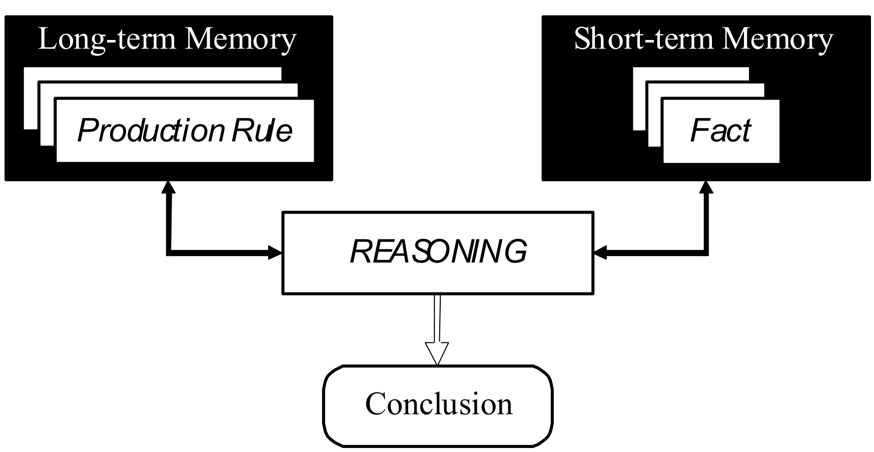
Expert System Structure

- In the early seventies, Newell and Simon from Carnegie-Mellon University proposed a production system model, the foundation of the modern rule-based expert systems.
- The **production model** is based on the idea that humans solve problems by applying their knowledge (expressed as production rules) to a given problem represented by problem-specific information.
- The production rules are stored in the long-term memory and the problem-specific information or facts in the short-term memory.

19

Expert System Structure (cont.)

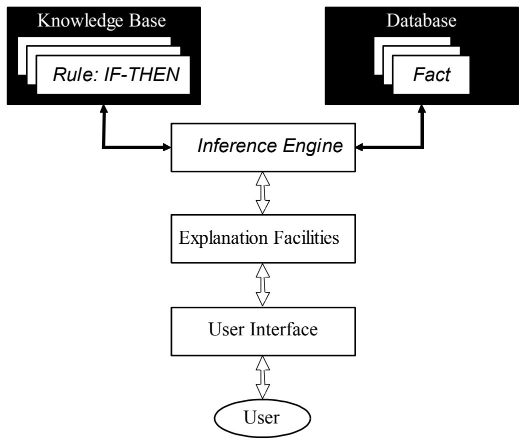
- Production system model



20

Expert System Structure (cont.)

- Basic structure of a rule-based expert system



Expert System Structure (cont.)

- The **knowledge base** contains the domain knowledge useful for problem solving. In a rule- based expert system, the knowledge is represented as a **set of rules.**
  - Each rule specifies a relation, recommendation, directive, strategy or heuristic and has the IF (condition) THEN (action) structure.
  - When the condition part of a rule is satisfied, the rule is said to **fire** and the action part is executed.
- The **database** includes a **set of facts** used to match against the IF (condition) parts of rules stored in the knowledge base.
- The **inference engine** carries out the **reasoning** whereby the expert system reaches a solution. It links the rules given in the knowledge base with the facts provided in the database.

Expert System Structure (cont.)

- The **explanation facilities** enable the user to ask the expert system **how** a particular conclusion is reached and **why** a specific fact is needed. An expert system must be able to explain its reasoning and justify its advice, analysis or conclusion.
- The **user interface** is the means of communication between a user seeking a solution to the problem and an expert system.

23

Expert System Characteristics

- An expert system is built to perform at a human expert level in a **narrow, specialized domain.** Thus, the most important characteristic of an expert system is its **high-quality performance.** No matter how fast the system can solve a problem, the user will not be satisfied if the result is wrong.
  - The **speed of reaching a solution is very important**. Even the most accurate decision or diagnosis may not be useful if it is too late to apply, for instance, in an emergency, when a patient dies or a nuclear power plant explodes.
- Expert systems apply **heuristics** to guide the reasoning and thus reduce the search area for a solution.
- **Explanation capability**: it enables the expert system to review its own reasoning and explain its decisions.
- Expert systems employ **symbolic reasoning** when solving a problem. Symbols are used to represent different types of knowledge such as facts, concepts and rules.

24

Expert System Characteristics (cont.)

