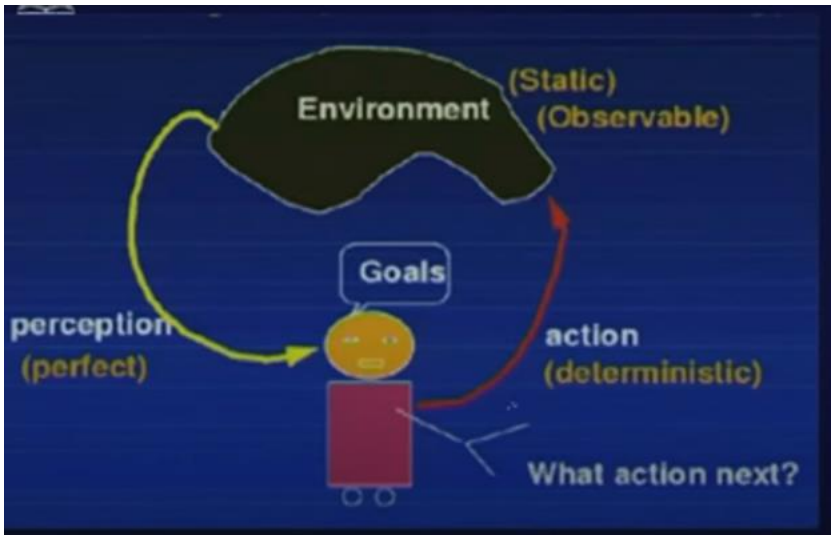
DAYANANDA SAGAR COLLEGE OF ENGINEERING Dept. of Information Science & Engg
(An Autonomous Institute Affiliated to VTU, Belagavi)
Shavige Malleshwara Hills, Kumaraswamy Layout, Bengaluru-560078
Dr. Rajeshwari.J
Associate Professor

Planning

5.1 Classical Planning:

-Synthesize goal directed behaviour

-Select action sequences(-handle casual dependencies)



What is planning in AI?

- The planning in Artificial Intelligence is about the decision making tasks performed by the robots or computer programs to achieve a specific goal.
- The execution of planning is about choosing a sequence of actions with a high likelihood to complete the specific task.

a) Planning problem: finding a sequence of actions that leads to a goal state starting from any of the initial states.

- Solution (obtained sequence of actions) is optimal if it minimizes sum of action costs.
- Search-based problem solving agents as a special case of planning agents

- Environment can be dynamic, nondeterministic, partially observable, continuous, multi-agent.
- Actions may take
    - time (have durations),
    - have continuous effects
    - be taken concurrently.

Planning as a form of general problem solving and it should be represented properly.

b)Representation of states:

-Planners decompose the agent world into logical conditions and represent a state as conjunction of positive literals**.**

- A state is represented more clear
  Example: States are represented by predicate such as at(x), have(x). Agent is at specific position x.

c)Representation of Goals:

A goal is a represented as conjunction of positive grounded literals such as Rich^famous, at(p2,Delhi).

The proposition state Rich^famous^happy satisfies the goal state Rich^famous

d)Action Representation

Specification of Actions:

1. Preconditions that must hold before the action can be executed.
2. Effects of executing the action.

---A set of actions can be represented by a single **action schema. It represent the number of different actions. A**n **action schema** is a *most general unifier (MGU)* of the set of grounded actions.

Action Schema consists of three parts:

i) The action name and parameter list

For example: fly(from to), fly is an action, from to are parameters.

ii)-Precondition-A conjunction of literals which must be true in a state before the an action can be represented.

iii)Effect: A conjunction of literals which indicates the state changes then the action is executed.

Action is applicable iff it satisfies the precondition otherwise there is no effect.

In Wumpus world example, simple action of moving a step forward had to be repeated for all four orientations, in response to this, planning researchers have settled on a **factored representation** (where states of the world is represented by a collection of variables).

e) Planning Domain Definition Language (PDDL)

allows us to express all 4 actions with one action schema

can describe the 4 components of a search problem:

1. initial state
2. actions available for each state
3. results of each action
4. goal test

Example: Air cargo transport

Actions in PDDL

Sample action:

• Action(Fly(P1, SFO, JFK),

Precond: At(P1, SFO) ∧ Plane(P1) ∧ Airport (SFO) ∧ Airport(JFK)

Effect:¬At(P1, SFO) ∧ At(P1, JFK))

• Can generalize into an action scheme

Sample action schema:

• Action(Fly(p, from, to),

Precond: At(p, from) ∧ Plane(p) ∧ Airport (from) ∧ Airport(to)

Effect:¬At(p, from) ∧ At(p, to))

- Variables are universally quantified
- Properties that change from situation to situation is called fluents.

Example: The blocks world

One of the most famous planning domains is known as the blocks world. This domain consists of a set of cube-shaped blocks sitting on a table.2 The blocks can be stacked, but only one block can fit directly on top of another. A robot arm can pick up a block and move it to another position, either on the table or on top of another block. The arm can pick up only one block at a time, so it cannot pick up a block that has another one on it. The goal will always be to build one or more stacks of blocks.

Init(On(A, Table) ∧ On(B, Table) ∧ On(C, A) ∧ Block (A) ∧ Block (B) ∧ Block (C) ∧ Clear (B) ∧ Clear (C))

Goal(On(A, B) ∧ On(B,C))

Action(Move(b, x, y),

PRECOND:On(b, x) ∧ Clear (b) ∧ Clear (y),

EFFECT:On(b, y) ∧ Clear (x) ∧ ¬On(b, x) ∧ ¬Clear (y)) .

Action(MoveToTable(b, x),

PRECOND: On(b, x) ∧ Clear (b) ∧ Block (b)

EFFECT: On(b, Table) ∧ Clear (x) ∧ ¬On(b, x))

We use On(b, x) to indicate that block b is on x, where x is either another block or the table. The action for moving block b from the top of x to the top of y will be Move(b, x, y). Now, one of the preconditions on moving b is that no other block be on it. In first-order logic, this would be ¬∃ x On(x, b) or, alternatively, ∀ x ¬On(x, b). Basic PDDL does not allow quantifiers, so instead we introduce a predicate Clear (x) that is true when nothing is on x.

Example: The spare tire problem: Do it as assignment.

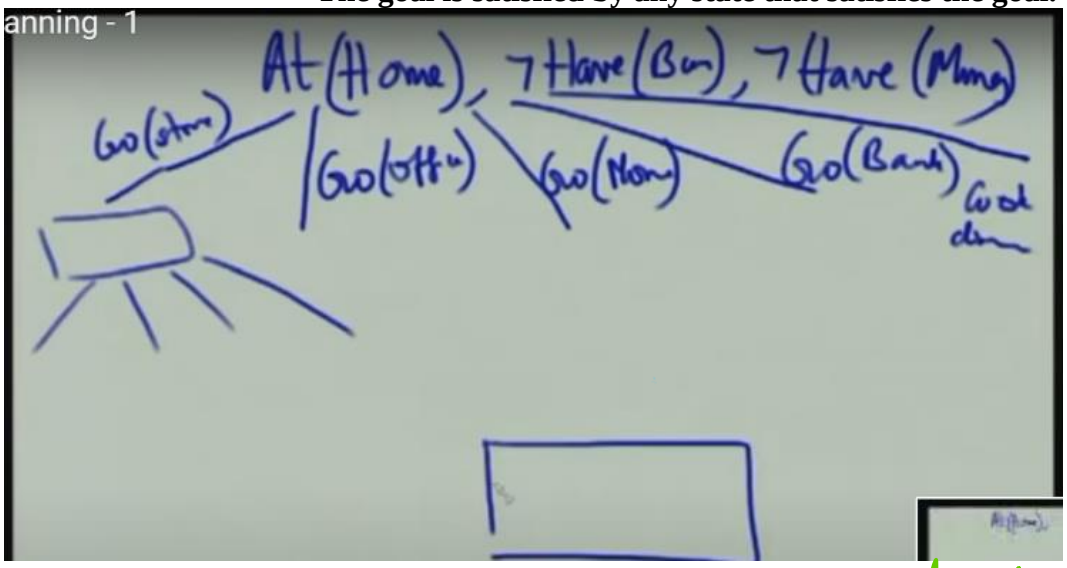
5.2 Algorithms for planning as state space search:

