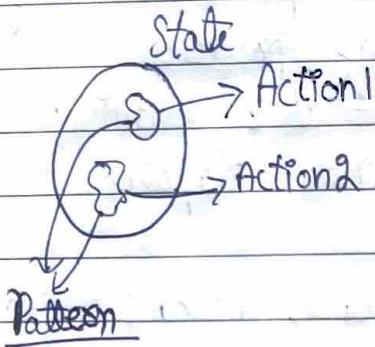


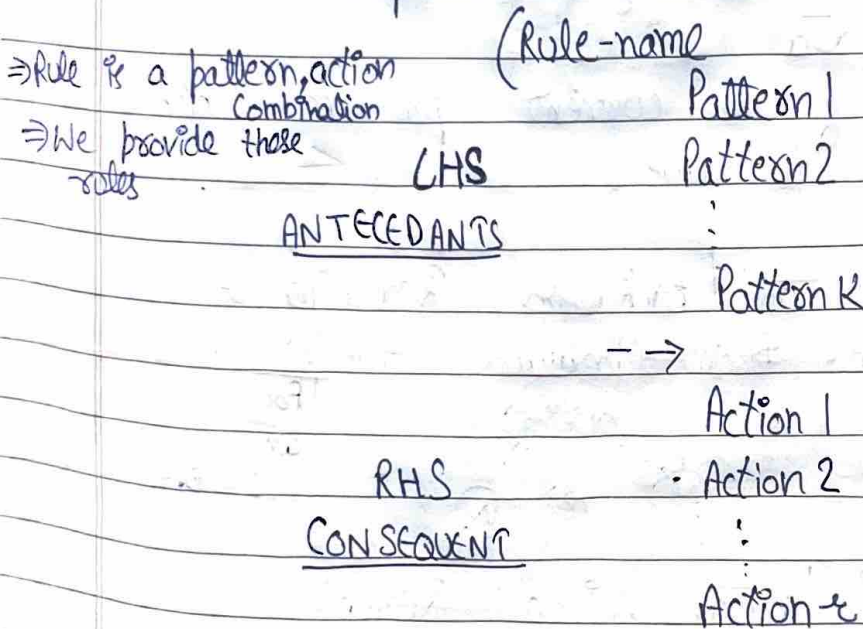
Dendral
XCON

Page No.	
Date	

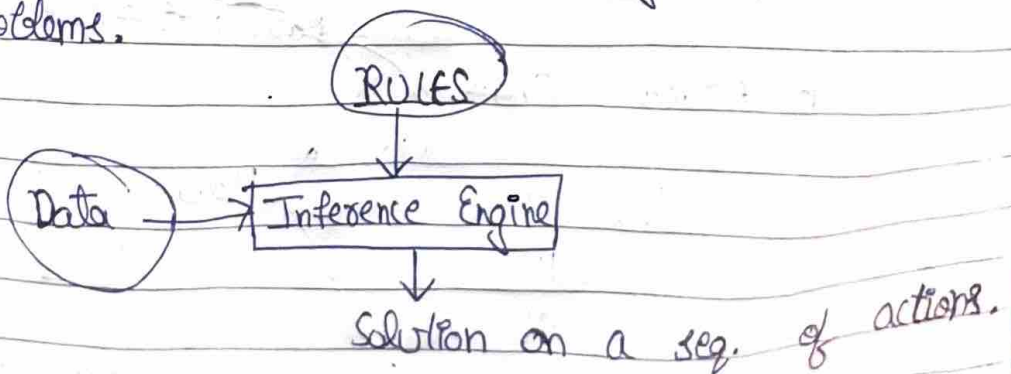
Rule based Systems / Expert Systems / Production Systems



Pattern is described in the form of a prod. or a rule



Idea is to capture knowledge of human in this form & use this knowledge to solve problems.



data is in working memory (WM)

⇒ Idea of rule based system is that problem solver or Domain Expert will only provide rules for solving problems

↳ Rules are like if some conditions are true then some action happens.

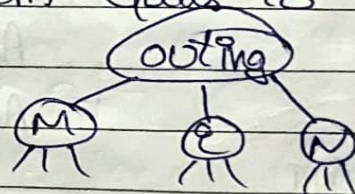
↳ So, only rules are given, inference engine does the work for them.

↳ This inference engine decides which rules to apply ↓ & then generate solⁿ.
on data

↳ Data is of the current problem which we are trying to solve.

↳ Goal Directed / Backward Reasoning :-
Reason from Goals to subgoals

like



For achieving Goals what sub-goals to achieve.

How to reason?

↳ Forward / Data Driven Reasoning

Algo that looks at data & says if it sees a certain pattern it will do this action & all

Backward Chaining

Vs

Forward Chaining

Match RHS → move towards LHS

How programs are implemented?

Match LHS then move towards RHS.

Rule based Programming Language.
(for building expert system).

Page No.	
Date	

⇒ Patterns in OPS5 language :-

⇒ (Class name attribute 1 -- attribute K)

eg ⇒ for Cards program

Class Description :- (Card name suit player rank)

These are WM elements.

~~(Card name suit player rank)~~

Pattern :- (Card)

Data is written like this

→ (Card ^name ace ^suit spades)

→ (Card ^name <x> ^rank >1)

→ (Card ^suit spades ^player <x>)

<-> → means variable.
It will match to anything.

of rule
→ LHS is made up of collection of such patterns

→ interpreted as
there is no such element
in WM such that player
x has a card of spade
(Hence negation)

ACTION :-WM element

① (Make (Card ^suit spades ---))

This action creates a WM element & puts it into WM
 i.e. it creates an element & put it into database

Remove action
 takes an
 argument

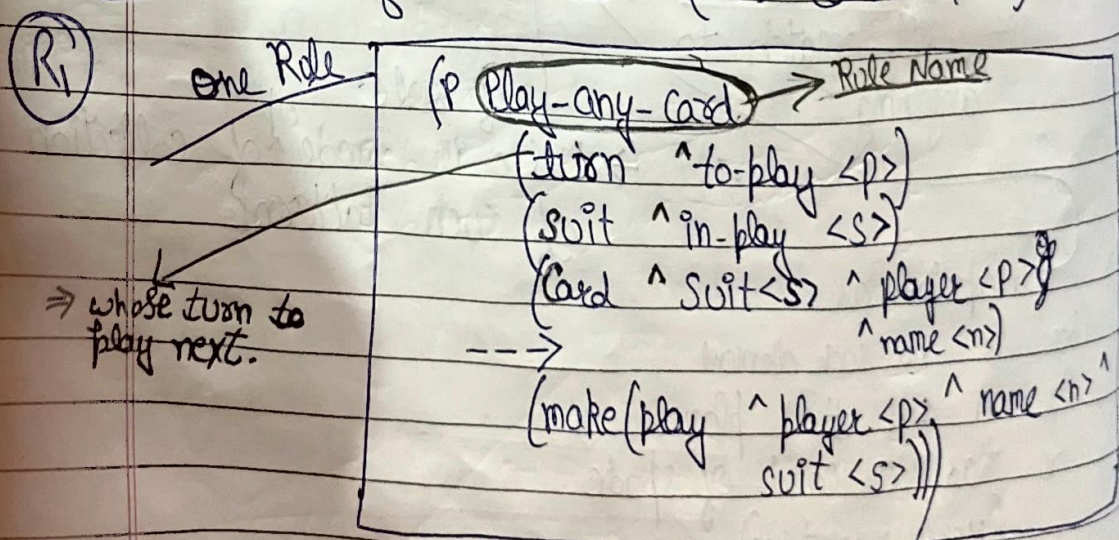
② (Remove 2)

means remove the data element
 which matches 2nd pattern.

Make + Remove \rightarrow Modify \rightarrow Update a WM element
 \Rightarrow other actions \Rightarrow Read, Write, Halt

\hookrightarrow There will be many diff classes in system.

\Rightarrow Examples of Rules: (for Card Games)



Say, turn, suit, card are names of class

Page No. Date Rule to play Highest Card

(R₉)

(P play-highest

(turn ^to-play <p>)

(suit ^in-play <s>)

(card ^player <p> ^suit <s> ^name <n> ^rank <r>)

-(card ^player <p> ^suit <s> ^rank <r>))

-->

(make(play ^player <p> ^name <n> ^suit <s>))

(P

(array ^index <i> ^val <n>)

(array ^index <j> > <i>) ^val <m> > <n>)

-->

(modify 1 ^val <m>
modify 2 ^val <n>))

Swapping 2 elements if this condⁿ is satisfied

This rule actually sorts the array

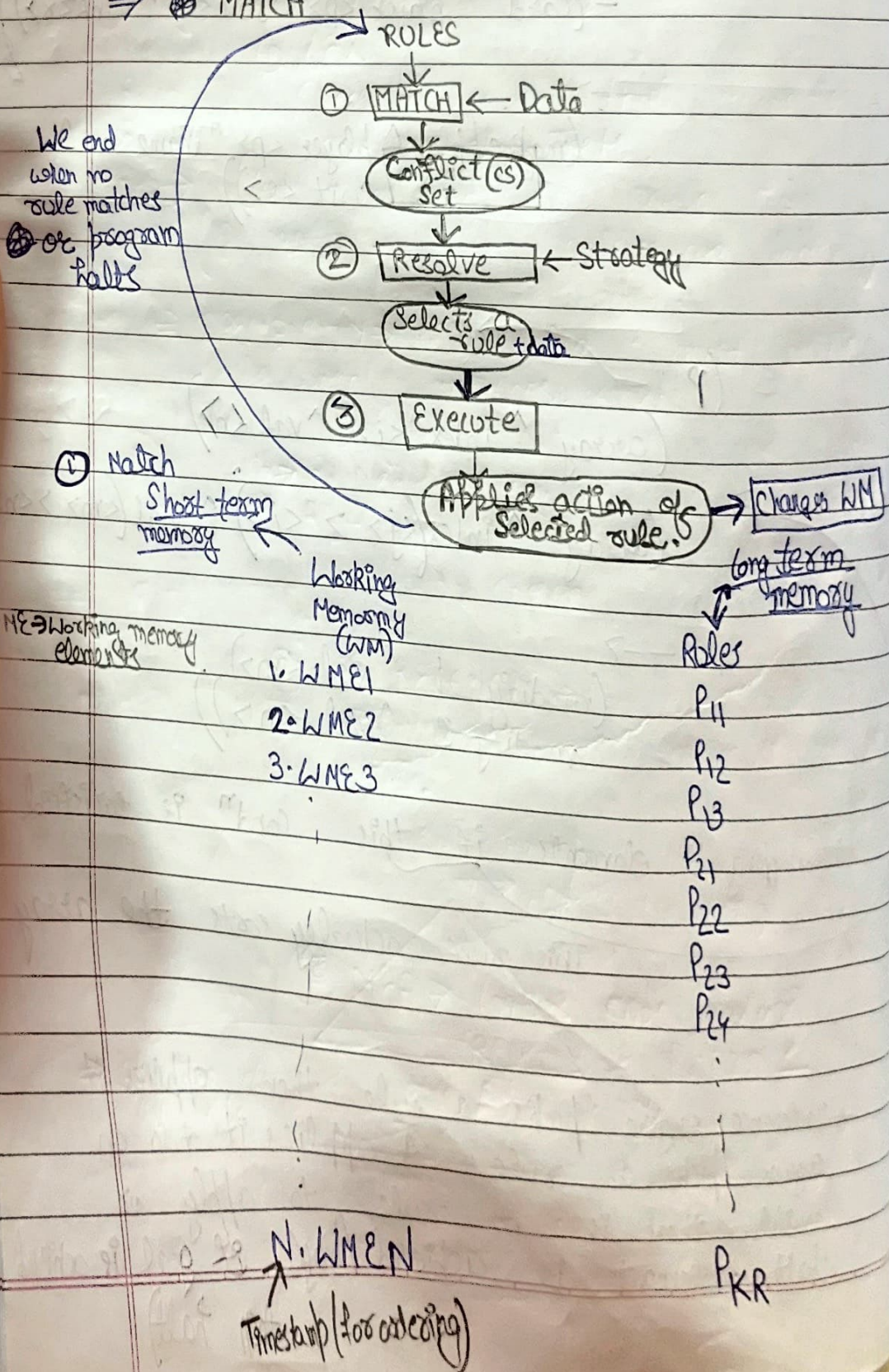
→ Inference engine picks a rule then applies it again picks a rule & applies it + so on until there is no rule to apply or goal is achieved (maybe then halt)

#

Inference Engine

⇒ Works in 3 phases

⇒ MATCH



We need to match ~~rules~~ rules & LME

Brute force way

Pick up P_{11} check with all
then P_{12} ————— & so on

⇒ Say, we have ~~sets~~ elements in WM as

① ② (turn ^ to-play s)

② ⑥ (suit & inplay club)

(3) (Card ^ player S ^ soft clubs ^ name 3)

⑨ ① Coord _____ 7)

⑤ ⑥ ⑦ _____ ⑧

⑥ ⑦ ⑧ ⑨

⑦ ⑤ (————— J

⑧ ⑨ (—————) ⑩

oo Rule R_1 (from prev pages) matches to all 6 instances.

1st instance (a) (b) (c)

2nd. — a b d

3rd — a b e

MATCH goal is to compute set of instances of rules & corresponding matching data & put it into a set called as conflict set.

Here, C.S. would have something like \rightarrow

$$C.S. = \left\{ \begin{pmatrix} R_1 & 1 & 2 & 3 \\ R_1 & 1 & 2 & 6 \end{pmatrix} \begin{pmatrix} R_1 & 1 & 2 & 4 \\ R_2 & & & \end{pmatrix} \begin{pmatrix} R_1 & 1 & 2 & 5 \\ R_2 & & & \end{pmatrix} \begin{pmatrix} R_3 & \dots \end{pmatrix} \right\}$$

② Resolve

It looks at C.S and picks one of them

This resolve needs some Strategy (Problem Solving Strategy).

eg → One Strategy is of

① LEX

Choose a rule that makes the max no. of Tests.

(tests means in each rule how many individual matches are happening)

⇒ SPECIFICITY → Choose the most specific rule which matches.

eg → R_1 says any of (a) (b) ... (h) instance
 R_2 says choose (h) ∴ (a → rank > high)
 ∴ R_2 chosen

② ~~RE~~ RECENCY → Choose a rule that matches the most recent data.

i.e. Timestamp
 (highest timestamp)

③ MEA → (means ends analysis)
 Recency of Pattern 1
 else Specificity

⇒ 1st bit of every rule if conflict then →

④ REFRACTORINESS

Same rule cannot fire with same data again.

⇒ Complexity →

- ↳ Execute & Resolve ~~is~~ ~~was~~ got less work
- ↳ Match has got very much work.
- ↳ Very time consuming (80-90)
 total time

⇒ Rete algorithm ⇒ Goal is to improve the efficiency of MATCH.

Current inefficiency of MATCH →

- ① In R_1 & R_2 pattern P_{11} & P_{21} are same but we check them separately.
 but if $R_1 \rightarrow P_{11}$ matches to some WME then $R_1 \rightarrow P_{21}$ should also match

∴ We want to share the tests which diff rules are doing.

So that each test we make only once.

- ② Execute → makes, deletes, modifies WMEs
 ∴ Some elements added, few removed but most same.
 ∴ Why do matching of all.

Changes in WM

⇒ In Rete Algorithm

Changes in C.S

Much less work

R_1, R_2 } previous pages
 $R_3 \rightarrow$

(p ruff

If current suit
not present then
play Trump suit
Card

① ~~suit~~ (suit ^ played < s >)

② ~~turn~~ (turn ^ < p >)

③ (trump ^ suit < t >)

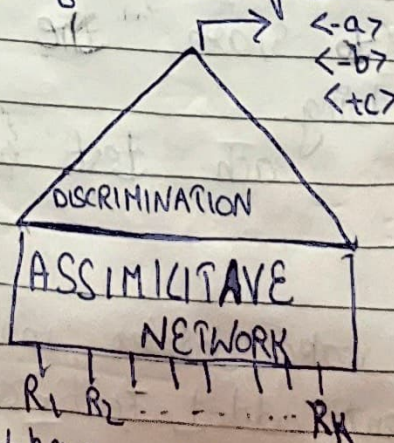
④ (card ^ suit < s > ^ name < r > ^ played < p >)

⑤ - (card ^ suit(s) ^ played < p > ~~played < s >~~)

Rete Algo →

→ Rete Net → Structure maintained by Rete algorithm.

↳ made up of 2 layers



+ve & -ve tokens are generated (Make, delete)
these tokens are inserted in Rete network.
These tokens are changes in working memory.