- In expert systems, **knowledge is separated from its processing** (the knowledge base and the inference engine are split up).
  - A conventional program is a mixture of knowledge and the control structure to process this knowledge. This mixing leads to difficulties in understanding and reviewing the program code, as any change to the code affects both the knowledge and its processing.
  - When an expert system shell is used, a knowledge engineer or an expert simply enters rules in the knowledge base. **Each new rule adds some new knowledge and makes the expert system smarter**.

Expert System vs Conventional Systems

Human Experts	*Expert Systems*	*Conventional Programs*
Use knowledge in the form of rules of thumb or heuristics to solve problems in a narrow domain.	Process knowledge expressed in the form of rules and use symbolic reasoning to solve problems in a *narrow domain*.	Process data and use algorithms, a series of well-defined operations, to solve general numerical problems.
In a human brain, knowledge exists in a compiled form.	Provide a *clear separation of knowledge from its processing*.	Do not separate knowledge from the control structure to process this knowledge.
Capable of explaining a line of reasoning and providing the details.	*Trace the rules fired* during a problem-solving session and *explain how* a particular conclusion was reached and *why* specific data was needed.	Do not explain how a particular result was obtained and why input data was needed.

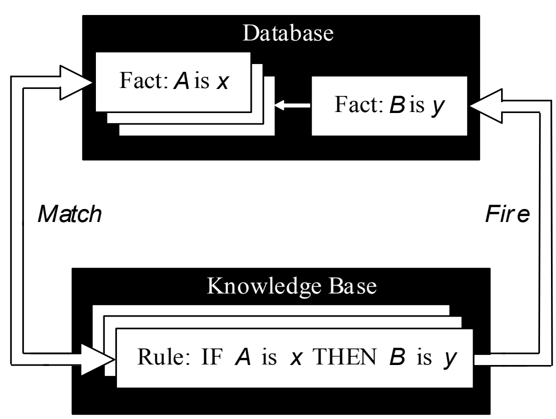
Expert System vs Conventional Systems (cont.)

Human Experts	*Expert Systems*	*Conventional Programs*
Use inexact reasoning and can deal with incomplete, uncertain and fuzzy information.	Permit *inexact reasoning* and can deal with incomplete,uncertain and fuzzy data.	Work only on problems where data is complete and exact.
Can make mistakes when information is incomplete or fuzzy.	*Can make mistakes* when data is incomplete or fuzzy.	Provide no solution at all, or a wrong one, when data is incomplete or fuzzy.
Enhance the quality of problem solving via years of learning and practical training. This process is slow, inefficient and expensive.	Enhance the quality of problem solving by adding new rules or adjusting old ones in the knowledge base. When new knowledge is acquired, *changes are easy* to accomplish.	Enhance the quality of problem solving by changing the program code, which affects both the knowledge and its processing, making changes difficult.

27

Inference Chaining

- The inference engine compares each rule stored in the knowledge base with facts contained in the database. When the IF (condition) part of the rule matches a fact, the rule is **fired** and its THEN (action) part is placed in the **agenda** for possible executed.
- The matching of the rule IF parts to the facts produces **inference chains.** The inference chain indicates **how an expert system applies the rules to reach a conclusion.**



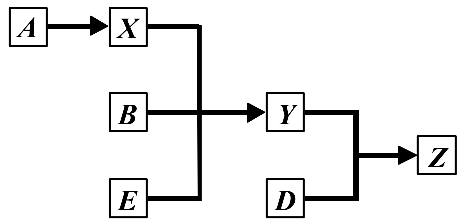
28

Inference Chaining (cont.)