DAYANANDA SAGAR COLLEGE OF ENGINEERING
Dept. of Information Science & Engg
(An Autonomous Institute Affiliated to VTU, Belagavi)
Shavige Malleshwara Hills, Kumaraswamy Layout, Bengaluru-560078
Dr. Rajeshwari.J
Associate Professor

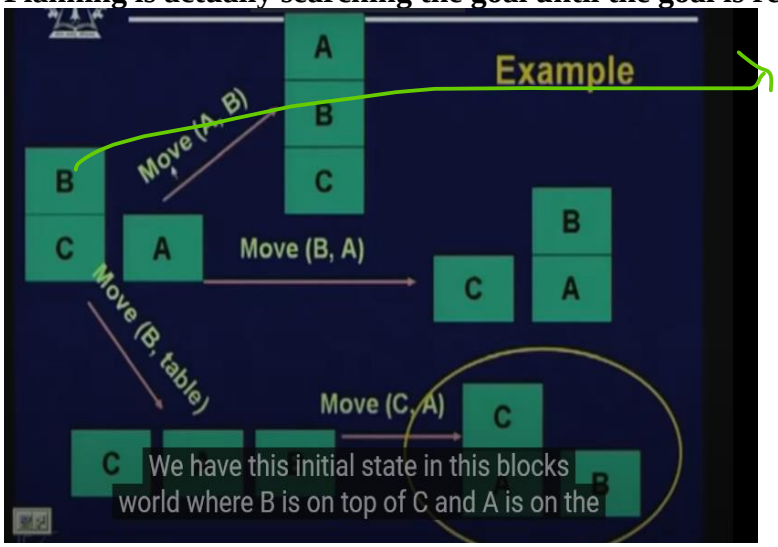
Planning problem is to determine a sequence of actions that when applied to initial state yields a state which satisfied the goal.

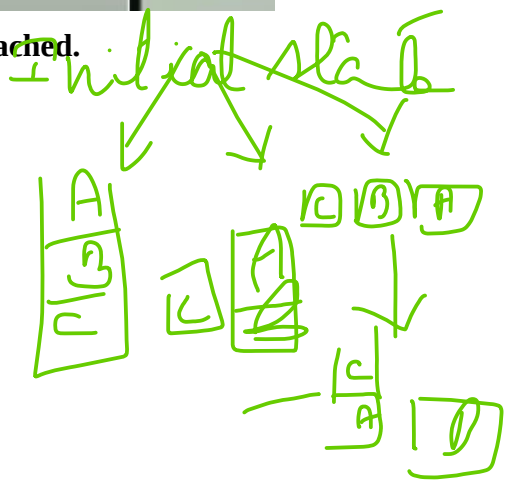
Planning as a search: This can be treated as search problem

- **The initial state is given**
- **The actions are operators mapping a state to a new state.**
- **The goal is satisfied by any state that satisfies the goal.**



Planning is actually searching the goal until the goal is reached.





Dr. Rajeshwari.J
Associate Professor

Search tree is generally quite large. The actions depend on each other via their preconditions. Planning algorithms are designed to take advantage of the special nature of the representation. It is possible to search both from the initial state or from the goal state. We have to pick expected actions which will lead to goal state.

a) Forward state space search: Search from what is known in the initial state and apply operators in order they are applied.
b) Backward state space search: Search from the description of the goal and identify actions that help to reach the gaol.

In forward search:

Problem formulation for progression:

Initial State:

Initial state of planning problem:

Action:

Applicable to the current state

First action precondition are satisfied , successor state are generated.

Add positive literals to add list and negative literals to delete list.

Goal test:

Whether the state satisfies the gaol of the planning

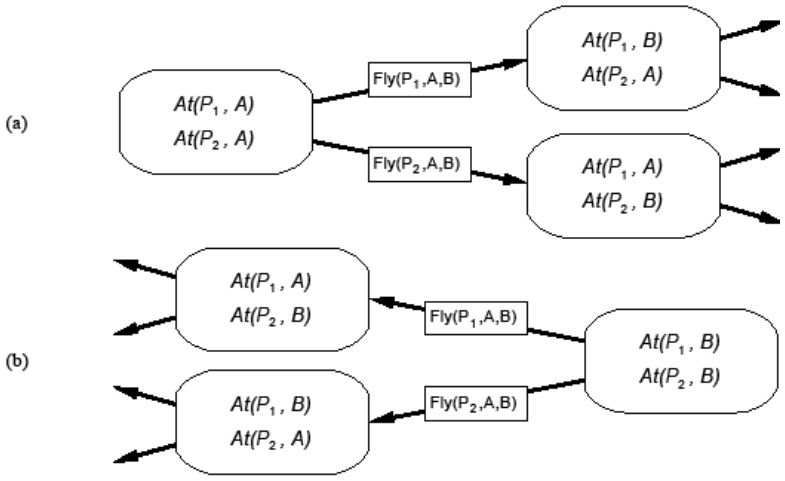
Step cost:

Each action is 1

Continuation:Progression:

- From initial state , search forward by selecting operators whose preconditions are satisfied. With literals are in the state.
- Search forward until goal is reached.

Dr. Rajeshwari.J
Associate Professor



- FSSS algorithm:

1)compute whether or not the state is goal state.

2) find the set of actions that are applicable for the state

3)Compute the successor state that is result of applying any action to a state.

4)Algorithm takes an input p=(o,s,g) of a planning problem p. (o contains a list of actions).

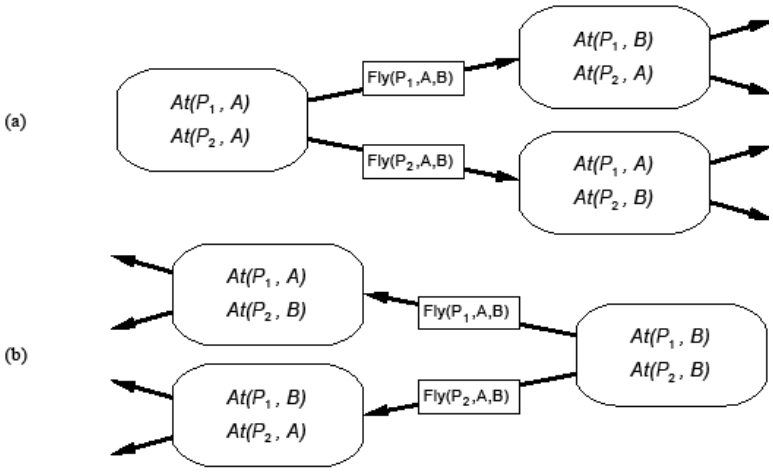
5) if P is solvable then algorithm returs a solution plan otherwise it returns a failure

Backward Search

Regression:

- The preconditions must hold in the previous situation and these become the subgoals which might be satisfied, by the initial conditions.
- Perform backward chaining from gaol.

Dr. Rajeshwari.J
Associate Professor



BSSS algorithm:put

- Start at the goal.
- Test the goal is initial state, otherwise,
- Apply inverse of the planning operators, to produce subgoals,
- The algorithm stops if we produce, a set of subgoals, that satisfies the initial state.

Heuristics for state space search:

How to find the admissible heuristic estimate:

- Distance from state to gaol.
- Look at the effects of the actions, and at the goals and guess how many actions are needed. Ie. We are not selecting all the actions, but selecting the one which leads to the goal state.
- The cost of solving the conjunction of is approximated by the sum of costs of solving each subgoal independently.

DAYANANDA SAGAR COLLEGE OF ENGINEERING
Dept. of Information Science & Engg
(An Autonomous Institute Affiliated to VTU, Belagavi)
Shavige Malleshwara Hills, Kumaraswamy Layout, Bengaluru-560078
Dr. Rajeshwari.J
Associate Professor

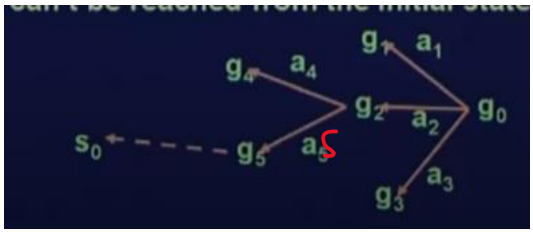
.

5.3 Planning Graph:

Motivation: A big source of inefficiency in search algorithms is the branching factor.

I,e number of childrens in each node.

Example: a backward search may try lots of actions that cant be reached from the initial state.



One way to reduce the branching factor is to use relaxing problem which removes some of the restrictions of the original problem.

Basic idea:

- -construct a graph that encodes constraints on possible plans
- -use this planning graph to constrain search for valid plan.
- -If planning graph exists then it is a subgraph of planning graph
- Convert the problem structure into planning graph called Graph plan.
- It gives the relation between the action and states, the precondition must be satisfy the action.
- The planning graph is layered graph, with alternate layers of propositions and action.
  - ✓ Layer P0,
  - ✓ Action A0
  - ✓ Layer P1
- Propositional problem will look at what is starting state, what objects in the domain, and it will produce all possible actions, and works with those actions.

Dr. Rajeshwari.J
Associate Professor

- We construct the planning graph from left to right,
- We keep inserting proposition and actions, and the propositions until we get the goal preposition appear on the preposition layer.
- There are two states in the planning graph problem
  - ✓ Construct the planning graph
  - ✓ Search for the solution.
- If you cannot get the solution, then extend the planning graph and search the solution that until you get the solution.
- Planning graph can give better heuristic.
- Here we can extract a solution directly from the planning graph, using specialized algorithm Graph plan.
- A planning graph consists of sequence of levels, where level 0 is the initial state.
- Each level contains set of literals and actions.
- Planning graph are the efficient way of the representation of planning graph, that can be used to
  - ✓ Achieve better heuristic
  - ✓ Directly construct plan
- Planning graph works only for proposition logic.

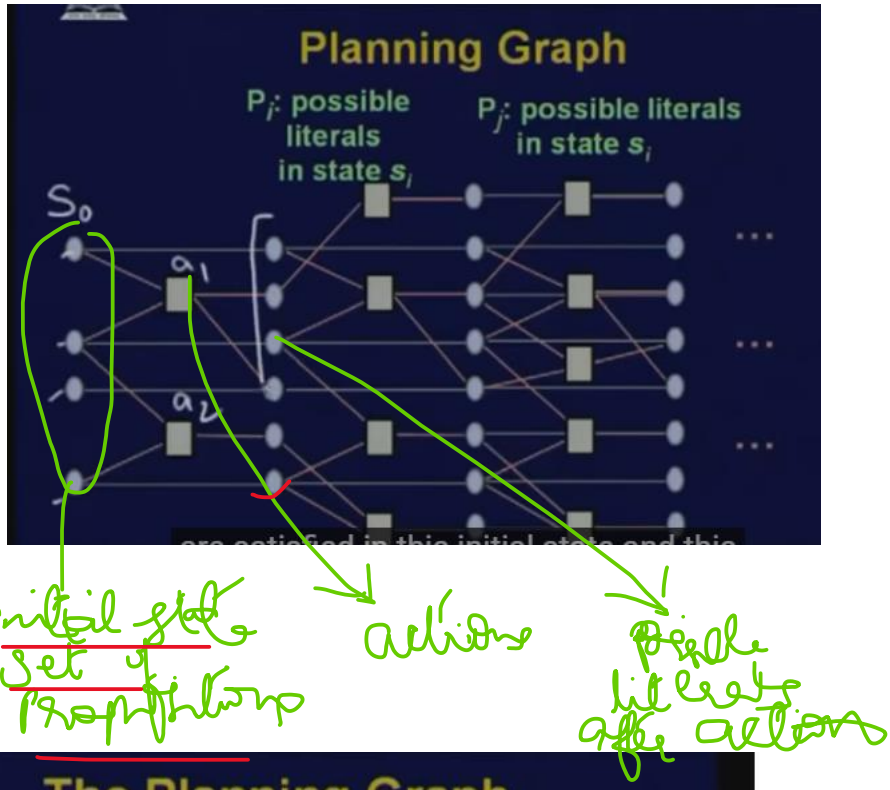
Planning graph has two types of nodes:

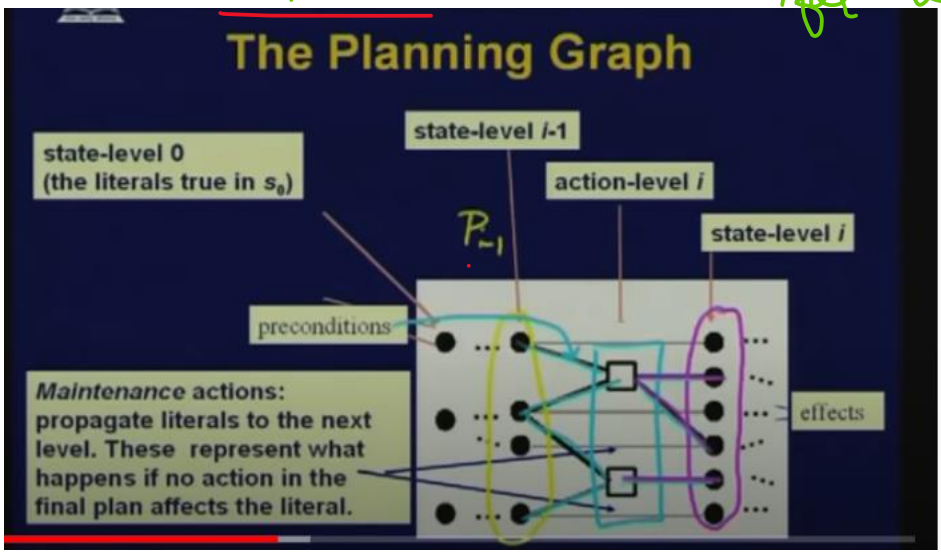
- Proposition: P
- Action: A

-Preposition and action levels alternate.

-Action levels include actions whose preconditions are satisfied in previous level plus no-op where same precondition is carried over.

DAYANANDA SAGAR COLLEGE OF ENGINEERING
Dept. of Information Science & Engg
(An Autonomous Institute Affiliated to VTU, Belagavi)
Shavige Malleshwara Hills, Kumaraswamy Layout, Bengaluru-560078
Dr. Rajeshwari.J
Associate Professor





Procedure for graph plan:

Graph plan algorithm:

For K=0, 1,2

-Graph expansion:

Dr. Rajeshwari.J
Associate Professor

-Create a planning graph that contains k levels. (apply some constraints in the planning graph which is a relaxed problem).

-Check whether the planning graph satisfies a necessary condition for plan existence.

-If it does then do solution extraction.

Backward search which search the actions which are in planning graph.

If we find a solution the return it.

-Let p1: all literals from the initial state.

-Add an action in level Ai if all its precondition is present in level pi

-Add a precondition in level pi if it is the effect of some action at level Ai-1.

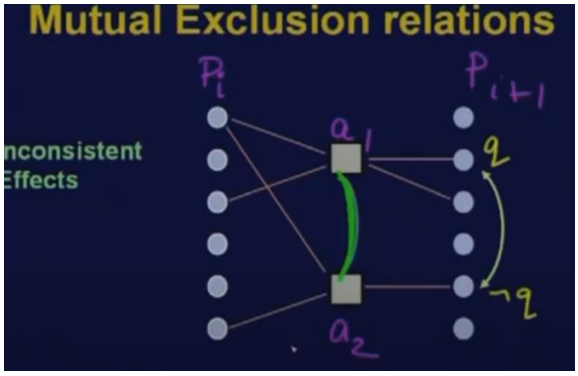
-Maintain a set of exclusion relations to eliminate incompatible propositions and actions(thus reducing the graph size).

Mutex:

If two actions or two propositions cannot be present in a valid plan then it is called a Mutex.

The two actions are mutex if:

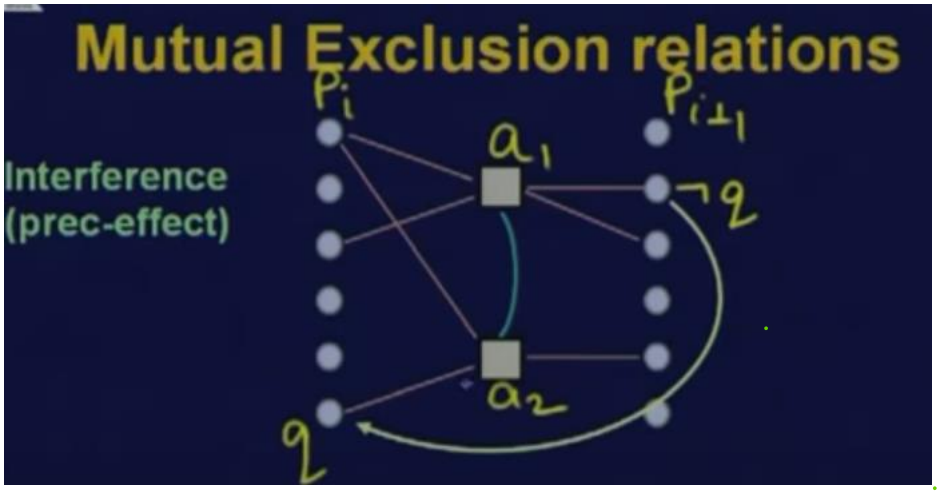
-*Inconsistent effects*: effect of one negates the effect of other.



Two actions are mutes if there effects are inconsistent.

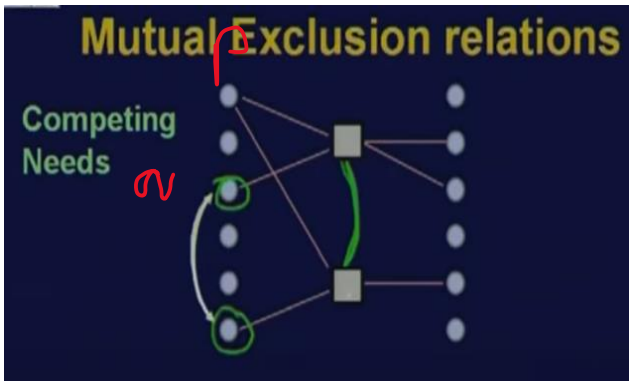
-***Interference: one deletes the precondition of other***

Dr. Rajeshwari.J
Associate Professor



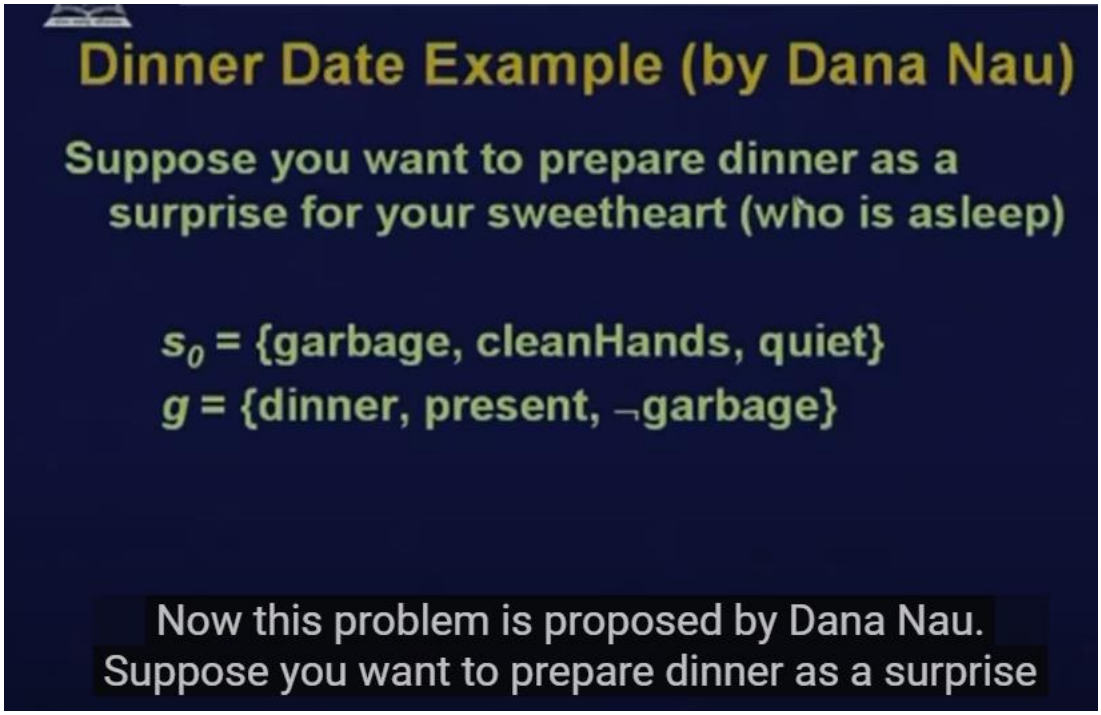
Action a1 and a2 are mutex if action a1 has an effect not q that is action a1 deletes q where q is the prcondtion of a2. Therefore these two actions are mutex. These two actions cannot be together.

-*competing needs:* if actions have mutually exclusive preconditions, if preconditions cannot occur together, then at next stage we cannot have those two actions together.

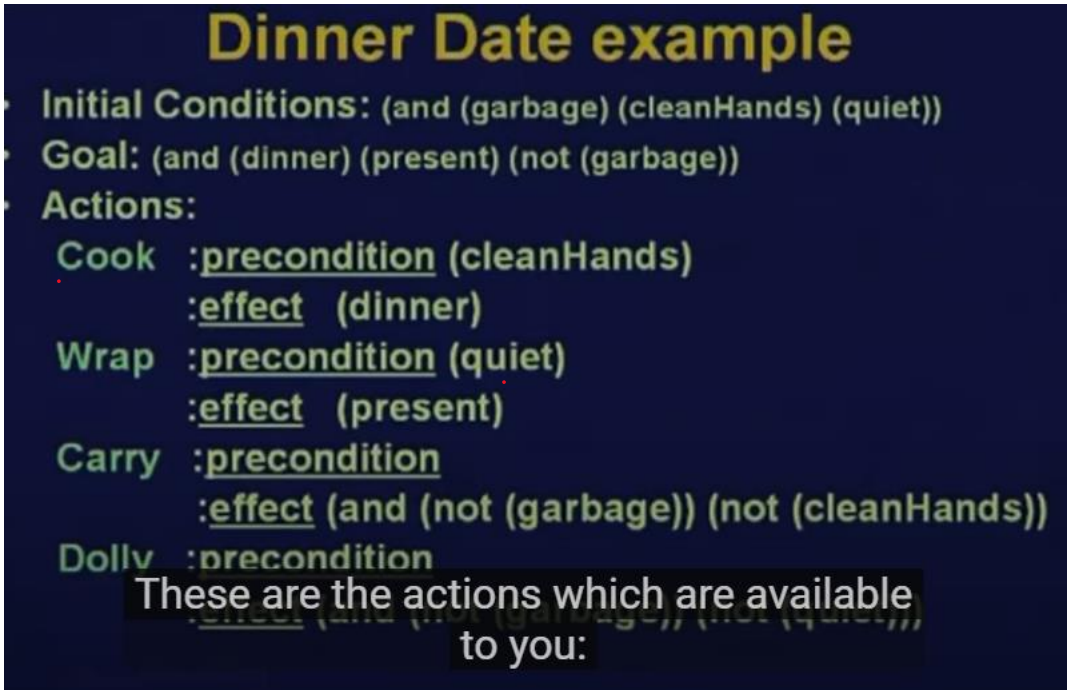


Example of a planning graph:

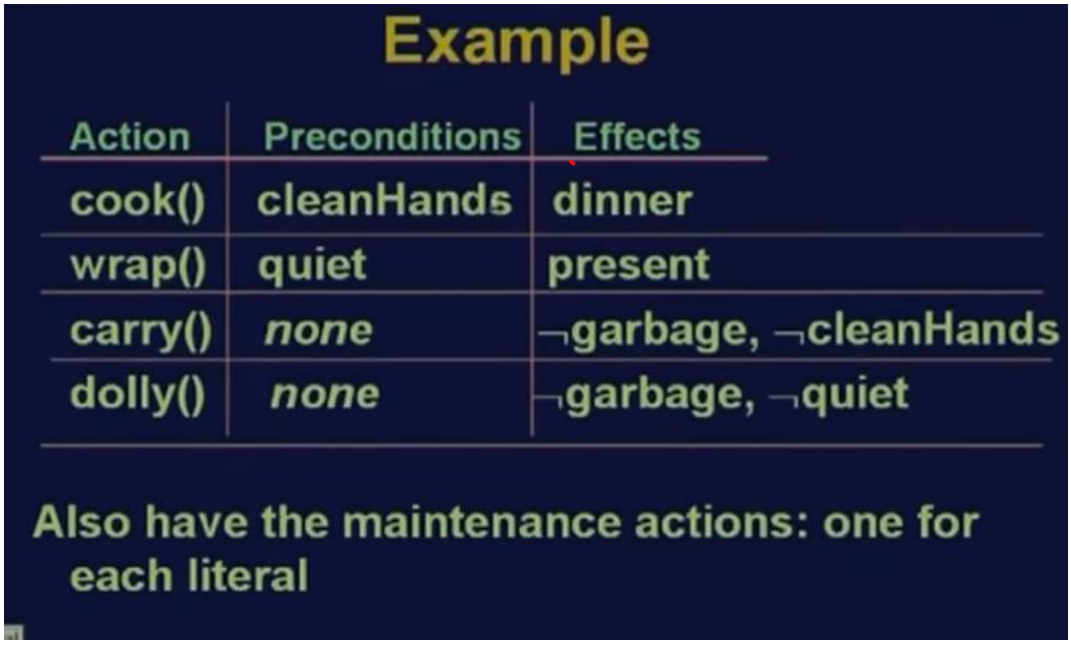
Dr. Rajeshwari.J
Associate Professor



So is the initial state and g is the goal stat(e.



Dr. Rajeshwari.J
Associate Professor



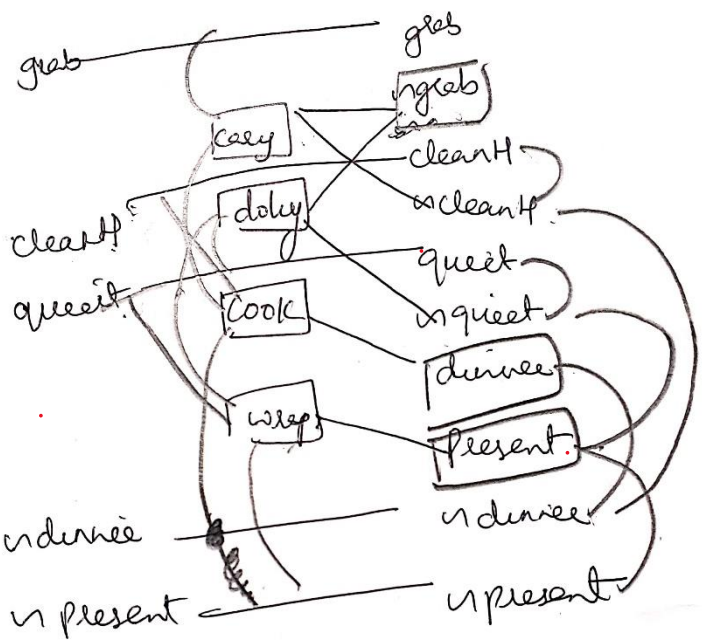
DAYANANDA SAGAR COLLEGE OF ENGINEERING Dept. of Information Science & Engg

(An Autonomous Institute Affiliated to VTU, Belagavi)

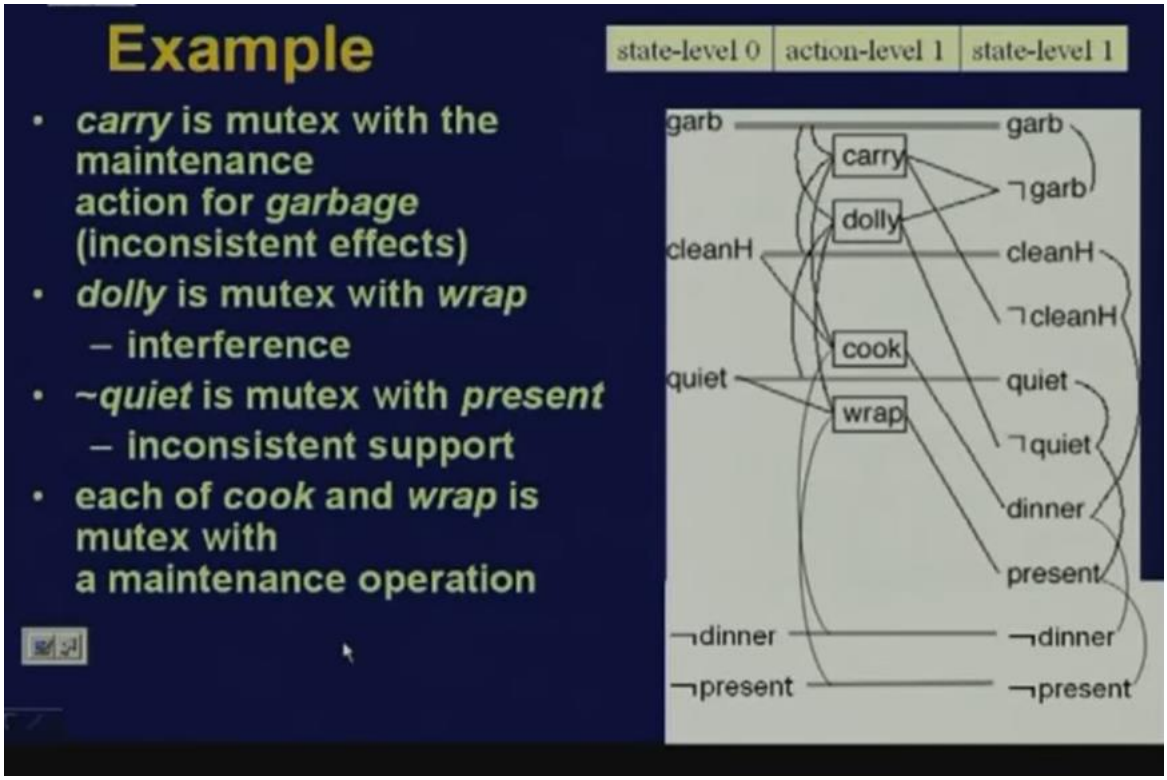
Shavige Malleshwara Hills, Kumaraswamy Layout, Bengaluru-560078

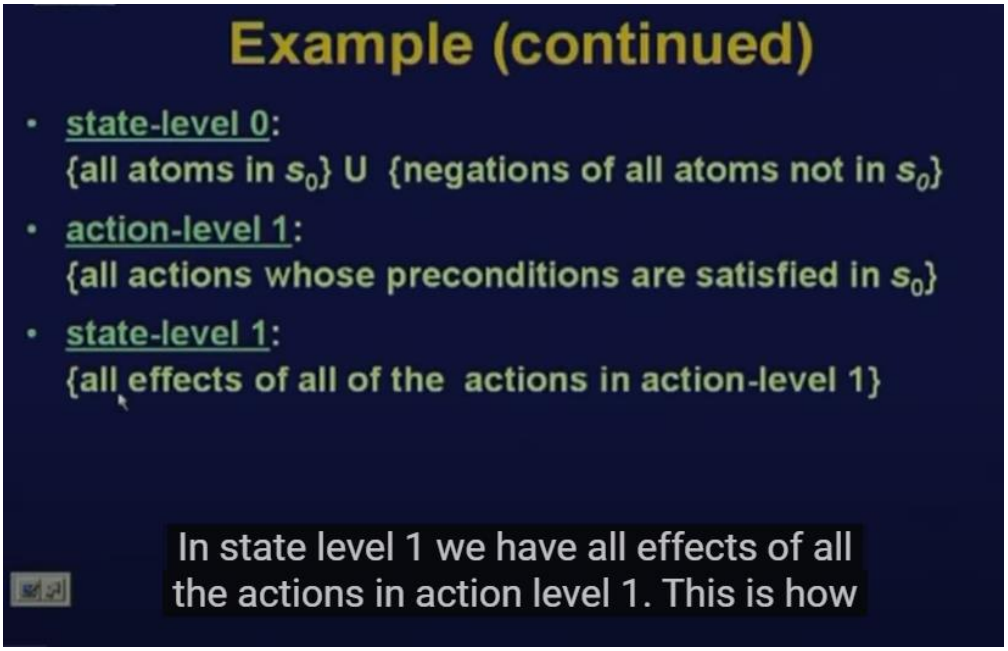
Dr. Rajeshwari.J

Associate Professor



DAYANANDA SAGAR COLLEGE OF ENGINEERING Dept. of Information Science & Engg
(An Autonomous Institute Affiliated to VTU, Belagavi)
Shavige Malleshwara Hills, Kumaraswamy Layout, Bengaluru-560078
Dr. Rajeshwari.J
Associate Professor





