• Expert systems (ES) provide expert quality advice, diagnosis or recommendations.

• ES solve real problems which normally would require a human expert.

• An example situation – using an ES in diagnosing rare diseases.

• Two basic steps in building an expert system:
  1. Extracting the relevant knowledge from the human expert by a knowledge engineer.
  2. S/he then develops the knowledge base of the ES.

• Knowledge acquisition generally involve interviewing the expert.

• An ES should be
  1. easily inspected and modified
  2. able to explain its reasoning

• General architecture of expert system:

  - User Interface
  - Problem-Specific Database
  - Inference Engine
  - Knowledge Base
  - Explanation System

• The Problem-Specific Database
  1. All info about the current problem.
  2. All conclusion that the system has been able to derive.

• The Knowledge Base
  1. Contains all of the relevant, domain-specific, problem-solving knowledge.
  2. Two perspectives: Nature and Format

• The User Interface
  1. Enable the system to pose questions to the users
  2. Provide explanations about why a particular question is asked
  3. Allow user queries
  4. Displaying the derived results

• Inference Engine
  1. Interpreter of the knowledge stored in the knowledge base.
  2. Find connections between the problem features and solutions.

• The Explanation Facilities
  1. Justify 'why' a question was asked and 'how' it reached some conclusion.
Logical Inferences

- The process of reasoning involves making inferences from known facts.
- Given a set of premises known (or thought to be true) and a reasoning method, certain conclusions can be inferred to also be true.
- Making inferences involves the derivation of new facts from a set of true facts.
- Predicate logic provides a set of sound rules of inference with which we can perform logical inferences.

Logical Inferences – Modus Ponens

- The best known of these is modus ponens
- The basis for rule-based reasoning
- If statements \( p \) and \( (p \rightarrow q) \) are known to be true, then we can infer that \( q \) is true.

\[
∀X (\text{has\_flu}(X) → \text{high\_temperature}(X))
\]

Example:
- If someone has flu then he has high temperature
  \[ VX (\text{has\_flu}(X) → \text{high\_temperature}(X)) \]

Logical Inferences – Modus Tolens

- If the relationship \( ∀X (\text{has\_flu}(X) → \text{high\_temperature}(X)) \) is true and if \( \text{peter} \) has no high temperature, then he doesn’t have flu. That is if
  \[ ¬\text{high\_temperature}(\text{peter}) \]
  which implies, through modus tolens,
  \[ ¬\text{has\_flu}(\text{peter}) \]

Deduction - 1

- Logically correct inference, i.e. deduction from true premises is guaranteed to result in true conclusions.
- The most accepted, understood method.
- The basis of both propositional and predicate logics.

Deduction - 2

- For example,
  - IF Object A is larger than Object B
    AND Object B is larger than Object C
  THEN Object A is larger than Object C
  \[ ∀A∀B∀C (\text{larger}(A,B) ∧ \text{larger}(B,C) → \text{larger}(A,C)) \]

- In predicate logic, this is represented as
**Deduction - 3**

- If the following axioms exist:
  
  \[
  \text{larger(house, car)} \\
  \text{larger(car, cat)}
  \]

- Through deductive reasoning
  
  \[
  \text{larger(house, cat)}
  \]
  
  can be derived.

**Abduction - 1**

- The reasoning method commonly used for generating explanations.
- Unlike deduction, it does not guarantee a true conclusion.
- While abductive inference is unsound, it is a quite useful technique and we use it often in our daily lives.
- Assuming the following rule
  
  \[
  \forall X \text{ (has_flu(X) }\rightarrow\text{ high_temperature(X))}
  \]

**Abduction - 2**

- Assuming the following axiom exists
  
  \[
  \text{high_temperature(john)}
  \]
- Abduction concludes
  
  \[
  \text{has_flu(john)}
  \]
- There could be other reasons why John has high temperature.