Rule 1: IF *Y* is true
AND *D* is true
THEN *Z* is true

Rule 2: IF *X* is true
AND *B* is true
AND *E* is true
THEN *Y* is true

Rule 3: IF *A* is true
THEN *X* is true



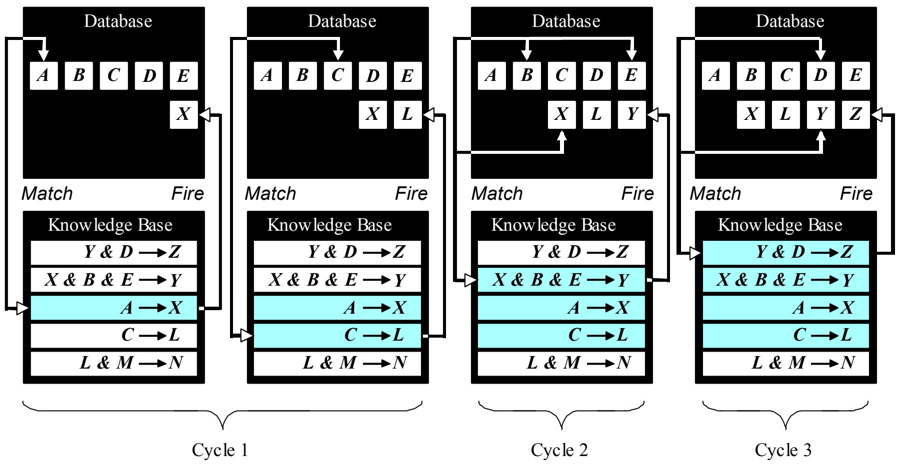
29

Forward Chaining

- Forward chaining is the **data-driven reasoning**.
- The reasoning starts from the known data and proceeds forward with that data.
- Each time only the topmost rule is executed.
- When fired, the rule adds a new fact in the database.
- Any rule can be executed only once.
- The match-fire cycle stops when no further rules can be fired.

30

Forward Chaining (cont.)



31

Forward Chaining (cont.)

- Forward chaining is a technique for **gathering information** and then inferring from it whatever can be inferred.
- However, in forward chaining, many rules may be executed that have nothing to do with the established goal.
- Therefore, if our **goal is to infer only one particular fact**, the forward chaining inference technique would be **inefficient**.

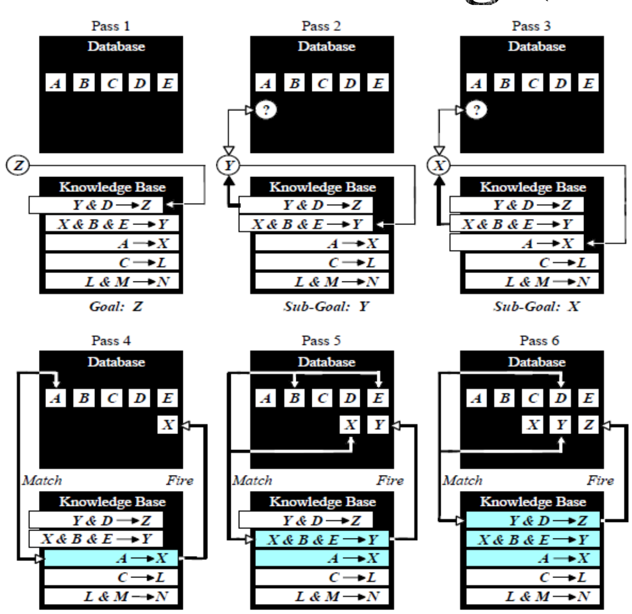
32

Backward Chaining

- **Backward chaining** is the **goal-driven reasoning.**
- In backward chaining, an expert system has the goal (a hypothetical solution) and the inference engine attempts to find the evidence to prove it.
- First, the knowledge base is searched to find rules that might have the desired solution (goal in their THEN (action) parts).
- If such a rule is found and its IF (condition) part matches data in the database, then the rule is fired and the goal is proved. However, this is rarely the case.
- Thus the inference engine puts aside the rule it is working with (the rule is said to stack) and sets up a new goal, a subgoal, to prove the IF part of this rule.
- Then the knowledge base is searched again for rules that can prove the subgoal.
- The inference engine repeats the process of stacking the rules until no rules are found in knowledge base to prove the current subgoal.

33

Backward Chaining (cont.)



34

Forward or Backward Chaining

- How do we choose between forward and backward chaining?
  - If an expert first needs to gather some information and then tries to infer from it whatever can be inferred, choose the forward chaining inference engine.
  - However, if your expert begins with a hypothetical solution and then attempts to find facts to prove it, choose the backward chaining inference engine.

35

Conflict Resolution

- Rule 1: IF the 'traffic light' is green
  THEN the action is go
- Rule 2: IF the 'traffic light' is red
  THEN the action is stop
- Rule 3: IF the 'traffic light' is red
  THEN the action is go

- Both of Rule 2 and Rule 3 can be set to fire when the condition part is satisfied.
- These rules represent a conflict set.
- The inference engine must determine which rule to fire from such a set.
- A method for choosing a rule to fire when more than one rule can be fired in a given cycle is called **conflict resolution.**

36

Conflict Resolution Methods

- Fire the rule with the **highest priority**. In simple applications, the priority can be established by placing the rules in an appropriate order in the knowledge base.

