UNIT-III KNOWLEDGE INFERENCE

Knowledge Representation - Production based Systems, Frame based Systems. Inference - Backward chaining, Forward chaining, Rule value approach, Fuzzy Reasoning. Certainty Factors, Bayesian Theory - Bayesian Network - Dempster - Shafer Theory.

Production based systems:

Production systems are also called as the rule based systems. This type of system uses Knowledge encoded in the form of production rules, that is, if ... then rules.

IF condition1 .... condition n

THEN Take Action 4.

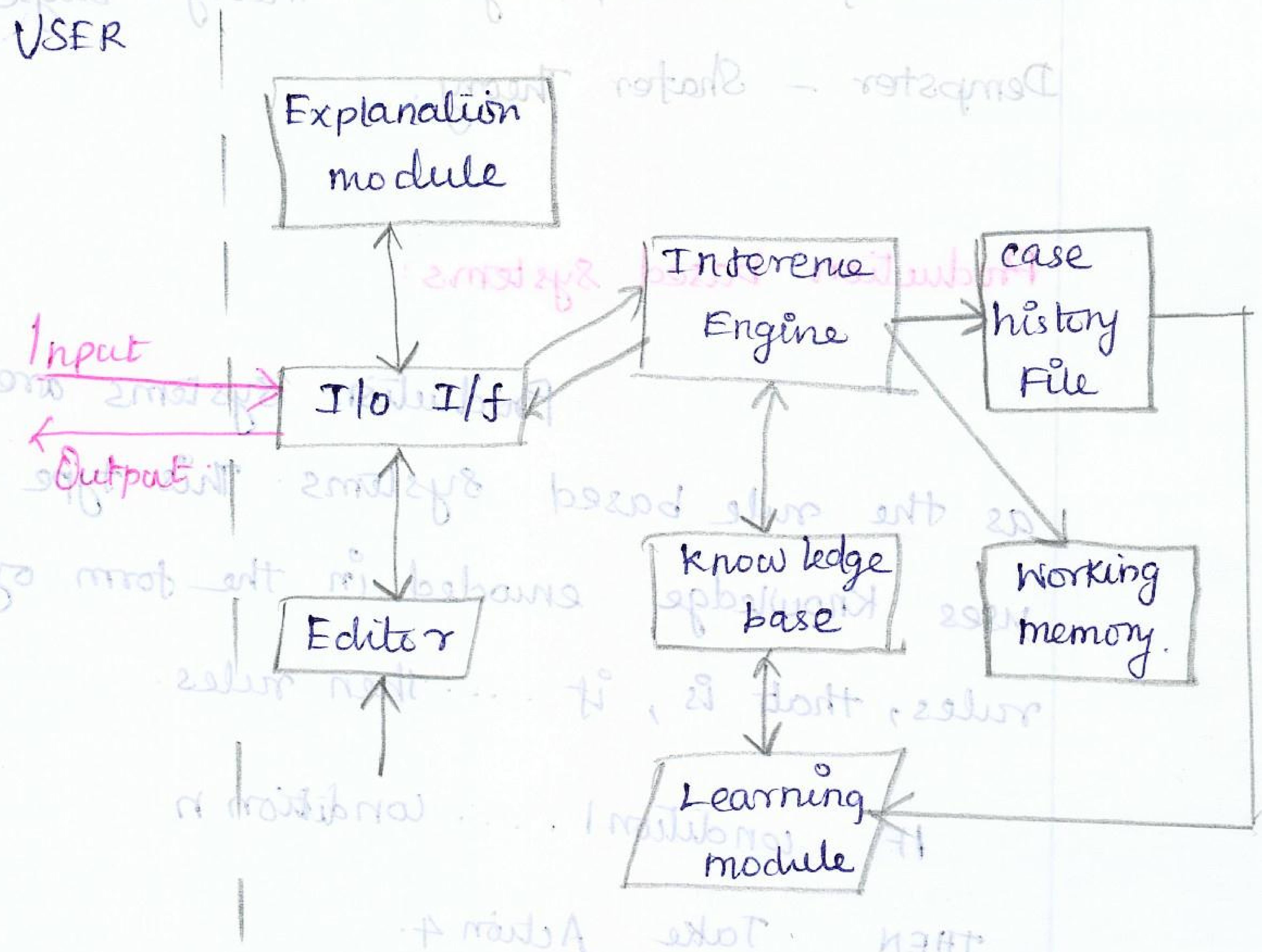
Example:

IF The temperature is greater than 200°, and
The water level is low

THEN Open the Safety valve.