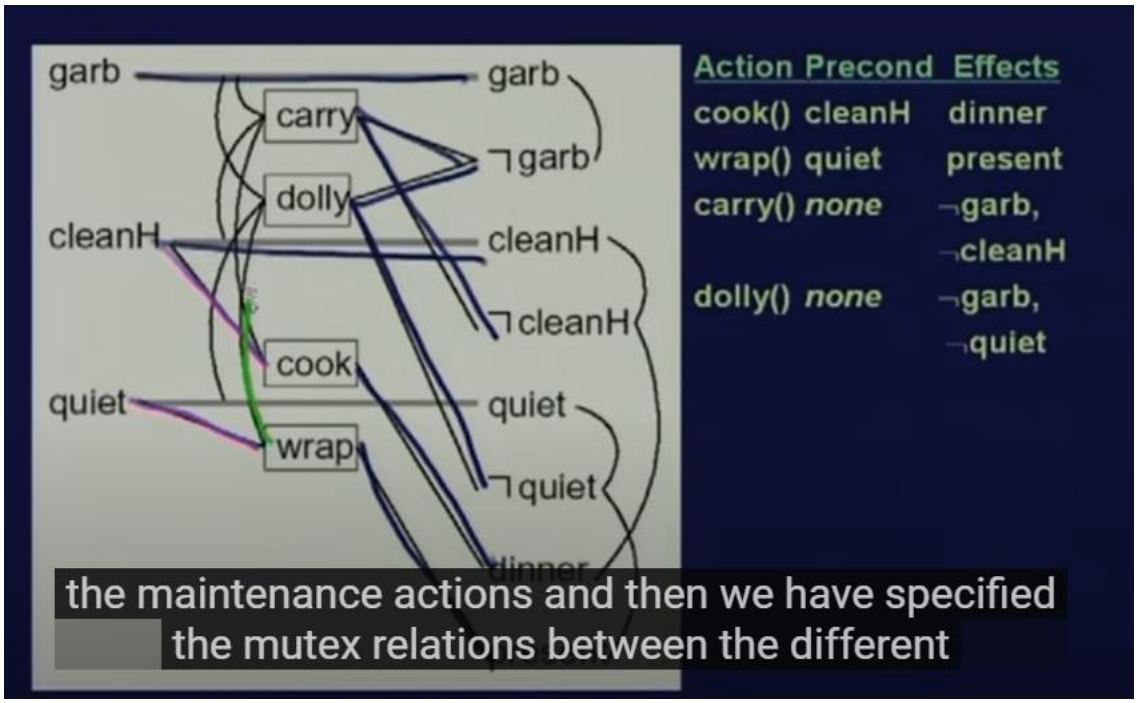
DAYANANDA SAGAR COLLEGE OF ENGINEERING Dept. of Information Science & Engg

(An Autonomous Institute Affiliated to VTU, Belagavi)

Shavige Malleshwara Hills, Kumaraswamy Layout, Bengaluru-560078

Dr. Rajeshwari.J

Associate Professor



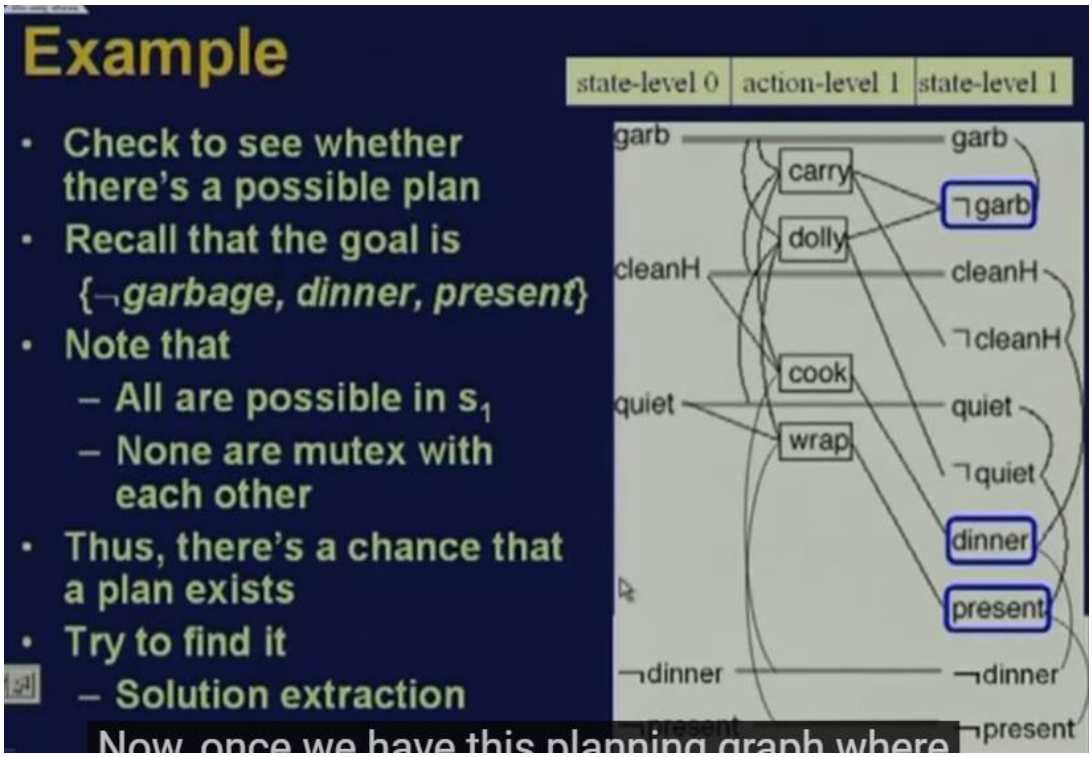
DAYANANDA SAGAR COLLEGE OF ENGINEERING Dept. of Information Science & Engg

(An Autonomous Institute Affiliated to VTU, Belagavi)

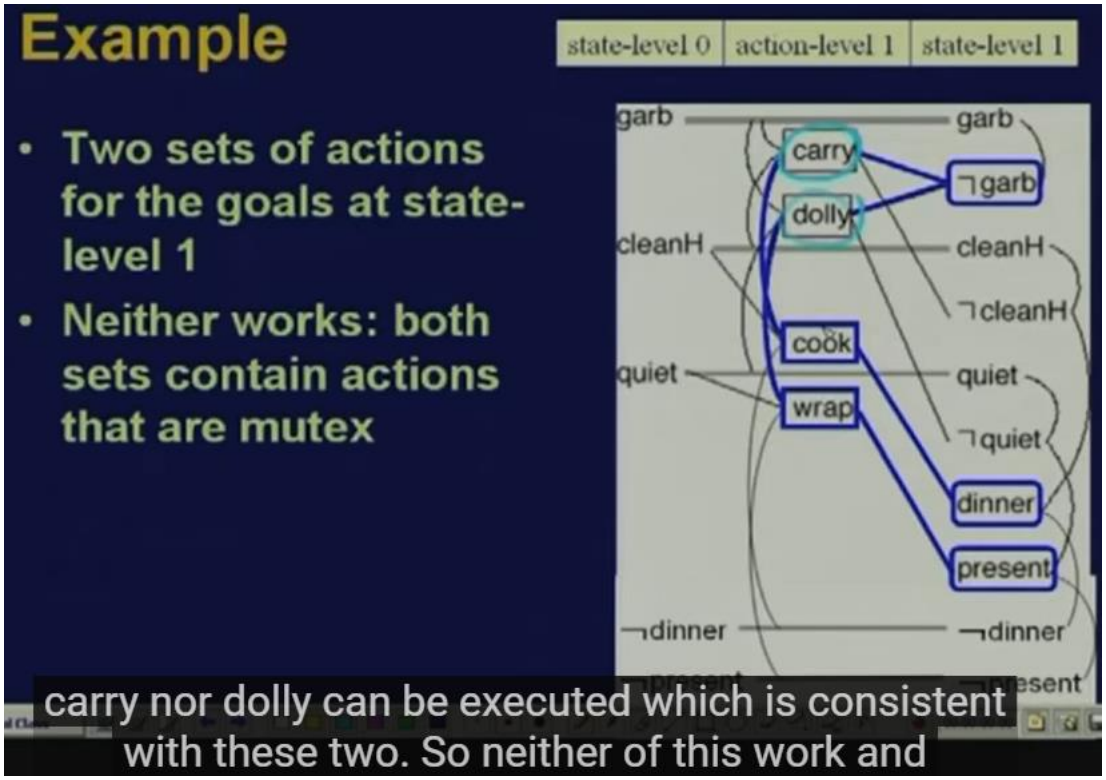
Shavige Malleshwara Hills, Kumaraswamy Layout, Bengaluru-560078

Dr. Rajeshwari.J

Associate Professor



Dr. Rajeshwari.J
Associate Professor



Not garbage, dinner, present in the goal, and they are not mutex. This satisfies the necessary condition that plan exist. Now we have to do plan extraction to find the solution.

