EXPERT SYSTEM DESIGN

TERMINOLOGY

A. EXPERT SYSTEM - EMULATES A HUMAN EXPERT

1. SPECIALIZED KNOWLEDGE

B. KNOWLEDGE-BASED SYSTEM - A SYSTEM THAT USES EXPERT SYSTEM DESIGN TECHNOLOGY BUT DOES NOT NECESSARILY CONTAIN THE KNOWLEDGE OF A HUMAN EXPERT.

C. THE TERMS KNOWLEDGE-BASED SYSTEM AND EXPERT SYSTEM ARE USED INTERCHANGEABLY.

D. KNOWLEDGE ENGINEER

E. KNOWLEDGE ENGINEERING

F. KNOWLEDGE ACQUISITION

G. RULE INDUCTION – AN EXPERT SYSTEM LEARNS RULES BY EXAMPLE. THE SYSTEM LEARNS RULES FROM TABLES OF DATA.

H. TWO TYPES OF KNOWLEDGE

1. DEEP (CAUSUAL) - EXPERT SYSTEMS LACK THIS TYPE OF KNOWLEDGE

2. SHALLOW (HEURISTIC)

I. META-KNOWLEDGE - KNOWLEDGE ABOUT KNOWLEDGE
   METARULE - KNOWLEDGE ABOUT RULES

J. EXPLANATION -

K. HYPOTHETICAL REASONING - REASONING THAT ALLOWS THE USER TO ASK "WHAT IF" TYPES OF QUESTIONS.
COMPONENTS OF AN EXPERT SYSTEM

A. KNOWLEDGE BASE
   1. FACTS
   2. RULES

B. INFERENCE ENGINE

C. USER INTERFACE

ADVANTAGES OF AN EXPERT SYSTEM

A. EXPLANATION
B. MULTIPLE EXPERTISE
C. PERMANENCE
D. INCREASED RELIABILITY
E. INTELLIGENT TUTORING
F. REDUCED COST
G. REDUCED DANGER
H. OTHERS

ADVANTAGES OF RULE-BASED SYSTEMS

A. SEPARATION OF KNOWLEDGE AND REASONING STRATEGY

B. RAPID PROTOTYPING CAPABILITIES

C. INCREMENTAL GROWTH

D. COGNITIVE PSYCHOLOGISTS USE RULES AS A MODEL TO EXPLAIN HUMAN INFORMATION PROCESSING. COGNITIVE PSYCHOLOGISTS STUDY HOW HUMANS PROCESS INFORMATION.

   1. LONG-TERM MEMORY
   2. SHORT-TERM MEMORY
CLASSIFYING EXPERT SYSTEMS

A. CONFIGURATION -  
B. DIAGNOSIS -  
C. INSTRUCTION -  
D. INTERPRETATION -  
E. MONITORING -  
F. PLANNING -  
G. PROGNOSIS -  
H. REMEDY -  
I. CONTROL -  

EXPERT SYSTEM APPLICATION DOMAINS

A. ILL-STRUCTURED PROBLEM REQUIRING NON-CONVENTIONAL PROGRAMMING TECHNIQUES  
B. WELL-BOUNDED APPLICATION DOMAIN  
C. IS THE DOMAIN KNOWLEDGE HEURISTIC  
D. IS THERE A WILLING EXPERT AND IS THE EXPERT ABLE TO ARTICULATE HIS KNOWLEDGE.  
E. IS THERE A NEED FOR AN EXPERT SYSTEM?  

EXPERT SYSTEM BUILDING TOOLS

A. PROGRAMMING LANGUAGES  
B. EXPERT SYSTEM SHELLS  
C. KNOWLEDGE ENGINEERING ENVIRONMENTS
INFERENC ENGINE STRATEGIES

1. FORWARD CHAINING (FACTS TO CONCLUSIONS)

2. BACKWARD CHAINING (REASONING IN REVERSE. WHAT FACTS MUST BE TRUE TO MAKE THE CONCLUSION TRUE).

THE RETE ALGORITHM

A. BASED ON THE RECOGNIZE-ACT CYCLE
B. USES A NETWORK TO STORE INFORMATION ABOUT RULES
C. STATIC DATA THAT DOES NOT CHANGE FROM CYCLE TO CYCLE IS IGNORED WHEN LOOKING FOR NEW RULE MATCHES.