Inference in production systems is accomplished by a process of chaining through the rules recursively, The inference process is typically carried out in an interactive mode with the user providing input parameters needed to complete the rule chaining process.

Example Production System:— Inference Process.



The inference engine accepts user input queries and responses to questions through the I/o interface and uses this dynamic information together with the static knowledge (the rules and facts) Stored in the Knowledge base. The knowledge base contains the facts and rules about some specialized Knowledge domain which is used to deriv

The inferring process is carried out recursively in three stages. (1) Match (2) Select and (3) Execute.

Example knowledge base:

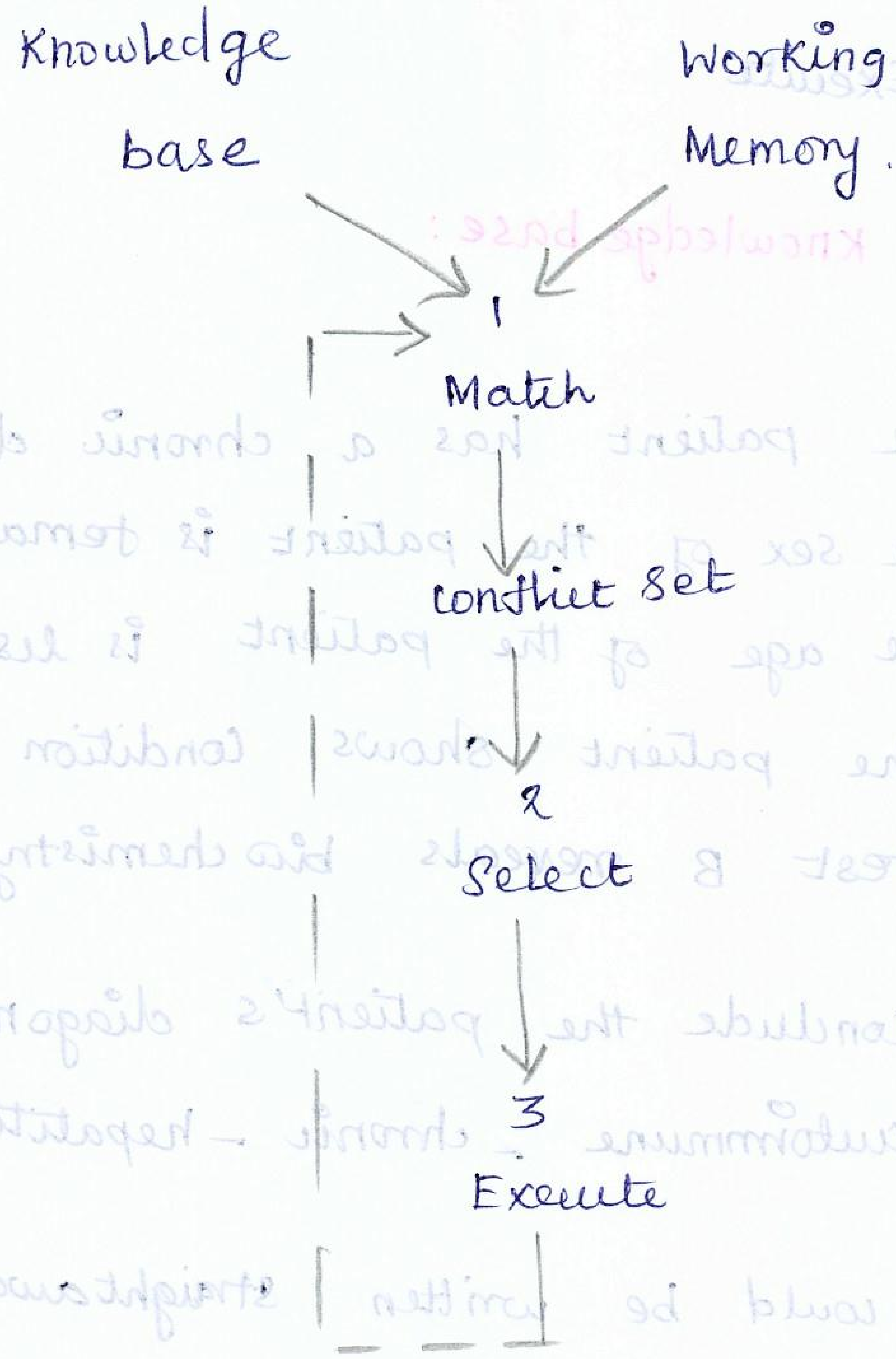
IF The patient has a chronic disorder, and
the sex of the patient is female, and
the age of the patient is less than 30, and
the patient shows condition A, and
test B reveals biochemistry condition C