RULE 1
(Priority 100)
……………

RULE 2
(Priority 90)
………………

37

Conflict Resolution Methods(cont.)

- Fire the **most specific rule**. This method is also known as the **longest matching strategy**. It is based on the assumption that a specific rule processes more information than a general one.

Rule 1:
IF the season is winter
AND the sky is cloudy
AND the forecast is rain
THEN the advice is 'stay home'

Rule 2:
IF the season is winter
THEN the advice is 'take an umbrella'

38

Conflict Resolution Methods(cont.)

- Fire the rule that uses the **data most recently entered** in the database. This method relies on **time tags attached to each fact** in the database. In the conflict set, the expert system first fires the rule whose antecedent uses the data most recently added to the database.
  - Useful in real-time expert system, where the data is continuously updated

Rule 1:
IF the forecast is rain [08:16 AM 11-Feb-2019]
THEN the advice is 'take an umbrella'
Rule 2:
IF the weather is wet [09:20 AM 11-Feb-2019]
THEN the advice is 'stay home'

Advantages of Expert Systems

- **Natural knowledge representation**
  - An expert usually explains the problem-solving procedure with such expressions as this: "In such-and-such situation, I do so-and-so".
  - These expressions can be represented quite naturally as IF-THEN production rules.
- **Uniform structure**
  - Production rules have the uniform IF-THEN structure.
  - Each rule is an independent piece of knowledge.
  - The syntax of production rules enables them to be self-documented.
- **Separation of knowledge from its processing**
  - The structure of a rule-based expert system provides an effective separation of the knowledge base from the inference engine.
  - This makes it possible to develop different applications using the same expert system shell.

Advantages of Expert Systems (cont.)

- **Dealing with incomplete and uncertain knowledge**
  - Most rule-based expert systems are capable of representing and reasoning with incomplete and uncertain knowledge.
  - Rules represent the uncertainty by numbers called the certainty factors
  - the certainty factors is used to establish the degree of confidence that the rule conclusion is true.

IF season is winter
AND sky is 'cloudy'
AND wind is low
THEN forecast is clear { cf 0.1 };
forecast is drizzle { cf 1.0 };
forecast is rain { cf 0.9 }

41

Disadvantages of Expert Systems

- **Inability to learn**
  - In general, rule-based expert systems do not have an ability to learn from the experience. Unlike a human expert, who knows when to "break the rules", an expert system cannot automatically modify its knowledge base, or adjust existing rules or add new ones. The knowledge engineer is still responsible for revising and maintaining the system.
- **Opaque relations between rules**
  - Although the individual production rules are relatively simple and self-documented, their logical interactions within the large set of rules may be opaque. Rule-based systems make it difficult to observe how individual rules serve the overall strategy.
- **Ineffective search strategy**
  - The inference engine applies an exhaustive search through all the production rules during each cycle. Expert systems with a large set of rules (over 100 rules) can be slow, and thus large rule-based systems can be unsuitable for real-time applications.

42

Uncertainty Management in Expert System

- Information can be incomplete, inconsistent, uncertain, or all three
- Information is often unsuitable for solving a problem.
- **Uncertainty** is defined as the **lack of the exact knowledge** that would enable us to reach a perfectly reliable conclusion.
- **Classical logic** permits only exact reasoning. It assumes that perfect knowledge always exists.

43

Sources of Uncertain Knowledge

- **Weak implications**
  - Domain experts and knowledge engineers have the painful task of establishing concrete correlations between IF (condition) and THEN (action) parts of the rules. Therefore, expert systems need to have the ability to handle vague associations, for example by accepting the degree of correlations as numerical certainty factors.
- **Imprecise language**
  - Our natural language is ambiguous and imprecise. We describe facts with such terms as often and sometimes, frequently and hardly ever. As a result, it can be difficult to express knowledge in the precise IF-THEN form of production rules.

44

Sources of Uncertain Knowledge (cont.)

- **Unknown data**
  - When the data is incomplete or missing, the only solution is to accept the value "unknown" and proceed to an approximate reasoning with this value.
- **Combining the views of different experts**
  - Experts often have contradictory opinions and produce conflicting rules. To resolve the conflict, the knowledge engineer has to attach a weight to each expert and then calculate the composite conclusion. But no systematic method exists to obtain these weights.

45

Bayesian Reasoning

- Suppose all rules in the knowledge base are represented in the following form:
  - IF *E* is true
    THEN *H* is true {with probability *p*}
- This rule implies that if event E occurs, then the probability that event H will occur is p.
- In expert systems, **H** usually represents a **hypothesis** and **E** denotes **evidence** to support this hypothesis.

