Expert Systems (overview)

“Easy to simulate an expert, hard to simulate a child”

Key Points:

- “rules” or “knowledge base” – things that are always true; usually IF-THEN or Horn clauses
- “facts” or “data base” – things that are true for this scenario
- “inference engine” or “shell” – performs logical deductions, usually forward/backward chaining
- Extracting rules from a human expert can be difficult:
  - expert might not be available
  - expert might not be willing
  - experts often don’t agree
  - expert(s) often don’t know or can’t express what they know or do
- IF-THEN style rules may conflict. How to prioritize?
  - do them all, in some order
  - do most specific one
  - use certainty factors or fuzzy logic
- Inference Engine:
  - Propositional or Boolean logic usually not sufficiently expressive for expert systems
  - First-Order logic adds non-Boolean variables, making it possible to write rules that generalize
  - Forward-Chaining (e.g. CLIPS) fires each satisfied rule – generates all conclusions.
  - Backward-Chaining (e.g. Prolog) starts with query (goal), only fires rules needed to answer query.
- It is often possible to cast goal-like conditions in a forward-chaining system, if “retract” is available.
- Some expert systems can fire rules with “askable” items in the preconditions.