For present we have to take wrap, for dinner we have to take cook and not garbage we have to take carry or dolly. Cook is mutex with carry and wrap is mutex with dolly. So none of these ways we can satisfy not garb action. So this plan cannot be extracted. So we have to expand the graph one more level.

Note: As we increase the number of levels the mutex decreases.

A valid plan is a planning graph where:

-Actions at the same level don't interfere.

-Each actions precondition are made true by the plan.

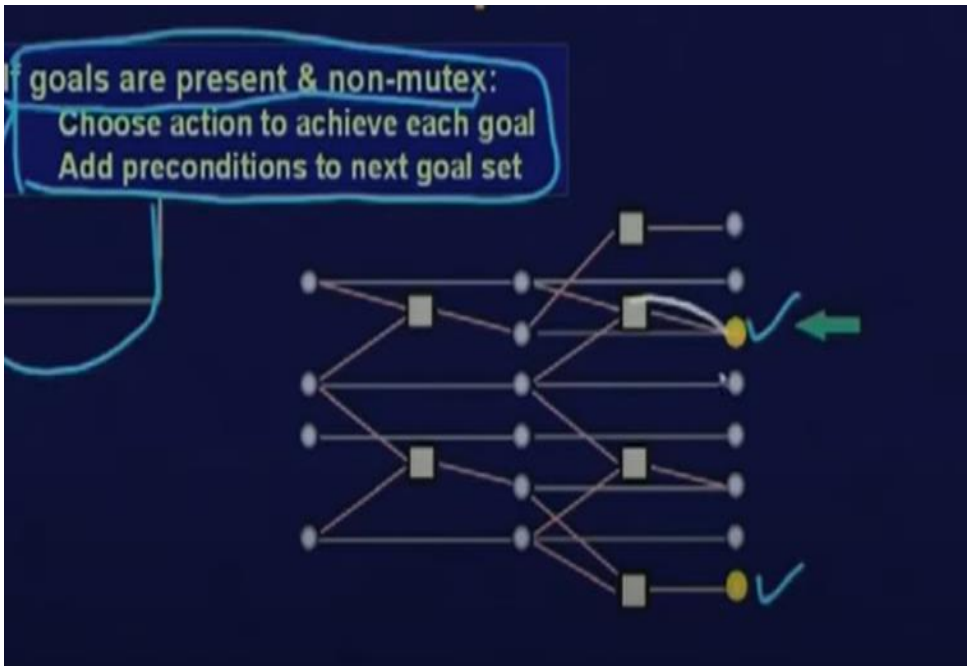
-Goals are satisfied.

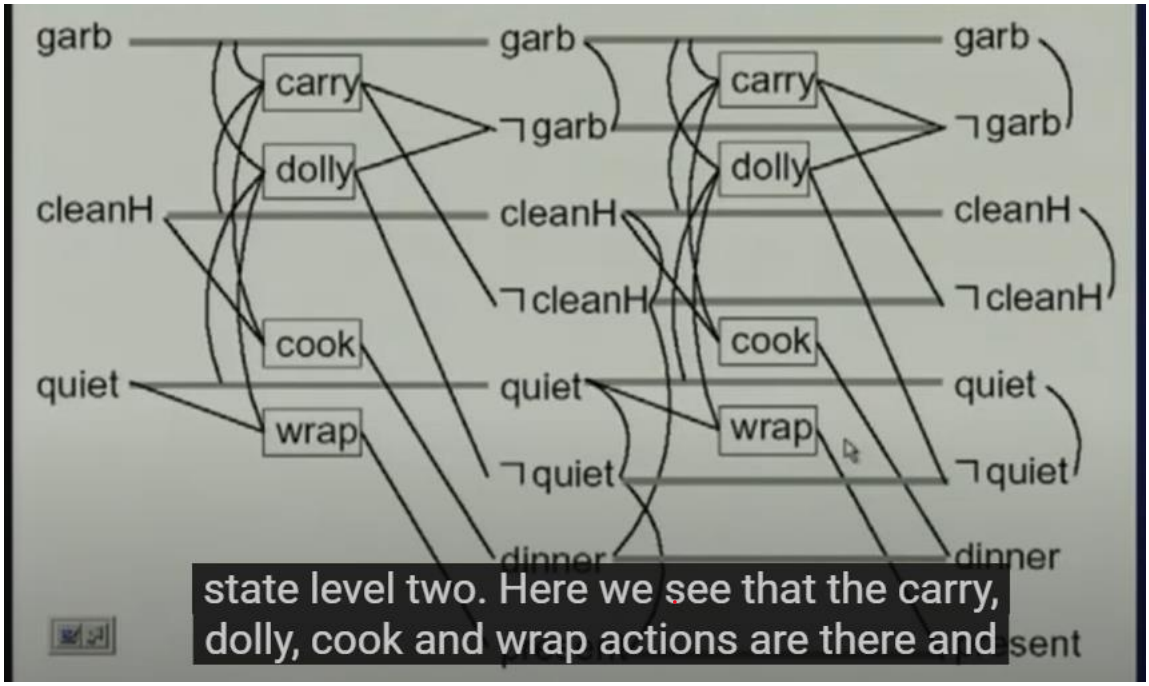
Searching for a solution plan:

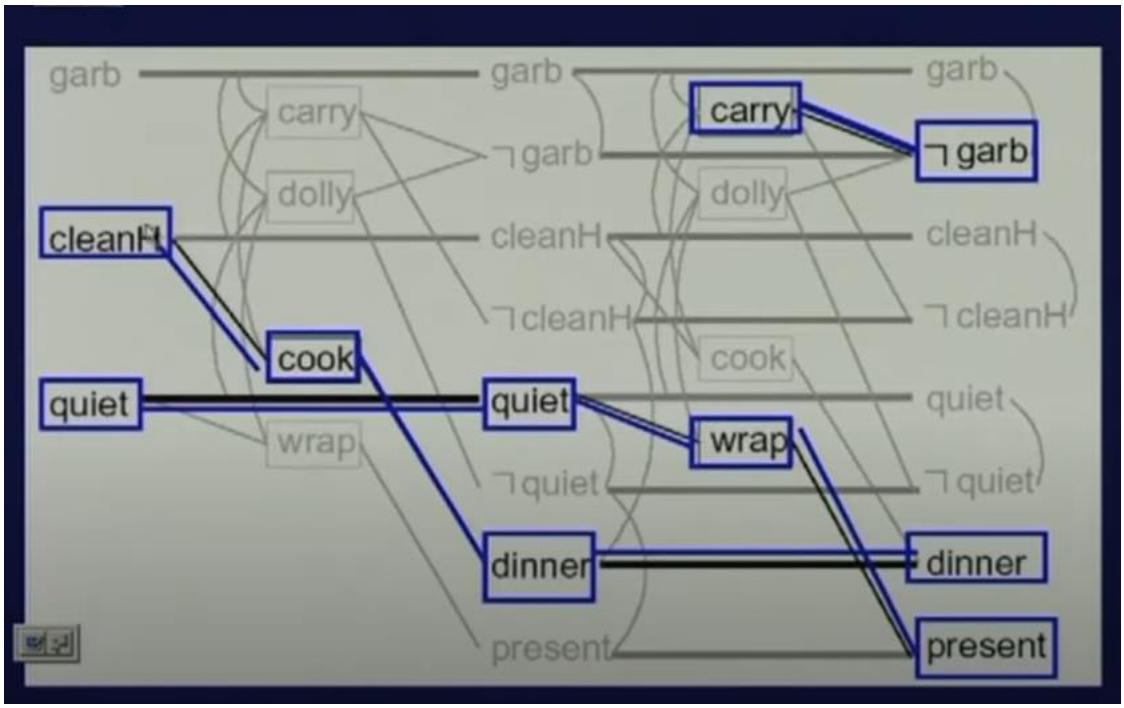
-Backward chain on the planning graph

-Achieve goal level by level.

