Partial order planning:-

works on subplanes
solves with subplans
combines with subplans

Putting on a pair of shoes:

Goal(RightShoeOn ∧ LeftShoeOn)
Init()

Action: Rightshoe
 Precond: Rightsockon
 effect: Rightshoe on

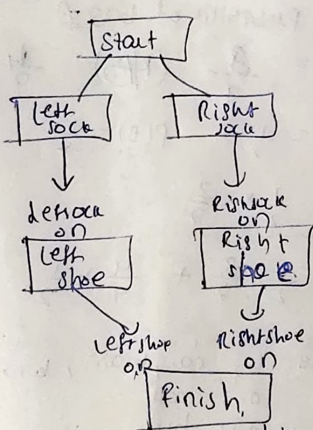
Action: Left shoe
 Precond: Leftsockon
 effect: Left shoeon

Action: Rightsock
 Precond: None
 effect: Rightsockon

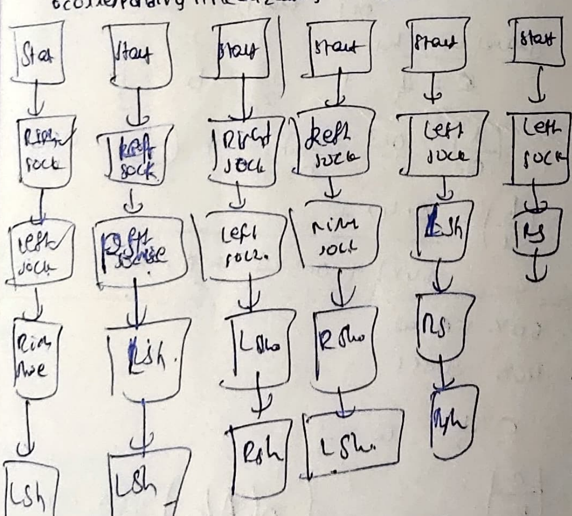
Action: Leftsock
 Precond: None
 effect: leftsockon

decomposition of Goalstate
↓
subgoals
↓
subplans
↓
combining subgoals
↓
final

Partial order plan:-



6 corresponding linearizations into top

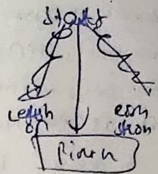


POP
→ Set of actions that make steps of plan
→ ordering constraints A<B (A before B)
→ casual link $A \xrightarrow{P} B$ (A achieves P for any B)
→ set of open precondition (set of precondition open, if not achieved by some action in plan)

Initial plan:-

Start:-
 Precond: None
 effect: add all propositions that are true

Finish:
 Precond: Goal state
 effect: None



ordering contrains: (start < finish)
casual links: { }
open precondition { precondition of finish }

Successor function:
Arbitrary chooses 1 open precondition P for an action B
make a successor plan by choosing 'a' that achieves P

Consistency:-
casual link: $A \xrightarrow{P} B$ and ordering constraint (A<B, S<A, A<fin)
resolve conflict: B<C C<A

Goalstate:
There are no open preconditions.

Final Plan

Actions: {Rights, RS, LS, Lsh}
orderings: {RS< Rsh, LS < Lsh}
open preco: { }
Links:
 $RS \xrightarrow{RSO} Rsh$
 $LS \xrightarrow{LSO} LSh$
 $Rsh \xrightarrow{RSho} Finish$
 $Lsh \xrightarrow{LSho} Finish$

Properties:-
Sound: guaranteed to achieve all goals
complete: If it possible to find plan, it will return plan
planners can work forward with postcondition,
 backward with precondition.

Flat tire:-

Goal(have good spare tire, mounted on car Axle)
Init()

Actions:-
 remove spare from trunk
 remove flat tire from axle
 putting spare on axle and
 leaving overnight unattended.