PROGRAMMING PARADIGMS

A. PROCEDURAL PARADIGMS - PROGRAMMER SPECIFIES HOW A PROBLEM SOLUTION WILL BE CODED.

1. IMPERATIVE PROGRAMMING - STATEMENT ORIENTED

   STATEMENTS ARE COMMANDS TO THE COMPUTER TELLING IT WHAT TO DO.

   TOP-DOWN DESIGN

   C ADA FORTRAN PASCAL...

2. FUNCTIONAL PROGRAMMING - COMBINE SIMPLE FUNCTIONS TO GIVE MORE COMPLEX FUNCTIONS.

   BOTTOM-UP DESIGN

   LISP, APL

FUNCTION = A RULE THAT MAPS MEMBERS OF ONE SET (THE DOMAIN) INTO ANOTHER SET (THE CODOMAIN)

F:R------>R

F(X) = X**2
DOMAIN = SET OF ALL X VALUES
RANGE = SET OF ALL Y VALUES

PURE LISP == PURELY FUNCTIONAL
COMMON LISP == CONTAINS NON-FUNCTIONAL ADDITIONS LIKE SET

B. NONPROCEDURAL PARADIGMS- EMPHASIZE WHAT IS TO BE ACCOMPLISHED, NOT HOW IT IS ACCOMPLISHED.

1. DECLARATIVE PROGRAMMING - UNDERLYING MECHANISM FOR SATISFYING A GOAL IS TRANSPARENT TO THE USER.

2. OBJECT-ORIENTED PROGRAMMING - DATA ARE CONSIDERED TO BE OBJECTS. OPERATIONS ON OBJECTS ARE DEFINED.
   a. DATA ABSTRACTION
   b. SPECIALIZATION

3. LOGIC PROGRAMMING

   PROLOG:
   - HORN CLAUSES
   - BACKWARD CHAINING
   - UNIFICATION
   - BACKTRACKING

   PROLOG CAN BE CONSIDERED AN EXPERT SYSTEM SHELL.

4. EXPERT SYSTEMS
   a. RULE-BASED
   b. FRAME BASED
   c. INFERENCE NETS

   TABLE PAGE 47

5. NONDECLARATIVE PROGRAMMING

6. INDUCTION-BASED PROGRAMMING
   PROGRAMS LEARN BY EXAMPLE.
INTRODUCTION TO CLIPS

I. CLIPS IS AN EXPERT SYSTEM SHELL WITH THREE COMPONENT PARTS
   A. FACT-LIST
   B. KNOWLEDGE-BASE
   C. INFERENCE ENGINE - FORWARD CHAINING

II. SYNTAX OF CLIPS
   A. CASE-SENSITIVE
   B. STRINGS ARE ENCLOSED IN DOUBLE QUOTES
   C. FACTS - ONE OR MORE FIELDS ENCLOSED IN MATCHING LEFT AND
      RIGHT PARENTHESES.

      (DAY THURSDAY) DAY IS A RELATION NAME

III. TO ENTER AND EXIT
   A. ENTER
   B. EXIT (EXIT)

IV. ADDING AND REMOVING FACTS
   A. (ASSERT (DAY THURSDAY)
      (WEATHER SUNNY AND WARM)
      (PLAN GO FISHING) ..... )

   B. (RETRACT <<< FACT-INDEX >>> )

      WHERE THE FACT INDEX NUMBER IS OBTAINED FROM THE COMMAND
      (FACTS) WHICH LISTS ALL OF THE FACTS FOUND IN THE SYSTEM.

      (RETRACT 2 7 5)
V. GENERAL RULE FORMAT

A. (DEFRULE <RULE NAME> { OPTIONAL COMMENT } ; PATTERNS
   ( )
   ( )

   ; THEN ARROW
   =>

   ; ACTIONS
   ( )
   ( )
   .
   .