THEN conclude the patient's diagonosis is
autoimmune - chronic - hepatitis.

This rule could be written straightaway in PROLOG as,

```
conclude (Patient, diagonosis, autoimmune_chronic_
                                          hepatitis):-
    Same (Patient, disorder, chronic),
    Same (Patient, sex, female),
    lessthan (patient, age, 30),
    Same (patient, symptom_a, value_a),
    Same (patient, biochemistry, value_c).
```

Production System Inference Cycle:



During the match stage, the contents of working memory are compared to facts and rules contained in the knowledge base. When consistent matches are found, the corresponding rules are placed in a conflict set. To find an appropriate and consistent match, substitutions may be required. Once all the matched rules have been added to the conflict set during a given cycle, one of the rules is selected for execution:

(i) Most recent use
(ii) Rule condition speufity
(iii) Smallest Rule number

The selected rule is then executed and the right hand side (or) action part of the rule is then carried out.

Explanation Module:

It provides the user with an explanation of the process (how and why) when requested.

how query → It permits the user to actually see the reasoning process followed by the system in arriving at the conclusion. If the user does not agree with the reasoning steps presented they may be changed using the editor.

Why query → The explanation module must be able to explain why certain information is needed by the inference engine to complete a step in the reasoning process before it can proceed.

Editor - Building a Knowledge base.

base, to delete outmoded rules, (or) to modify existing rules in some way.

I/O Interface:

The I/O interface permits the users to communicate with the systems in a more natural way by permitting the use of simple selection menus (or) the use of a restricted language which is close to a natural language.

Learning Module / History File

These are not common components of expert system. When they are provided, they are used to assist in building and refining the knowledge base.

Frame System Architecture:

Instead of rules these systems employ more structured representation schemes like associative (or) semantic network, frame and rule structures, decision trees (or) even specialized networks like neural networks.

Frames are structured sets of closely related knowledge, such as an object (or) concept name, the objects main attributes and their corresponding values, and possibly some attached procedures.

The attributes, values, procedures are stored in specified slots and slot facets of the frame.

Individual frames are usually linked together as a network much like the nodes in the associative network.

General Frame Structure:

(<frame name>

(<slot1> (<facet1> <value1> ... <valuen>)

:

(<slot2> (<facet1> <value1> ... <value n>)

:

)

Example

```
(ford (figo (VALUE car))
      (COLOR (VALUE Silver))
      (MODEL (VALUE 4-door))
      (GAS-MILEAGE (DEFAULT fget))
      (RANGE (VALUE if-needed))
      (WEIGHT (VALUE 2600))
      (FUEL-CAPACITY (VALUE 18))
      :
)
```

Frame can be represented using some high level language [Eg. LISP].

For example a function which defines a frame might be called with,

```
(fdefine f-name <Parents> <slots>)
```

New frame name

Parent frames which are having link with the new frame.

Functions in Frames:

```
(fget f-name slotname facetname)
```

(fslots f-name) → Returns names of slots

(ffacets f-name slot-name) → Returns name of the facets.

(fput f-name slotname facetname) → Adds data to a specified location

(fremove f-name slotname facet name) → Removes data from specified location.

Example:

```
(fdefine general-train land-transport
    (type (VALUE passenger))
    (class (VALUE firstclass secondclass sleeper))
    (food (restaurant (VALUE hot.meals))
        (fastfood (VALUE cold-snacks))
        - -
        -
        - -
```

Forward chaining & Backward chaining.

Forward chaining:

Working from the facts to a conclusion. Sometimes called the data driven approach. To chain forward, match data in working memory agains-

Backward chaining:

Working from the conclusion to the facts. Some times called the goal-driven approach.

→ To chain backward, match a goal in working memory against conclusions of rules in the rule base

→ when one of them fires, this is liable to produce more goals.

→ So the cycle continues.

Eg: If corn is grown on poor soil, then it will get blackfly.

If soil hasn't enough nitrogen, then it is poor soil.

FC → This soil is low in nitrogen; therefore this is poor soil; therefore corn grown on it will get blackfly.

BC → This corn has blackfly therefore it must have been grown on poor soil; therefore the soil must be low in nitrogen.

Forward chaining Algorithms:

function FOL-FC-ASK (KB, α) returns a substitution (or) false

repeat until new is empty.

new ← { }

for each θ such that $(P_1 \wedge \ldots \wedge P_n)\theta = (P_1' \wedge \ldots \wedge P_n')\theta$

for some $P_1' \ldots P_n'$ in KB

$q' \leftarrow$ SUBST (θ, q)

if q' is not a renaming of a sentence already in KB or new, then do

add q' to new

$\phi \leftarrow$ UNIFY (q', α)

if ϕ is not fail then return ϕ

add new to KB

return false.

Backward chaining:

function FOL-BC-ASK (KB, goals, θ) returns a set of substitutions

inputs: KB, a Knowledge Base

goals, a list of conjuncts forming a query

θ, the current substitution, initially the empty substitution {}

local variables: ans, a set of substitutions, initially empty

if goals is empty then return $\{\theta\}$

$q' \leftarrow$ SUBST $(\theta$, FIRST (goals))

for each r in KB where Standardize_Apart (r) =

$(P_1 \wedge \ldots \wedge P_n \Rightarrow q)$ and $\theta' \leftarrow$ Unify (q, q') succeeds.

ans $\leftarrow$ FOL-BC-ASK (KB, $[P_1 \ldots P_n]$ | REST (goals)]

Choice of Strategy:

Backward chaining is the best choice:

(i) Goal is given in the problem statement (or Guessed at the beginning of the consultation.

(ii) If the system asks some piece of data (For Eg, Test the gram of blood rather than expecting all the facts to be presented to it)

Forward chaining is the best choice if:

i) All the facts are provided.

(ii) Many possible goals and a smaller no. of patterns of data.

(iii) There isn't any sensible way to guess what the goal is at the Beginning of the consultation.

Mixed chaining:

The strategy sometimes to chain in one direction then switch to the other direction, so that:

1) the diagonosis is found with maximum efficiency

Bayes Rule:

An important goal for many problem solving systems is to collect evidence as the system goes along and to modify its behavior on the basis of the evidence that can be defined in bayes theory.

$$P(H/E) = \frac{P(E|H)}{P(E)} P(H)$$

$P(H|E)$ — Posterior probability of hypothesis given evidence.

$P(E|H)$ — Likelihood of evidence given hypothesis

$P(E)$ — Normalizing constant

$P(H)$ — Prior probability of hypothesis

Example:

$$P(\text{cancer} \mid \text{Test}(+ve)) = \frac{P(\text{Test}(+ve) \mid \text{cancer})\, P(\text{cancer})}{P(\text{Test}+)}$$

$P(\text{Test}+ \mid \text{cancer}) = 0.9$, $P(\text{cancer}) = 0.0001$,

$P(\text{Test}+) = 0.0010899$

$$P(\text{cancer} \mid \text{Test}+) = 0.9 \times 0.0001 / 0.0010899$$

$$= 0.08$$

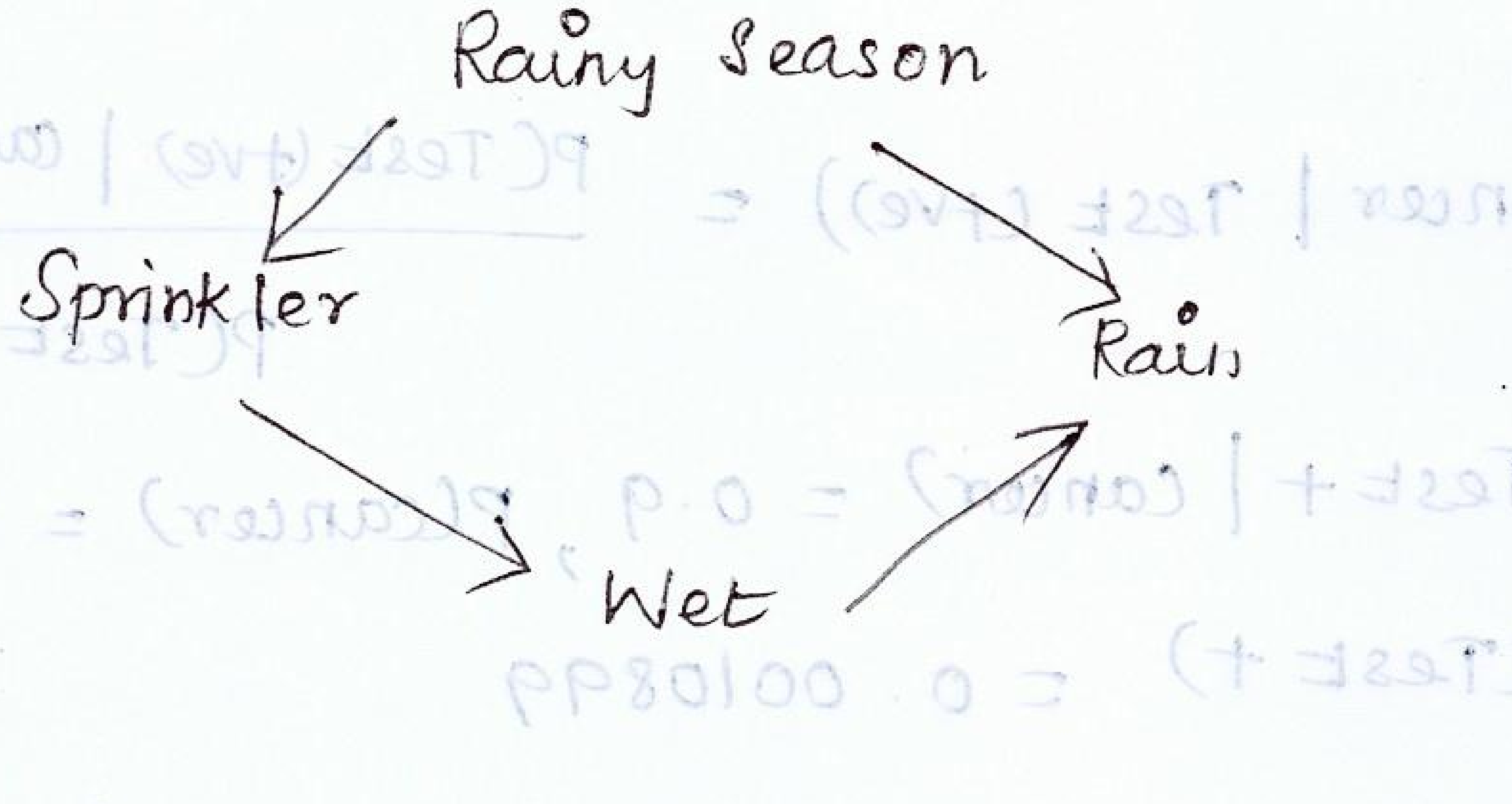
13

Bayesian Networks:

The main idea is that to describe the real world, it is not necessary to use a huge joint probability table in which we list the probabilities of all conceivable combinations of the event. Most events are conditionally independent of most other ones, so their interactions need not be considered. Instead we can use a more local representation in which we will describe clusters of the events that interact.

One usual way to portray the problem domain is with a network of nodes which represent propositional variables connected by arcs which represent causual influences (or) dependencies among the nodes. The strengths of the influences are quantified by conditional probabilities of each variable.

For Example:



Attribute	Probability
P(Wet | Sprinkler, Rain)	0.95
P(Wet | Sprinkler, ¬ Rain)	0.9
P(Wet | ¬ Sprinkler, Rain)	0.8
P(Wet | ¬ Sprinkler, ¬ Rain)	0.1
P(Sprinkler | Rainy Season)	0.0
P(Sprinkler | ¬ Rainy Season)	1.0
P(Rain | Rainy Season)	0.9
P(Rain | ¬ Rainy Season)	0.1
P(Rainy Season)	0.5

Once such a network is constructed an inference engine can use it to maintain and propagate beliefs. When a new information is received, the effects can be propagated throughout the network untill equilibrium probabilities are reached.

To use the type of probablistic inference it is first necessary to assign probabilities to all basic facts in the knowledge base as mentioned above.

Finally when the outcome of an inference chain results in one or more proposed conclusions, the alternatives must be compared, and one or more selected on the basis of its likelihood.

Dempster - Shafer Theory :

A general (abstract) formulization Sees Belief function as a Special case of upper Probabilities.

Definition: A belief function "Bel" defined on a Space 'W' Satisfies the following three properties.

B1 : $Bel(\phi) = 0$

B2 : $Bel(W) = 1$

B3 : $Bel(U_1 \cup U_2) \geq Bel(U_1) + Bel(U_2) - Bel(U_1 \cap U_2)$

$$Bel(U_1 \cup U_2 \cup U_3) \geq Bel(U_1) + Bel(U_2) + Bel(U_3) - Bel(U_1 \cap U_2) - Bel(U_1 \cap U_3) - Bel(U_2 \cap U_3) + Bel(U_1 \cap U_2 \cap U_3).$$

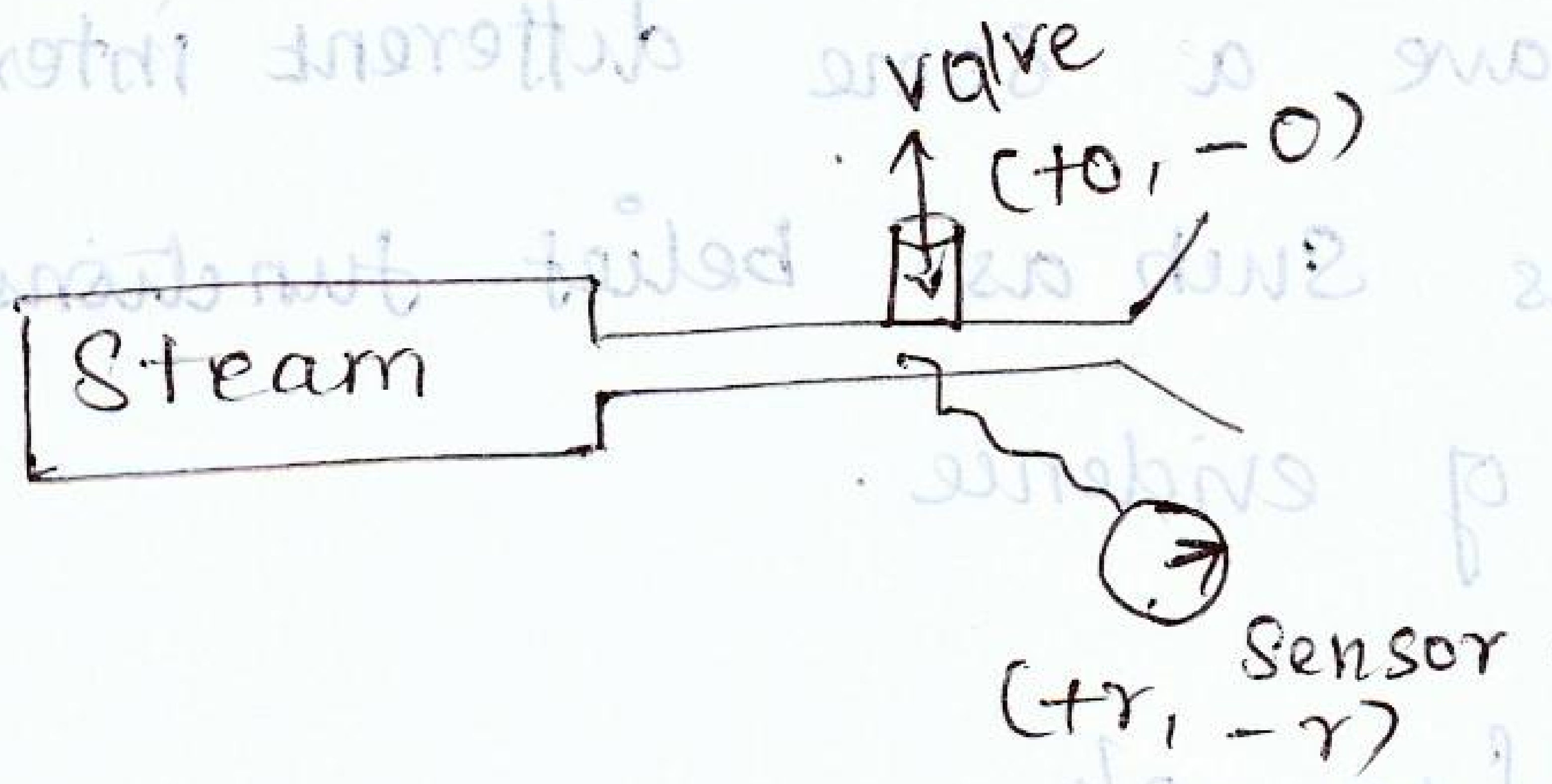
Example:

Suppose one is interested in the question whether the value is closed (or) open.

The only information about the state of the value is provided by a sensor.

It is known that the sensor is unreliable in exactly 20% of the cases

$r \rightarrow$ hidden parameter.



$W = \{+o, -o\}$; $H = \{+r, -r\}$

$\mu(+r) = 0.8$ $\mu(-r) = 0.2$

Mapping $\Gamma : H \Rightarrow 2^W - \{\phi\}$

$\Gamma(+r) = \{o\}$

$\Gamma(-r) = \{+o, -o\}$

	Bel	Pl
$\{+o\}$	0.8	1
$\{-o\}$	0	0.2

$Bel(U) \underset{def}{=} \mu(\{h \in H : \Gamma(h) \subseteq U\})$

$Pl(U) \underset{def}{=} \mu(\{h \in H : \Gamma(h) \cap U \neq \phi\})$

Shafer's Interpretation:

(i) In Dempster's scenario belief function are constructed by means of multi valued mappings.

(ii) Bel and its dual, Pl(Plausibility), are special kind of lower/upper probability functions:

&

$P_{Bel} = \{\mu : \mu(U) \geq Bel(U) \ \forall \ U \subseteq W\}$

Shafer gave a some different interpretation of these ideas such as: belief functions are part of a theory of evidence.

$$W = \{+o, -o\}$$

$$m(\{+o\}) = 0.8; \quad m(\{+o, -o\}) = 0.2,$$

$$m(\{-o\}) = 0; \quad m(\phi) = 0$$

Mass function $m(U)$ ⇒ The extent to which the evidence supports 'U'.

A mass function on W is a function

$m : 2^W \rightarrow [0, 1]$ such that the following two conditions hold.

$$m(\phi) = 0$$

$$\Sigma_{U \subseteq W} m(U) = 1$$

Belief / Plausibility Function based on m:

Let m be a mass function on W Then for every $U \subseteq W$.

$$Bel(U) =_{def} \Sigma_{U' \subseteq U} m(U')$$

$$Pl(U) =_{def} \Sigma_{U' \cap U \neq \phi} m(U').$$

Bel and Pl are dual.

If Bel is a belief function on W, then there is a unique mass function m over Ω Such that Bel is the belief function based on m. This mass function is given by the following equation.

$$\forall \; U \subseteq W, \; m(U) = \sum_{U' \subseteq U} (-1)^{|U \setminus U'|} \, Bel(U')$$

Dempster-Shafer Theory as a theory of evidence has to account for the combination of different Sources of evidence. This rule is an intuitive axiom that can best be seen as a heuristic rule rather than a well grounded axiom.

Fuzzy Reasoning:

The characteristic function for fuzzy sets provides a direct linkage to fuzzy logic. The degree of membership of x in $\bar{A}$ corresponds to the truth value of the statement x is a member of $\bar{A}$ where $\bar{A}$ defines some propositional (or predicate class.