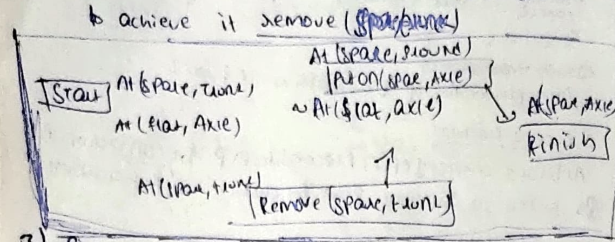
→ Init(At(Flat,Axle) ∧ At(Spare, Trunk))
→ Goal(At(Spare, Axle))
→ Action(Remove(Spare, Trunk),
 Precond: At(Spare, Trunk)
 effect: ~At(Spare, Trunk) ∧ At(Spare, Ground)
→ Action(Remove(Flat, Axle),
 Precond: At(Flat, Axle)
 effect: ~At(Flat, Axle) ∧ At(Flat, Ground))
→ Action(Puton(Spare, Axle),
 Precond: At(Spare, Ground) ∧ ~At(Flat, Axle)
 effect: ~At(Spare, Ground) ∧ At(Spare, Axle))
→ Action(Leave overnight,
 Precond:
effect: ~At(Spare, trunk) ∧ ~At(Spare, Ground) ∧ ~At(Spare, Axle)
 ∧ ~At(Flat, Ground) ∧ ~At(Flat,

At(Spare, Trunk) ∧ At(flat, tyre)
Find:
Goal with Precondi At(Spare, Axle)

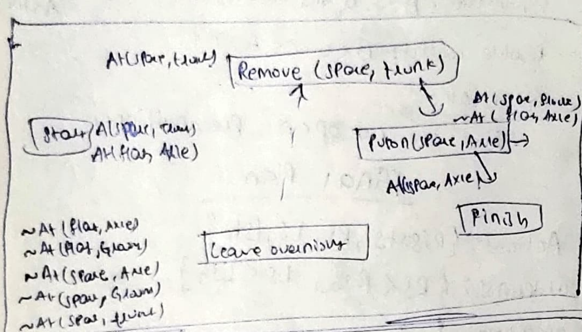
Sequence of action:-

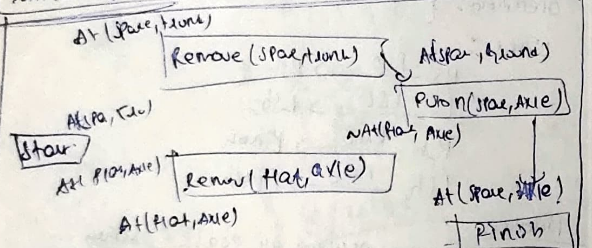
1) Pick only open precondition At(Spare, Axle) of finish
Choose only applicable action Puton(Spare, Axle)

2) Pick at(Spare, Ground) Precondition of Puton(Spare, Axle)
to achieve it remove(~~Spare, Trunk~~)



3) Pick At(flat, Axle) precondition of Puton(Spare, Axle)





Bayes theorem:

$$P(A|B) = \frac{P(A \cap B)}{P(B)} \qquad P(B|A) = \frac{P(B \cap A)}{P(A)}$$

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$

① Probability of 2 girls given atleast 1 girl

$$(2G/1G) = \frac{P(1G|2G) \cdot P(2G)}{P(1G)}$$

$$= \frac{1 \times \frac{1}{4}}{\frac{3}{4}} = \frac{1}{3}$$

② Insurance company insured 2000 scooters, 4000 car driving 6000 truck drivers. Probability of accident involving scooter - 0.01, car - 0.03, truck - 0.15. Probability of scooty driver accident

$P(e1) = \frac{2}{12} = 1/6$ | $P(A/e1) = 0.01$
$P(e_2) = 4/12 = 1/3$ | $P(A|e_2) = 0.03$
$P(e_3) = 6/12 = 1/2$ | $P(A|e_3) = 0.15$

$$P(e1/A) = \frac{P(e_1) \cdot P(A|e1)}{P(e1) \cdot P(A|e1) + P(e) \cdot P(A/e_2) + P(e_3) P(A/e_3)}$$

$$= \frac{\frac{1}{6} \times 0.001}{\frac{1}{6} \times 0.001 + \frac{1}{3} \times 0.03 + \frac{1}{2} \times 0.15}$$

$$= \frac{1}{6+1+45} = \frac{1}{52}$$

③ man is known to speak 3/4 times. He says on a throw "it is a six". Proba six actually