B. A RULE WITH NO PATTERNS WILL HAVE THE DEFAULT PATTERN
   (INITIAL-FACT) ADDED TO ITS PATTERN LIST

VI. RULE EXECUTION

A. (RUN <LIMIT>) ; LIMIT= NUMBER OF RULES TO BE FIRED

B. RULES ONLY SEE FACTS THAT HAVE BEEN ENTERED AFTER THE RULES
   ARE ENTERED.

C. TO LIST THE RULES PRESENTLY ON THE AGENDA
   (AGENDA)
   GIVES: SALIENCE NAME FACT-IDENTIFIERS

D. REFRACTION IS USED TO AVOID TRIVIAL LOOPS

E. (RULES) DISPLAYS ALL RULES

F. (PPRULE RULE-NAME) TO DISPLAY ONE RULE
VII. LOADING AND SAVING FILES

A. `(LOAD "FILE-NAME")` TO LOAD A FILE OF RULES

B. `(LOAD-FACTS "FILE-NAME")` TO LOAD A LIST OF FACTS

C. `(SAVE "FILE")` SAVES ALL RULES TO A FILE

D. COMMENTS ;;;;;

VIII. PRINTOUT COMMAND

A. `(PRINTOUT T "THIS IS A TEST" crlf)

IX. DEFFACTS

A. `(DEFFACTS FCTS "INPUT A FEW FACTS"
    (DAY FRIDAY)
    (WEATHER CLOUDY)
    (TEMPERATURE 75) )

B. DEFFACTS MAY BE LOADED IN THE SAME FILE AS THE RULES FILE.
   A
   `(reset)` SHOULD THEN BE PERFORMED

   EACH TIME YOU WISH THE INITIAL FACTS IN THE DEFFACTS STATEMENT
   TO BE REALIZED, A `(reset)` SHOULD BE PERFORMED.

C. `(list-deffacts)

D. `(ppdefact name)

E. `(undeffacts <deffacts-name>)`

X. REMOVING RULES AND RULES AND FACTS

A. `(excise <RULE-NAME>)`

B. `(clear)` -- removes all information in the CLIPS environment

XI. DEBUGGING

A. `(watch {facts, rules, activations, all})`
B. (matches rule-name)

C. (set-break <rule-name>)}
PATTERN MATCHING

I. USING VARIABLES

A. THE ? FOLLOWED BY ONE OR MORE CHARACTERS SPECIFIES A VARIABLE VALUE.

(defrule name-print "print a name"
 (isa dog ?name)
 =>
 (printout t ?name " is a dog " crlf))

B. FACT ADDRESSES AND RETRACTION

1. TO RETRACT A FACT ITS ADDRESS MUST BE KNOWN
   A (RETRACT ADDRESS) COMMAND THEN ELIMINATES THE FACT

(defrule name-print "print and delete a name"
   ;;
   ?old-fact <- (is dog ?name)
   =>
   (retract ?old-fact)
   (assert (dog ?name)
   (printout t ?name " is a dog " crlf))

C. SINGLE FIELD WILDCARDS

1. A ? IS USED TO SPECIFY A SINGLE FIELD WILD CARD

   (PERSON ?NAME ? ?)

   WILL MATCH WITH

   (PERSON SAM BLUE BLACK)
   (PERSON BILL BROWN GREEN)
   .
   ETC..
D. MULTIFIELD WILDCARDS AND VARIABLES

1. A $? REPRESENTS ZERO OR MORE OCCURANCES OF A FIELD.

   (defrule info
      (?name  $?info)
   =>
      (printout t "the person with name" ?name crlf)
      (printout t "has data information" crlf)
      (printout t $?info))
DEALING WITH UNCERTAINTY

A. PROBABILITY-BASED TECHNIQUES
   1. OBJECTIVE PROBABILITY
   2. EXPERIMENTAL PROBABILITY
   3. SUBJECTIVE PROBABILITY

   BAYES' THEOREM IS BEST WHEN IT CAN BE APPLIED

B. INEXACT OR HEURISTIC REASONING (DEAL WITH HUMAN BELIEF RATHER THAN CLASSIC FREQUENCY)

   AS AN EXAMPLE, WITH CERTAIN INEXACT REASONING STRATEGIES THE SUM TOTAL OF ALL POSSIBLE OUTCOMES NEED NOT ADD TO ONE AS IT DOES WITH CLASSICAL PROBABILITY THEORY.

   1. UNCERTAINTY OF RULE ANTECEDENTS.
   2. UNCERTAINTY OF RULE CONSEQUENTS.
   3. UNCERTAINTY OF THE MEANING OF THE RULES.
CERTAINTY FACTORS

A. DIFFICULTIES WITH BAYESIAN STATISTICS (FIELD OF MEDICINE)

1. EXTENDS OVER ALL DISEASES
2. IMPOSSIBLE TO DETERMINE VALUES FOR ALL OF THE PROBABILITIES SEEN WITHIN THE FORMULA
3. MEDICAL EVIDENCE TENDS TO ACCUMULATE BY BITS AND PIECES.

B. BELIEF VS. DISBELIEF

1. IF P(GRADUATING | "A" IN PRESENT COURSE ) = .7

   THIS NEED NOT IMPLY THAT THE PROBABILITY OF NOT GRADUATING GIVEN AN "A" IS .3

   CERTAINTY FACTORS ALLOW A BELIEF TO BE EXPRESSED WITHOUT IMPLYING A VALUE TO THE DISBELIEF OF THE SAME FACT.