$u_A(x) = 1$ the proposition A is True

$u_A(x) = 0$ " is false.

For Example let $\tilde{A}, \tilde{A}_1, \tilde{B}$ and $\tilde{B}_1$ be statements Characterized by fuzzy sets. Then one form of the generalized modus ponens reads

Premise : x is $\tilde{A}_1$

Implication : If x is $\tilde{A}$ then y is $\tilde{B}$

Conclusion : y is $\tilde{B}_1$

Example:

Premise : This banana is very yellow

Implication : If a banana is yellow then the banana is ripe

Conclusion : This banana is very ripe.

Now let x and y be two universes and let $\tilde{A}$ and $\tilde{B}$ be fuzzy sets in x and $x \times y$.

Fuzzy Relations — $\overline{R_A}(x)$, $\overline{R_B}(x,y)$, $\overline{R_C}(y)$

Fuzzy sets — x, $x \times y$, y.

$$\overline{R_C}(y) = \overline{R_A}(x) \circ \overline{R_B}(x,y)$$

composition of $\bar{A}, \bar{B}$

$$= \max_x \min\{u_A(x), u_B(x,y)\}.$$

Let

$\bar{R}$ = approximately equal, a fuzzy relation defined by

$$\bar{R}_x : \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 1 & 0.5 & 0 & 0 \\ 2 & 0.5 & 1 & 0.5 & 0 \\ 3 & 0 & 0.5 & 1 & 0.5 \\ 4 & 0 & 0 & 0.5 & 1 \end{array}$$

$$\bar{R}_c(y) = \max_x \min \{ u_A(x), u_R(x,y) \}$$

$$= \max_x \{ \min[(1,1), (.6,.5), (.2,0), (0,0)], \min[(1,0.5), (0.6,1), (0.2,.5), (0,0)], \min[(1,0), (.6,.5), (.2,1), (0,.5)], \min[(1,0), (0.6,0), (.2,.5), (0,1)] \}$$

$$= \max_x \{ [1,.5,0,0], [.5,.6,.2,0], [0,0.5,0.2,0], [0,0,0.2,0] \}$$

$$= \{ [1], [0.6], [0.5], [.2] \}$$

$\therefore$ The solution is $\bar{R}_c(y) = \{(1|1), (2|.6), (3|.5), (4|.2)\}$.

Stated in terms of a fuzzy modus ponen.

We might interpret this as the inference

Premise : x is little.

Implication : x and y are approximately equal

conclusion : y is more (or) less little.

Certainty Factors

Certainty Factors express belief in an event based on evidence.

$\Rightarrow$ 1.0 (or) 100 $\Rightarrow$ Absolute Truth.

$\Rightarrow$ 0 $\Rightarrow$ certain falsehood.

combining several certainty factors in one rule where parts are combined using AND and OR logical operators.

AND:

IF inflation is high, CF = 50 percent, (A)

AND unemployment rate is above 7, CF = 70 percent (B),

AND bondprices decline, CF = 100 percent, (C)

THEN stock prices decline

CF(A, B, C) = Minimum [CF(A), CF(B), CF(C)].

OR

IF inflation is low, CF = 70 percent, (A), (OR)

bond prices are high, CF = 85 percent, (B)

THEN Stock prices will be high

$CF(A, B) := \text{Maximum}[CF(A), CF(B)]$

$\Rightarrow$

The CF for "Stock prices to be high" = 85 Percent

(In OR only one IF premise needs to be true).

Example :-

Rules R_1 : If blood test result is yes
then the disease is malaria
(CF 0.8)

R_2 : If living in malaria zone
then the disease is malaria (CF 0.5)

R_3 : If bit by a flying bug
Then the disease is malaria (CF 0.3)

Questions :

What is the CF for having malaria, if

① The first two rules are considered to be true

② All the three rules are considered to be true.

Answer ①

$CF(R_1, R_2) = CF(R_1) + CF(R_2) * (1 - CF(R_1))$

Answer ②

$$CF(R_1, R_2, R_3) = CF(R_1, R_2) + CF(R_3) * [1 - CF(R_1, R_2)]$$

$$= 0.9 + 0.3 * (1 - 0.9) = 0.93.$$