**Abduction - 3**

- Given the following:
  
  \[
  \{ A \rightarrow B \} \\
  B \text{ is true}
  \]
- Abduction allows us to say
  
  \[
  A \text{ is possibly true}
  \]

**Induction - 1**

- Reasoning from particular facts or individual cases to a general conclusion.
- The basis of scientific discovery.
- The most common form is:
  
  \[
  P(A) \text{ is true} \\
  P(B) \text{ is true}
  \]
- Then by induction we conclude
  
  \[
  \forall X, P(X) \text{ is true.}
  \]

**Induction - 2**

- Observing John over a period of time and noted that whenever he had high temperature, it turned out that he had flu.
- One could induce that
  
  \[
  \forall X, \text{ high_temperature(X) }\rightarrow\text{ has_flu(X)}
  \]
- Obviously this is not always true.
Inferences in Rule-based Systems - 1

- In rule-based ES, domain knowledge is represented as IF-THEN rules.
- Data are facts about the current situation.

```
Fact: A is x
Fact: B is y

Knowledge base
Rule: IF A is x THEN B is y
```

Inferences in Rule-based Systems - 2

```
A
X
B
Y
E

Inference Chain

Rule 1: IF X AND B AND E THEN Y
Rule 2: IF A THEN X
```

Forward Chaining (Data – Driven)

- Reasoning starts from data and proceeds forward with that data.
- Each time only the top most 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 rule can be fired.

```
Rule 1: Y & D → Z
Rule 2: X & B & E → Y
Rule 3: A → X
Rule 4: C → L
Rule 5: L & M → N
```

### Forward Chaining (Data – Driven)

Cycle 1

```
Database
Knowledge Base

Cycle 1
```

### Forward Chaining (Data – Driven)

Cycle 1

```
Database
Knowledge Base

Cycle 1
```
Forward Chaining (Data – Driven)

A \rightarrow X
X \& B \& E \rightarrow Y

L \& M \rightarrow N

Knowledge Base

Pass 1
Goal: Z

Sub-goal: Y

Pass 2

Sub-goal: X

Pass 3

Backward Chaining (Goal – Driven)

• The ES has the goal (a hypothetical solution) and the inference engine attempts to prove it.
• Find rules that might have the desired solution in the THEN parts.
• Subgoal(s) might be needed in proving the IF parts.
• Search rules that can prove the subgoals.
• The process repeats.

Backward Chaining (Goal – Driven)

A \rightarrow X
X \& B \& E \rightarrow Y

L \& M \rightarrow N

Knowledge Base

Pass 1
Goal: Z

Z

Pass 2
Sub-goal: Y

Pass 3
Sub-goal: X
Backward Chaining (Goal – Driven)

Knowledge Base

○ Y & D -> Z
○ C -> L
○ A -> X
○ X & B & E -> Y
○ L & M -> N

Sub-goal: X

Pass 4

Match

Sub-goal: Y

Pass 5

Z

Fire

Goal: Z

Pass 6

Forward or Backward?

• Forward Chaining – for analysis and interpretation.
  For example, DENDRAL determines the molecular structure of unknown soil based on its mass spectral data.

• Backward Chaining – for diagnosis
  For example, MYCIN diagnoses infectious blood diseases.

• Combining forward and backward chaining?

Backward Chaining (Goal – Driven)

Knowledge Base

○ Y & D -> Z
○ C -> L
○ A -> X
○ X & B & E -> Y
○ L & M -> N

Sub-goal: X

Pass 4

Match

Sub-goal: Y

Pass 5

Z

Fire

Goal: Z

Pass 6

Forward or Backward?

• Forward Chaining – for analysis and interpretation.
  For example, DENDRAL determines the molecular structure of unknown soil based on its mass spectral data.

• Backward Chaining – for diagnosis
  For example, MYCIN diagnoses infectious blood diseases.

• Combining forward and backward chaining?

Backward Chaining (Goal – Driven)

Knowledge Base

○ Y & D -> Z
○ C -> L
○ A -> X
○ X & B & E -> Y
○ L & M -> N

Sub-goal: X

Pass 4

Match

Sub-goal: Y

Pass 5

Z

Fire

Goal: Z

Pass 6

Forward or Backward?

• Forward Chaining – for analysis and interpretation.
  For example, DENDRAL determines the molecular structure of unknown soil based on its mass spectral data.

• Backward Chaining – for diagnosis
  For example, MYCIN diagnoses infectious blood diseases.

• Combining forward and backward chaining?

Readings

   On Frame-based Expert Systems (Essential Reading)
   On Handling Uncertainty in Rule-based Expert Systems (Highly Recommended Reading)