C. CF(H,E) = MB(H,E) - MD(H,E)

D. CALCULATING CERTAINTY FACTORS

   CF= MB - MD

   -----------------
   1 - min(MB,MD)
DEMPSTER-SCHAFER THEORY

1. A DEFINED ENVIRONMENT (UNIVERSE) REPRESENTING A FIXED SET OF MUTUALLY EXCLUSIVE AND EXHAUSTIVE ELEMENTS.

2. MASS = DEGREE OF BELIEF IN EVIDENCE.

3. CLASSICAL PROBABILITY THEORY - ANYTHING NOT SUPPORTING MUST BE REFUTING SINCE IGNORANCE IS NOT ALLOWED.

4. AIRLINER IDENTIFICATION..

   AN AIRCRAFT IS TO SEND AN ID CODE IF IT IS FRIENDLY.

   ENVIRONMENT = {AIRLINER, BOMBER, FIGHTER}

   CLASSICAL PROBABILITY
   ---------------------

   P(HOSTILE|NO-RESPONSE) = .7

   P(NOT-HOSTILE|NO-RESPONSE) = .3 { CLASSICAL PROBABILITY REQUIRES THIS PROBABILITY MEASURE }

   DEMSTER - SCHAFER
   -----------------

   M1(HOSTILE) = M1(BOMBER, FIGHTER) = .7

   M1(O) = 1 - .7 = .3 "MEANS NONBELIEF" WHICH IS NEITHER BELIEF OR DISBELIEF IN THE EVIDENCE TO A DEGREE OF .3. THIS MEANS THAT PART OF THE .3 COULD STILL REPRESENT A BELIEF IN THE EVIDENCE.

   DISBELIEF + ADDITIONAL BELIEF = .3

6. SUM M(X) = 1 P(0) IS POWER SET OF X IN P(0)
7. COMBINING EVIDENCE

EVIDENTIAL INTERVALS
---------------------

BEL = LOWER LEVEL
PLS = UPPER BOUND

[ .90 , 1 ]

.90 = BEL SUPPORT LEVEL (BEL)
1.0 = PLAUSIBILITY (PLS)

BEL = THE TOTAL BELIEF OF A SET AND ALL OF ITS SUBSETS.

BEL(X) = SUM M(Y)
Y SUBSET OF X

BEL(X) = M1(B,F) + M1(B) + M1(F) = .7 + 0 + 0

MASS DEFINED IN TERMS OF BELIEF ON TOP OF PAGE 285

8. COMBINING TWO BELIEF FUNCTIONS

BEL1 + BEL2 (B,F) =
(M1 + M2 (B,F) ) + ( M1 + M2 (B))
+ (M1 + M2 (F))

9. EVIDENTIAL INTERVAL COMPUTATIONS..

EI(S) = THE EVIDENTIAL INTERVAL OF SET S

EI(S) = [BEL (S),1 - BEL(S') ]

10. PLAUSIBILITY(X) = 1 - BEL(X') = 1 - SUM (M (X')
Y SUBSET X
FUZZY LOGIC

REASONING USING NATURAL LANGUAGE WHERE WORDS HAVE AMBIGUOUS MEANING.

"COLD SMALL GOOD BAD"

OBJECTS BELONG PARTIALLY OF A SET. THE MEMBERSHIP FUNCTION OR COMPATIBILITY FUNCTION IS DEFINED AS::

\[ U(X) : X \rightarrow [0,1] \quad \{ \text{WHERE } [0,1] \text{ IS AN INTERVAL OF MEMBERSHIP} \} \]

PAGE 295 -- MEMBERSHIP FUNCTION FOR TALL -- FIG. 5-8

THE GENERAL MEMBERSHIP FUNCTION THAT IS FREQUENTLY USED IS THE S - FUNCTION SHOWN ON 296. FIG. 5-9.

THE MEMBERSHIP FUNCTION FOR FIG 5-8 IS DEFINED ON PAGE 297.

ANOTHER COMMONLY USED MEMBERSHIP FUNCTION IS SHOWN IN FIG 5-10 ON PAGE 298.

A FUZZY SUBSET TALL = \{ 0/5, .125/5.5 , .5/6 ,.875/6.5, 1/7, 1/8\}

FUZZY SET OPERATORS ON 302 - 307..

FUZZY RELATIONS: