Expert Systems

- rules based expert systems
  - to make a machine to solve an intellectual problem, one had to know the solution
  - knowledge in a specific domain
- knowledge
  - theoretical or practical understanding of a subject
  - sum of what is currently known
- expert
  - someone with the knowledge

Expert Systems

- can generally express knowledge as rules
  - try teaching an alien how to cross the road
- rules in simplest form
  - if <antecedent> then <consequent>
  - can have multiple antecedents (and or)
- antecedent incorporates
  - object and value
  - similar for consequent

Expert Systems

- rules can be
  - Relation
    - if ‘fuel tank’ is empty then car is dead
  - Recommendation
    - if season is autumn & sky is cloudy & forecast is drizzle then ‘take umbrella’
  - Directive
    - if car is dead & ‘fuel tank’ is empty then action is ‘refuel the car’

Expert Systems

- Strategy
  - if car is dead then action is ’check the fuel tank’; step 1 is complete
  - if step 1 is complete & ‘fuel tank’ is full then action is ’check battery’; step 2 is complete
- Heuristic
  - if spill is liquid & ‘spill PH’ < 6 & ‘spill smell’ is vinegar then ‘spill material’ is ‘acetic acid’

Expert Systems

- knowledge from expert is added to computer
  - computer now acts intelligently
  - would like to add new knowledge
  - want computer to show knowledge in readable format
  - want computer to explain how conclusion reached
- most popular expert systems are rules based
  - as a result, a number of expert system shells are available
- expert system shell - expert system without the knowledge

Expert Systems

- structure of rule based system
  - knowledge base
  - database
    - set of facts used to match against conditions of the rules
  - inference engine
    - link rules in KB with facts in DB
  - explanation facilities
    - how a conclusion is reached, why a fact is needed
  - user interface
Expert Systems

- explanation capability
  - simplistic view
  - explanation requires basic understanding of domain
    - can attach fundamental principles to the rules (as text)
    - details depend on application and users
  - different from conventional programming
    - separation of knowledge and processing
    - can work on incomplete knowledge
      - may make mistakes, but conventional programs won’t work
    - changes are easy - just add new rules

- chaining of rules in inference
  - when conditional is matched with fact, then the rule is fired - then action is executed
  - firing a rule may add new facts to the database
  - produces an inference chain
    - indication of how the rules are fired

- consider database with facts A, B, C, D, E
- and the rules

Expert Systems

- built to perform at human level in narrow specialised domain
- characteristics
  - high quality performance - right decision
  - speed of performance
    - apply rules of thumb or heuristics where appropriate like humans
  - explanation capability
    - review of reasoning, explanation of decisions
    - trace of rules fired

- external program
- fact
- rule
- knowledge base database
- user
- expert

Expert Systems

- Expert

- inference engine
- explanation facility
- user interface
- developer interface
- external database
- external program

Expert Systems

- 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
- working with rule 3 fired first
  - inference chain shown is produced to arrive at a conclusion Z
Expert Systems

- **Forward Chaining**
  - example above used forward chaining
  - data driven reasoning
  - start from known data and proceed forward with that data
  - each time, only the topmost rule is fired
  - when fired, it adds a new fact to the database
  - rule can only be executed once
  - match-fire cycle stops when no further rules can be fired

- same example, but add two more rules
  - (4) C → L
  - (5) L & M → N

Expert Systems

- draw the forward chaining for above set
  - on each pass, identify which rules may be fired
  - execute the topmost on first
  - for above
    - on first pass, can fire rules 3 and 4, fire 3 first, then 4 on same cycle

- forward chaining
  - technique for gathering information and inferring what can be inferred
  - may lead to unnecessary execution of rules

Expert Systems

- backward chaining
  - goal driven reasoning
  - expert system has the goal, inference engine attempts to prove it
  - search the KB for rules with goals in their action parts
  - if rule is found and if condition matches data in the database
  - then rule is fired and goal is proved
  - usually, add rule to stack
  - then set up new goal (subgoal) and prove this
  - repeat until no rules found in KB to prove current goal

