Expert Systems

• Computer-based systems that emulate the reasoning process of a human expert

• Different purposes:
  – Consulting
  – Diagnosis
  – Learning
  – Decision support
  – Designing, planning, etc.
Architecture of an Expert System
The Knowledge Base

• The KB, aka long-term memory, contains general knowledge belonging to the domain of interest

• Knowledge normally represented as (fuzzy) production rules
  – Connect antecedents with consequents, premises with conclusions or conditions with actions
  – Most common form: “IF A THEN B” (being A and B fuzzy sets)
The Facts Database

- Also known as short-term memory or blackboard interface
- Contains the current state (facts)
- It is updated after the firing of production rules
  - Previous state ‡ Rule firing ‡ Current state
- Previous facts are removed and the memory is updated with the current facts
The Inference Engine

- Operates on a series of production rules and makes fuzzy inferences. Approaches:
  - Data driven: supported by the generalized MP
    - The ES uses supplied data to evaluate relevant production rules and draw conclusions
  - Goal driven: exemplified by the generalized MT
    - The ES search for data specified in the IF clauses that will lead to the objective
    - These data can be found either in
      - The KB
      - THEN clauses of other production rules
      - Querying the user
The FES may use knowledge regarding the production rules in the KB. This includes meta-rules regarding:

- Stopping criteria
- Preconditions to fire determined rules
- Whether a fact should be inferred or requested from the user

Purpose: facilitate computation pruning unnecessary paths
Explanatory Interface

- Facilitates communication between the user and the expert system
- Enables the user to determine how the ES obtained intermediate or final conclusions
  - Or why specific information is being requested from the user
- Crucial for building user confidence in the system
- Useful for identification of errors, omissions, inconsistencies, etc.
Knowledge Acquisition Module

• Included only in some expert systems
• Makes it possible to update the KB or the metaknowledge base through interaction with experts
• Must implement suitable algorithms for machine learning (Socratic learning or example-based learning)
  – Artificial Neural Networks
  – Genetic Algorithms, etc.
Expert System Shell

• If the domain knowledge domain is removed from the ES, the remaining structure is a “shell”

• An inference engine embedded in an appropriate shell is reusable for different domains

• Examples of non-fuzzy and fuzzy shells
  – Prolog Expert System Shell (PESS)
  – Java Expert System Shell (JESS)
  – Fuzzy Prolog
Design of the Inference Engine

• When designing the fuzzy inference engine we have to consider the following:
  – Determine the type of inference engine
    • Data-driven (forward chaining)
    • Goal-driven (backward chaining)
  – Select a suitable fuzzy implication
    • Determine whether or not the MP or MT are required and choose an appropriate implication
      – MP is normally required for forward chaining
      – MT is normally required for backward chaining
Multi-Conditional Reasoning

- Fuzzy Expert Systems make use of approximate, multi-conditional reasoning:

  Rule 1: If $X$ is $A_1$, then $Y$ is $B_1$
  Rule 2: If $X$ is $A_2$, then $Y$ is $B_2$
  
  .......................
  Rule N: If $X$ is $A_n$, then $Y$ is $B_n$
  Fact: $X$ is $A'$
  
  ================
  Conclusion: $Y$ is $B'$
The Interpolation Method

• Method for multi-conditional reasoning

• Step 1: Calculate the degree of consistency between the given fact and the antecedent of each rule

\[ r_j (\mu_{A_j'}) = h(\mu_{A_j'} \cap \mu_{A_j}) \]

we take \[ T = \min_{x \in X} \sup \left[ \min(\mu_{A_j'}(x), \mu_{A_j}(x)) \right] \]

• Step 2: Calculate the conclusion by truncating each \( B_j \) to the value \( r_j(\mu_{A_j'}) \) and take the union of the truncated sets

\[ \mu_{B_j'}(y) = \sup_{j \in \{1, \ldots, n\}} \left[ \min(r_j(\mu_{A_j'}), \mu_{B_j}(y)) \right] \]
The Interpolation Method (Example)