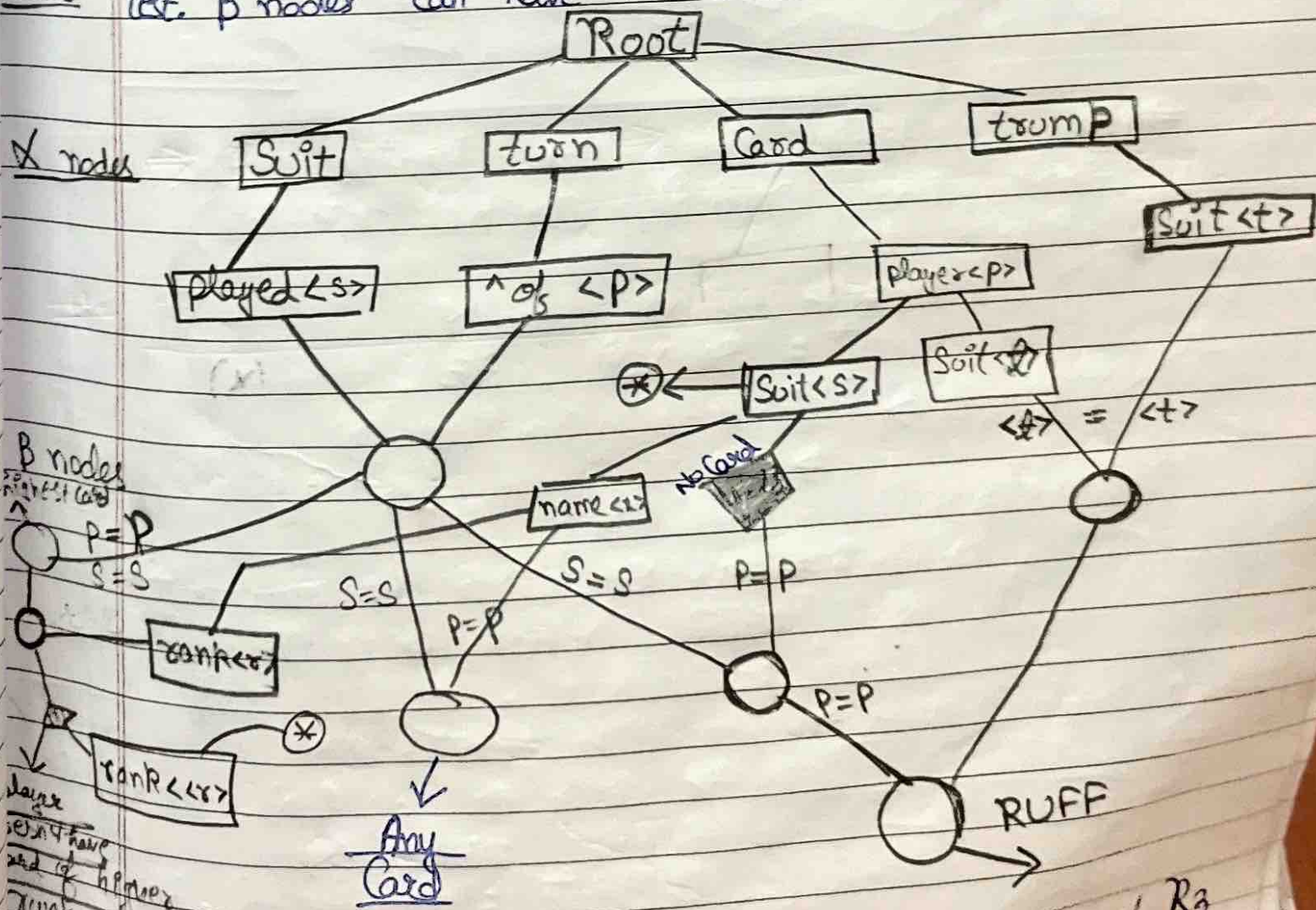
##

~~Planning~~

⇒ Input is working memory data, they move through a network & goes to rules with which they are concerned.

⇒ Top part is discriminative, it tries to separate tokens of diff kind.

test of class ⇒ X nodes have exactly one parent & they do a certain test. B nodes can have multiple parents. (we join 2 only).



⇒ Made from perspective of R3
one box 4 one B node

In some cycle matching of rule ① & rule ③ is happening.

Rete network is compilation of rules

oo If we have rete net, we don't need rules anymore.

oo We need to construct rete network