Expert Systems

- using above example, the goal is Z
  - more effective than backward chaining
  - fired three rules in backward chaining, 4 in forward

- which to use
  - if need to gather information first, forward chaining
  - if beginning with hypothetical solution and attempt to find facts to prove it, use backward chaining

  - forward - natural way for analysis and interpretation
  - backward - diagnostic purposes

Expert Systems

- can combine forward and backward chaining
  - basic inference mechanism is usually backward chaining
  - when new fact is added, use forward chaining to maximise use

- Thermostat
  - simple rules based expert system
  - uses Leonardo expert system shell

---

Rules:

**Rule 1**
if the day is Monday
or the day is Tuesday
or the day is Wednesday
or the day is Thursday
or the day is Friday
then today is a workday

**Rule 2**
if the day is Saturday
or the day is Sunday
then today is the weekend

**Rule 3**
if today is a workday
and time is 'between 9am and 5pm'
then operation is 'during business hours'

**Rule 4**
if today is a workday
and time is 'before 9am'
then operation is 'not during business hours'

**Rule 5**
if today is a workday
and time is 'after 5pm'
then operation is 'not during business hours'

**Rule 6**
if today is the weekend
then operation is 'not during business hours'
Expert Systems

• goal is given as seek thermostat_setting
• uses the objects
  – month, day, time, today, operation, season, thermostat_setting
  – each takes one of a set of allowed values
  – one object and its value constitute a fact
• there are 8 possible solutions for the thermostat setting
• system asks user to input data to reason with

Expert Systems

• data inputted
  – month, day, time
• from this, system reasons what the setting should be
• using forward chaining
  – will test rules and see if the known facts match
  • fire any goals that match
  • arrive at solution
  – rules 1, 3, 9, 17 fired
Expert Systems

- Pass 6,
  - operation instantiated to “during business hours”
- Pass 7, rule 17
  - thermostat setting instantiated to 18 degrees

- Conflict Resolution
  - when firing one rule affects the activation of other rules
  - should only fire one rule if two conflict

Conflict Resolution

- Rule 1 if the traffic light is green then action is go
- Rule 2 if traffic light is red then action is stop
- Rule 3 if traffic light is red then action is go

- there is a conflict with rule 2 and 3
- inference engine must decide which to fire
- in forward chaining
  - both rules should be fired
  - order of rules is now important
- resolution strategy
  - establish a goal and stop firing rules when goal met

Conflict Resolution

- other strategies
  - prioritise rules
    - place them in appropriate order in the database
  - fire the most specific rule
    - longest matching strategy
      - a specific rule processes more information than a general one
    - fire rules using most recent data
      - relies on time tags
  - above are fine when number of rules is relatively small

Metaknowledge

- when number of rules grows
  - difficult for knowledge engineer to oversee the program
  - transfer some responsibility to the expert system
    - supply knowledge about the knowledge it possesses or metaknowledge
    - represented by metarules
      - determines strategy for the use of task specific rules in the system
      - should be given highest priority in system

Metaknowledge

- example
  - MetaRule 1
    - Rules supplied by experts have higher priorities than rules supplied by novices
  - MetaRule 2
    - Rules governing the rescue of human lives have higher priorities than rules concerned with clearing overloads on power system equipment
- Knowledge Engineer transfers knowledge of the domain expert to the expert system
  - learns how the rules are used
  - gradually creates new body of knowledge - knowledge of the overall behaviour of the expert system

Expert Systems

- advantages of rules based systems
  - natural knowledge representation
    - in this situation, this is what is done - easy to represent as rules
  - uniform structure
    - self-documenting, independent pieces of knowledge
  - separation of knowledge from processing
  - dealing with incomplete and uncertain knowledge
    - uncertainty with probability or certainty factors

Damien Costello, Dept of Computing & Maths, GMIT
Expert Systems

• disadvantages
  – opaque relations between rules
    • difficult to observe how individual rules affect overall strategy
  – ineffective search strategy
    • exhaustive search through all rules on each cycle
    • slows down the reasoning process
  – inability to learn
    • in general, rules based systems cannot learn from experience
    • human expert will break the rules if necessary