46

Bayesian Reasoning (cont.)

- The **Bayesian rule** expressed in terms of hypotheses and evidence looks like this:

$$p(H|E) = \frac{p(E|H) \times p(H)}{p(E|H) \times p(H) + p(E|\neg H) \times p(\neg H)}$$

- p(H) is the **prior probability** of hypothesis H being true
- p(E|H) is the **conditional probability** that hypothesis H being true will result in evidence E
- p(¬H) is the **prior probability** of hypothesis H being false
- p(E|¬H) is the **conditional probability** of finding evidence E even when hypothesis H is false.
- p(H|E) called **posterior probability** of hypothesis H upon observing evidence E.

47

Bayesian Reasoning (cont.)

- In case of multiple hypotheses H_1, H_2,..., H_m and multiple evidences E_1, E_2 ,..., E_n. The hypotheses and the evidences must be mutually exclusive and exhaustive.
- Single evidence E and multiple hypotheses follow:

$$p(H_i|E) = \frac{p(E|H_i) \times p(H_i)}{\sum_{k=1}^{m} p(E|H_k) \times p(H_k)}$$

48

Bayesian Reasoning (cont.)

- Multiple evidences and multiple hypotheses follow:
  - requires to obtain the conditional probabilities of all possible combinations of evidences for all hypotheses, and thus places an enormous burden on the expert.

$$p(H_i|E_1\,E_2\ldots E_n)=\frac{p(E_1\,E_2\ldots E_n|H_i)\times p(H_i)}{\sum_{k=1}^{m} p(E_1\,E_2\ldots E_n|H_k)\times p(H_k)}$$

$$p(H_i|E_1\,E_2\ldots E_n)=\frac{p(E_1|H_i)\times p(E_2|H_i)\times\ldots\times p(E_n|H_i)\times p(H_i)}{\sum_{k=1}^{m} p(E_1|H_k)\times p(E_2|H_k)\times\ldots\times p(E_n|H_k)\times p(H_k)}$$

49

Bayesian Reasoning (cont.)

- The Bayesian reasoning requires **probability values as primary inputs**.
  - The assessment of these values usually involves human judgement.
  - Humans cannot elicit probability values consistent with the Bayesian rules.
- The conditional probabilities may be **inconsistent** with the prior probabilities given by the expert.
  - The reason for the inconsistency is that the expert made **different assumptions** when assessing the conditional and prior probabilities.
- **Certainty factors** theory is a popular alternative to Bayesian reasoning

50

Certainty Factors Theory and Evidential Reasoning

- **Certainty Factor (cf)**
  - Number to measure the expert's belief.
  - The maximum value of the certainty factor is, say, +1.0 (definitely true) and the minimum -1.0 (definitely false).
  - For example, if the expert states that some evidence is almost certainly true, a cf value of 0.8 would be assigned to this evidence.

Term	*Certainty Factor*
Definitely not	−1.0
Almost certainly not	−0.8
Probably not	−0.6
Maybe not	−0.4
Unknown	−0.2 to +0.2
Maybe	+0.4
Probably	+0.6
Almost certainly	+0.8
Definitely	+1.0

Certainty Factors Theory (cont.)

- In expert systems with certainty factors, the knowledge base consists of a set of rules that have the following syntax:
  - IF <evidence>
    THEN <hypothesis> {cf }
  - cf represents belief in hypothesis H given that evidence E has occurred.
- The certainty factors theory is based on two functions:
  - measure of belief MB(H,E) : the measure of increased belief in H due to E
  - measure of disbelief MD(H,E) : the measure of increased disbelief in H due to E.

Certainty Factors Theory (cont.)

- The certainty factors theory is based on two functions:
  - measure of belief MB(H,E) : the measure of increased belief in H due to E
  - measure of disbelief MD(H,E) : the measure of increased disbelief in H due to E.

$$MB(H,E) = \begin{cases} 1 & \text{if } p(H) = 1 \\ \dfrac{max\left[p(H|E), p(H)\right] - p(H)}{max\left[1,0\right] - p(H)} & \text{otherwise} \end{cases}$$

$$MD(H,E) = \begin{cases} 1 & \text{if } p(H) = 0 \\ \dfrac{min\left[p(H|E), p(H)\right] - p(H)}{min\left[1,0\right] - p(H)} & \text{otherwise} \end{cases}$$

53

Certainty Factors Theory (cont.)