E_1 = occurence of 6
E_2 = not 6

$E_1 = \left(\frac{1}{6}\right) \quad E_2 = \left(\frac{5}{6}\right)$

$P\left(\frac{T}{e_1}\right) = \frac{3}{4}, \quad P\left(\frac{T}{e_2}\right) = \frac{1}{4}$

$$P\left(\frac{e_1}{T}\right) = \frac{\frac{1}{6} \times \frac{3}{4}}{\frac{1}{6} \times \frac{3}{4} + \frac{5}{6} \times \frac{1}{4}} = \frac{3}{8}$$

④ Bag A 3 red, 4 green. Bag B 4 red, 5 green

found red. Probability of bag A

$(R/A) = \frac{3}{7} \quad (R/B) = \frac{4}{9}$

$P(A) = \frac{1}{2} \quad P(B) = \frac{1}{2}$

$$P\left(\frac{A}{R}\right) = \frac{\frac{1}{2} \times \frac{3}{7}}{\frac{1}{2} \times \frac{3}{7} + \frac{4}{9} \times \frac{1}{2}}$$

moving office by car, train, bus, getting late
0.3 car
0.4 bus
0.1 train

travel using bus

$c = \frac{1}{3} \quad t = \frac{1}{3} \quad b = \frac{1}{3}$

$\left(\frac{l}{c}\right) = 0.3 \quad \left(\frac{l}{b}\right) = 0.4 \quad \left(\frac{l}{t}\right) = 0.1$

$$\left(\frac{b}{l}\right) = \frac{0.4 \times \frac{1}{3}}{0.4 \times \frac{1}{3} + 0.3 \times \frac{1}{3} + 0.1 \times \frac{1}{3}}$$

60% knew
40% guess

$c \ \frac{1}{2} \quad w = \frac{1}{2}$

$\left(\frac{c}{k}\right) =$

$P(e_1) = 1/6$
$P(e_2) = 1/3$
$P(e_3) = 1/2$

$P(A/e_1) = 0.01$
$P(A/e_2) = 0.03$
$P(A/e_3) = \cancel{0.15}$

$$P(e_1|A) = \frac{P(A/e_1) P(e_1)}{P(A)}$$

$a_{ij} = P(S_i | S_j)$ S_j = current state

$\Pi_i = P(S_i)$. S_i = Previous state

Starting state prob — Initial probabi

① To predict weather pattern for next

given 7 days

S_3	S_3	S_1	S_1	S_3	S_2	S_3
sun	sun	Rain	Rain	sun	cloudy	sun

- Today is sunny

Transitional matrix

	R	C	S
R	0.4	0.3	0.3
C	0.2	0.6	0.2
S	0.1	0.1	0.8

S_1 = Rain. S_2 = Cloudy S_3 = Sunny

Transitional Probabib

$P(S_3) \cdot P(S_3|S_3) \cdot P(S_3|S_3) \cdot (P(S_1|S_3) \cdot$
$P(S_1|S_1) \cdot P(S_3|S_1) \cdot P(S_2|S_3) \cdot P(S_3|S_2)$

$\Pi_3 \cdot a_{33} \cdot a_{33} \cdot a_{31} \cdot a_{11} \cdot a_{13} \cdot a_{32} \cdot a_{23}$.