- The values of MB(H, E) and MD(H, E) range between 0 and 1.
- The strength of belief or disbelief in hypothesis H depends on the kind of evidence E observed.
- Some facts may increase the strength of belief, but some increase the strength of disbelief.
- The **total strength of belief or disbelief** in a hypothesis:

$$cf = \frac{MB(H,E) - MD(H,E)}{1 - min\left[MB(H,E), MD(H,E)\right]}$$

54

Certainty Factors Theory (cont.)

- Simple Rule
  - IF A is X
    THEN B is Y
- An expert may not be absolutely certain that this rule holds. In some cases, even when the IF part of the rule is satisfied and object A takes on value X, object B can acquire some different value Z.
  - IF A is X
    THEN B is Y {cf 0.7};
    B is Z {cf 0.2}

55

Certainty Factors Theory (cont.)

- The certainty factor assigned by a rule is **propagated** through the reasoning chain.
- This involves establishing the net certainty of the rule consequent when the evidence in the rule antecedent is uncertain:

$$\mathbf{cf\ (H,e) = cf\ (E,e) \times cf\ (H,E)}$$

- For example,
  - IF sky is clear
    THEN the forecast is sunny {cf 0.8}
  - The current certainty factor of sky is clear is 0.5, then

$$\mathbf{cf(H,E) = 0.5 \times 0.8 = 0.4}$$

  - This result can be interpreted as “It may be sunny”.

56

Certainty Factors Theory (cont.)

- For **conjunctive** rules

IF <evidence E_1>
⋮
AND <evidence E_n>
THEN <hypothesis H> {cf}

- The certainty of hypothesis H, is established as follows:

cf (H,E1∩ E2∩…∩En) = min [cf (E1,e), cf (E2,e),…,cf (En,e)] x cf

- For example,
  - IF sky is clear
    AND the forecast is sunny
    THEN the action is 'wear sunglasses' {cf 0.8}
    - the certainty of sky is clear is 0.9 and the certainty of the forecast of sunny is 0.7, then

cf (H,E1∩E2) = min [0.9, 0.7] × 0.8 = 0.7 × 0.8 = 0.56

57

Certainty Factors Theory (cont.)

- For **disjunctive** rules

IF <evidence E_1>
⋮
OR <evidence E_n>
THEN <hypothesis H> {cf}

- The certainty of hypothesis H, is established as follows:

cf (H,E1∪ E2∪…∪En) = max[cf (E1,e), cf (E2,e),…,cf (En,e)] x cf

- For example,
  - IF sky is overcast
    OR the forecast is rain
    THEN the action is 'take an umbrella' {cf 0.9}
  - the certainty of sky is overcast is 0.6 and the certainty of the forecast of rain is 0.8, then

cf (H,E1∪E2) = max [0.6, 0.8] × 0.9 = 0.8 × 0.9 = 0.72

58

Certainty Factors Theory (cont.)

- Combined certainty factor
  - When the same consequent is obtained as a result of the execution of two or more rules, the individual certainty factors of these rules must be merged to give a combined certainty factor for a hypothesis
- Rule 1: IF A is X
  THEN C is Z {cf 0.8}
- Rule 2: IF B is Y
  THEN C is Z {cf 0.6}
- What certainty should be assigned to object C having value Z if both Rule 1 and Rule 2 are fired?
  - Common sense suggests that, if we have two pieces of evidence (A is X and B is Y) from different sources (Rule 1 and Rule 2) supporting the same hypothesis (C is Z), then the **confidence in this hypothesis should increase and become stronger** than if only one piece of evidence had been obtained.

59

Certainty Factors Theory (cont.)

- Combined certainty factor

$$cf(cf_1, cf_2) = \begin{cases} cf_1 + cf_2 \times (1 - cf_1) & \text{if } cf_1 > 0 \text{ and } cf_2 > 0 \\ \dfrac{cf_1 + cf_2}{1 - min\left[|cf_1|, |cf_2|\right]} & \text{if } cf_1 < 0 \text{ or } cf_2 < 0 \\ cf_1 + cf_2 \times (1 + cf_1) & \text{if } cf_1 < 0 \text{ and } cf_2 < 0 \end{cases}$$

- cf_1 is the confidence in hypothesis H established by Rule 1
- cf_2 is the confidence in hypothesis H established by Rule 2
- $|cf_1|$ and $|cf_2|$ are absolute magnitudes of cf_1 and cf_2 respectively

60

Thank You

61