$= 1 \cdot 0.8 \cdot 0.8 \cdot 0.1 \cdot 0.4 \times 0.3 \times 0.1 \times 0.2$

$= 1.536 \times 10^{-4}$

0.000536

0.0001536

	R	D
R	0.2	0.65
D	0.3	0.7

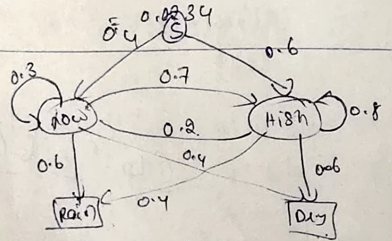
Calculate seq {Dry, Rain, Rain, Dry}

Π [0.4] Rain, [0.6] Dry

P{Dry} . P{D|Dry} . P{R|R} . P(D|R)

= 0.6 × 0.65 × 0.2 × 0.3

0.0234



Hidden state = low, high

observed visible states = Rain & Dry

$\Pi_i = \begin{bmatrix} 0.4 \\ 0.6 \end{bmatrix}$ low, high

P(Dry, Rain) = P{(Dry, Rain), (low, low)} + P{(Dry, Rain), (low, high)} + P{(Dry, Rain) (high, low)} + P(Dry, Rain) (high, high)

1) P(Dry | Rain) + P(low | low) =

P(Dry | low) . P(rain | low) . P(low) . P(low | low)

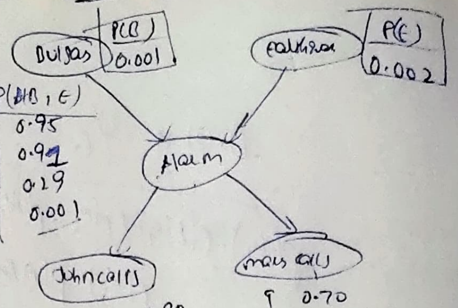
(0.4) (0.6) . (0.4) . (0.3) = 0.288

2) P(Dry | Rain) . P(low | high)

nodes → hypothesis

arc → dependencies

Bayesian Belief Network :-



B	E	P(A | B, E)
T	T	0.95
T	F	0.94
F	T	0.29
F	F	0.001

A	P(J)
T	0.90
F	0.05

A	P(M)
T	0.70
F	0.01

Prob that alarm has sounded but neither burglary nor earthquake occurred and both mary and John calls

$P(j \wedge m \wedge a \wedge \sim b \wedge \sim e)$

$= P(j|a)\, P(m|a)\, P(a|\sim b \sim e)\, P(\sim b)\, P(\sim e)$

$= 0.9 \times 0.7 \times 0.001 \times 0.999 \times 0.998$

$= 0.00062$

① Prob that john call

$P(J) = P(J|a) \cdot P(a) + P(J|\sim a) P(\sim a)$

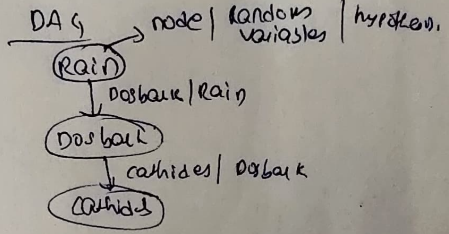
$= P(a) = P(a|b,e) * P(b,e) + P(a|\sim b e) * P(\sim b e)$
$+ P(a|b \sim e) * P(b \sim e) + P(a|\sim b \sim e) * P(\sim b \sim e)$

$= 0.90 \times 0.00252 + 0.05 * 0.9974 = 0.0521$

DAG (Directed Acylic Graph)

CPT (conditional probabi. Tables)

DAG → node | random variables | hypothesis.



Condition Prob

	R	~R
D	9/48	18/48
~D	3/48	18/48

→ It is Probablis Graphical method that represent conditional dependencies b/w random variables through DAG

→ also it is suitable for probabilistic relationship between multiple events.

→ BBNs represent the Joint probabi distribu

Joint Probabi = P(w, h, u) =

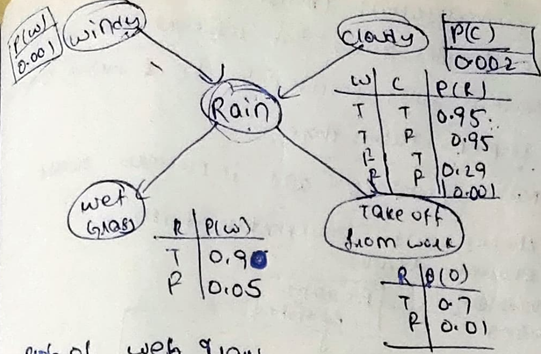
$P(w) * P(h|w) * P(u|w)$

$P(x_1, x_2 \ldots) = \prod_{i=1}^{n} P(x_i \mid \underbrace{Parent(x_i)}_{\text{denotes belief}})$

→ Probabilities makes inference and DTU are bodies

→ each node of DAG associated with CPT

→ Acyclic graph where all nodes of graph represent hypothesis & arc connecting two nodes represent dependency b/w them.



P(W)
0.001

P(C)
0.002

W	C	P(R)
T	T	0.95
T	F	0.95
F	T	0.29
F	F	0.001

R	P(W)
T	0.90
F	0.05

R	P(O)
T	0.7
F	0.01

Prob of wet grass:-

$P(W) = P(W|R) \times \boxed{P(R)} + P(W|\sim R) \times P(\sim R)$

$= 0.9 \times \quad\quad 0.05 \times$

$P(R) = P(R|W,C) * P(W \wedge C) +$
$P(R|\sim W, C) * P(\sim W \wedge C) +$
$P(R|\sim W, \sim C) * P(\sim W \wedge \sim C) +$
$P(R|W, \sim C) * P(W \wedge \sim C)$

$= 0.95 * 0.001 * 0.002 +$
$0.29 * 0.999 * 0.002 +$
$0.001 * 0.999 * 0.998 +$
$0.95 * 0.001 * 0.998.$

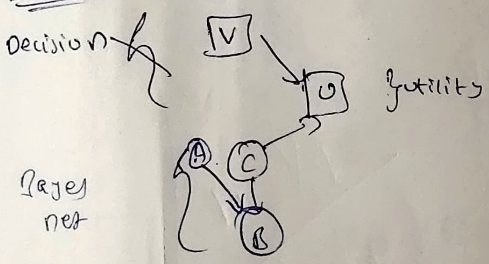
$P(R) = 0.0252$

characteristics:-

- Prior knowledge of domain, less prone to model over fitting
- can handle missing & incomplete data.

disadv construction of DAG is hard.

DECISION NETWORKS:-



Expected utility:

$E(U|V) = \Sigma_c P(c) U(+v, c)$

$EU(+v|+b) = \Sigma_c P(C|+b) U(+v, c)$

max expec utility:

$MEU(\phi) = \max_v EU(v)$

$MEU(+b) = \max_v EU(V|+b)$

$MEU(B) = \Sigma_b P(b) MEU(b)$

value of perfect information

$VPI(e'|e) = MEU(e, e') - MEU(e)$

VPI (A)

A independent C

$P(C|A) = P(C)$

$VPI(A) = 0$

Architecture of expert system

~~Components of ES:-~~ ES

- ES are computer applications developed to solve complex problems in a particular domain at the extraordinary human-intelligence and expertise

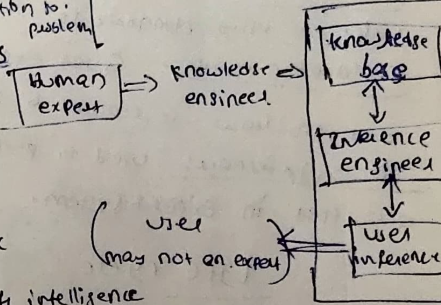
Characteristics

- high performance
- understandable
- Reliable
- highly responsive

capabilities	incapable
- Advising	- substituting human decision maker
- instructing and assisting human in decision making	- possessing human capabilities
- demonstrating	- producing accurate o/p for inadequate knowledge base
- deriving a solution	- Refining their own knowledge.
- explaining	
- interpreting i/p	
- predicting result	
- suggesting alternative option to problem	

components Mycin ES

- Knowledge base.
- Inference engine
- user interface



Knowledge Base:-

- contains domain-specific quality knowledge
- knowledge represents intelligence
- ~~Good~~ ES is defined which has high quality & accurate knowledge

Inference engine uses fwd/bwd chaining

- brain of es ② have rules to solve problems
③ apply rules and facts to solve problem
④ reasoning ~~to determine solution~~ abt info in KB
⑤ it helps in deducing problem to find solution
⑥ helpful for formulation conclusion

user interface:-

Crucial part of ES.

Takes i/p from user and passes it to inference engine

Display result to user.

user communicates with es using user interface

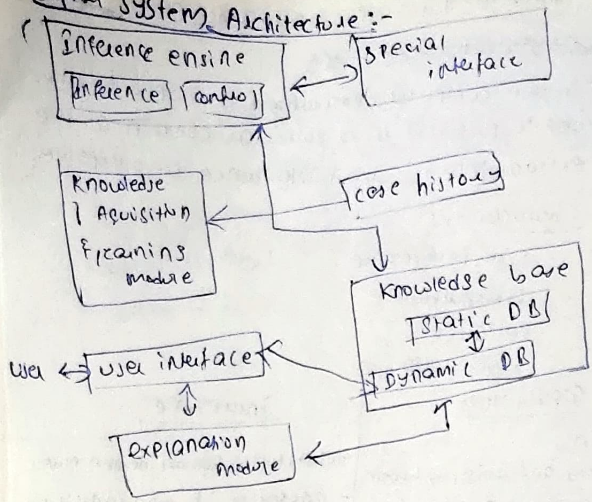
Process of building ES:-

KE, DE work on defining problem

→ KE translates K into comp understand lang and designs ES in a way he can use K when needed.

→ K expert determines knowledge and reasoning process and explains

Expert System Architecture :-



Knowledge Acquisition :- It allows system to acquire more knowledge regarding the problem

case history :- It stores files created by inference using dynamic DB

Explanation module :- Gives explanation to user on how it reached to such decision.

Special inference :- used to perform specialized activities in expert system.

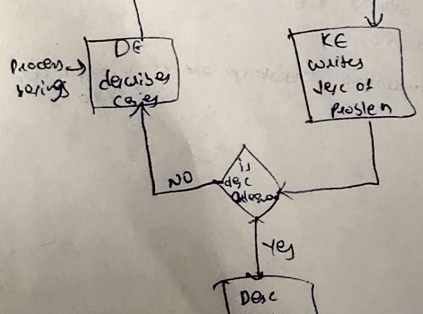
LIFE cycle

Stages

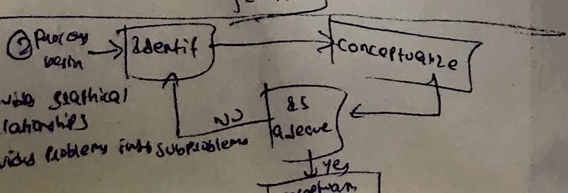
1) Prob identify :-
2) Decision about mode of development
3) Prototype develop
   - Knowledge acquisition
   - Know representation
4) Planning a full scale system
5) Final implementation.

1) Identification (determining charac of prob)
2) conceptualization (finding concept to produce solution)
3) Formalization (Design structure to organize knowledge)
4) Implementation (Prototype improvement & implement)
5) Testing (validating rules)

① knowledge engineer



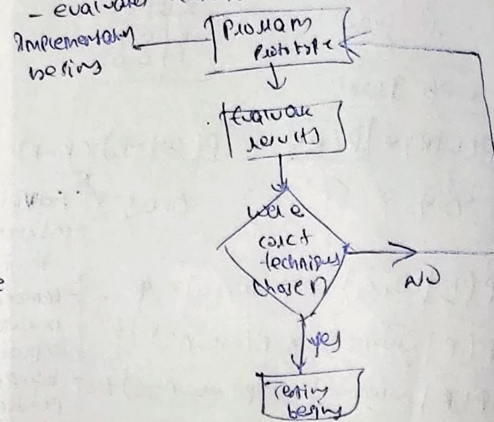
② Process begin



→ Provide graphical relationships
→ divides problems into subproblems

③ Formalisation :- (Design Phase)
- KE develops rules that represent ES
- review rules proposed by KE & suggest changes

④ Implementation Phase
- makes prototype and implements rules
- checks and reformulate design
- evaluates results



⑤ Testing
- Tests for weakness
- gets to know loopholes
- validates
- feedback