-At level k, pick a subset of non mutex, actions to achieve, current goals. There preconditions becomes goal for k-1 level.







Thus we have found a plan with preconditions CleanH and quiet. To achieve the goal.

5.4 Acting under Uncertainty:

When an agent knows enough facts about its environment, the logical plans and actions produce guaranteed work.

Unfortunately, agents never have access to the whole truth about the environment, agents. Agent acts under uncertainty.

We'll use medical/dental diagnosis examples extensively

our new prototype problem relates to whether a dental patient has a cavity or not

the process of diagnosis always involves uncertainty & this leads to difficulty with logical representations (propositional logic examples)

Dr. Rajeshwari.J
Associate Professor

(1) toothache ⇒ cavity: If we refer (1) is just wrong since other things cause toothaches

(2) toothache ⇒ cavity ∨ gumDisease ∨ .....

// will need to list all possible causes. Some of them have gumdisease, swelling, cavity or several other cases.

(3) cavity ⇒ toothache

//tries a causal rule but it's not always the case that cavities cause toothaches & fixing the rule requires making it logically exhaustive.

Logic is not sufficient for medical diagnosis, due to

--our Laziness: it's too hard to list all possible antecedents or consequents to make the rule have no exceptions..

The above example is relationship between toothache & cavities and is not a logical consequence in either direction.

Instead, we have to go for knowledge of the domain which provides a degree of belief in diagnostic sentences

& the way to represent this is with probability theory.

Epistemological Commitment:

The possible states of knowledge for logic

-→ sentences are true/false/unknown

→ for probability theory, there's a numerical degree of belief in sentences, between 0 (certainly false) and 1 (certainly true)

->a probabilistic agent makes statements with respect to the knowledge state, & these may change as the state of knowledge changes

->for example, an agent initially may believe there's an 80% chance (probability 0.8) that the patient with the toothache has a cavity, but subsequently revises that as additional evidence is available.

In probability theory,

The probability that patient has cavity .8

The above statement is about the agents belief, not directly about the world.

These precepts create the evidence which are based on probability statements.

Dr. Rajeshwari.J
Associate Professor

Making choices among plans/actions:

Choose based on utility value. Every state has a degree of utility/usefulness to the agent & the agent will prefer those with higher utility.

the agent incorporates probabilistic predictions of action outcomes, selecting the one with the highest expected utility.

function DT-AGENT (percept) returns an action

persistent: belief-state, probabilistic beliefs about the current state of the world

action, the agent's action

update belief-state based on action and percept

calculate outcome probabilities for actions,

//given action descriptions and current belief state

select action with the highest expected utility,

//given probabilities of outcomes and utility information

return action.

Dr. Rajeshwari.J
Associate Professor

5.4 Inference using Joint Probability Distribution:

Joint probability distribution completely specifies an agent's probability assignment to all propositions in the domain.

Joint probability distribution assigns probabilities to all possible atomic events.

An n-dimensional table with a value in every cell giving the probability of that specific state occurring.

probabilities in the joint distribution sum to 1.

	Toothache	~Toothache
Cavity	.04	.06
~Cavity	.01	.89

Adding across row or column gives us the unconditional probability of a variable.

P(cavity ∨ toothache)= .04+.06+.01=.11

One particularly common task is to extract the distribution over some subset of variables or a single variable

P(cavity)=.04+.06=.10

adding the entries in the first row gives the unconditional or marginal probability of cavity. This process is called marginalization, or summing out.

Conditional probabilities : Conditional probabilities can be found from joint probability distribution.

Probability of cavity given the evidence of toothache.

P(cavity/toothache)=P(cavity^toothache)/P(toothache)=0.04/.04+.01=0.8

Probability of ~cavity given the evidence of toothache.

P(~cavity/toothache)=P(~cavity^toothache)/P(toothache)=.01/.04+.01=0.2

DAYANANDA SAGAR COLLEGE OF ENGINEERING Dept. of Information Science & Engg

(An Autonomous Institute Affiliated to VTU, Belagavi)

Shavige Malleshwara Hills, Kumaraswamy Layout, Bengaluru-560078

Dr. Rajeshwari.J

Associate Professor

Another Example:

13.3 Inference Using Full Joint Distribution

	toothache		¬toothache	
	catch	¬catch	catch	¬catch
cavity	0.108	0.012	0.072	0.008
¬cavity	0.016	0.064	0.144	0.576

- E.g., there are six atomic events for cavity ∨ toothache:
  0.108 + 0.012 + 0.072 + 0.008 + 0.016 + 0.064 = 0.28
- Extracting the distribution over a variable (or some subset of variables), *marginal probability*, is attained by adding the entries in the corresponding rows or columns
- E.g., P(cavity) = 0.108 + 0.012 + 0.072 + 0.008 = 0.2
- We can write the following general marginalization (summing out) rule for any sets of variables **Y** and **Z**:

$$\mathbf{P}(\mathbf{Y}) = \Sigma_{\mathbf{z} \in \mathbf{Z}} \mathbf{P}(\mathbf{Y}, \mathbf{z})$$

Dr. Rajeshwari.J
Associate Professor

	toothache		¬toothache	
	catch	¬catch	catch	¬catch
cavity	0.108	0.012	0.072	0.008
¬cavity	0.016	0.064	0.144	0.576

- Computing a conditional probability
  P(cavity | toothache) =
  P(cavity ∧ toothache)/P(toothache) =
  (0.108 + 0.012)/(0.108 + 0.012 + 0.016 + 0.064) =
  0.12/0.2 = 0.6
- Respectively
  P(¬cavity | toothache) =
  (0.016 + 0.064)/0.2 = 0.4
- The two probabilities sum up to one, as they should

.

Independence:

Dental problems have no influence on other.

P(cloudy |toothache, catch, cavity) = P(cloudy) . Equation-1

From this, we can deduce

P(toothache, catch, cavity, cloudy) = P(cloudy)P(toothache, catch, cavity) .

The property we used in Equation-1 is called independence (also marginal independence and absolute independence). In particular, the weather is independent of one's dental problems. Independence between propositions a and b can be written as P(a | b) = P(a) or P(b | a) = P(b)

Dr. Rajeshwari.J
Associate Professor

5.4 Conditional Probability: What is the probability of an event given knowledge of another Event.

Example:

P(Raining/Sunny)

P(Raining/Cloudy)

P(Raining/Cloudy, cold)

Bayer's rule is given by:

Given a Knowledge of one or more random variable, we can improve upon our belief of another random variable using the Bayers rule:

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

Prior Probability

Prior / Evidence

Bayer's requires three terms:

Two prior probabilities

A conditional probability.

P(B/A) is the probability of Hypothesis B(predict) conditional on a new piece of Evidence A

Here P(A/B) is the probability of evidence given the hypothesis.

P(B):is the prior probability of hypothesis.

P(A): is the prior probability of Evidence.

Example 1:

Dr. Rajeshwari.J
Associate Professor

For example, a doctor knows that the disease meningitis causes the patient to have a stiff neck, say, 70% of the time. The doctor also knows some unconditional facts: the prior probability that a patient has meningitis is 1/50,000, and the prior probability that any patient has a stiff neck is 1%. Letting s be the proposition that the patient has a stiff neck and m be the proposition that the patient has meningitis, we have

$$
\begin{aligned}
P(s \mid m) &= 0.7 \\
P(m) &= 1/50000 \\
P(s) &= 0.01 \\
P(m \mid s) &= \frac{P(s \mid m)P(m)}{P(s)} = \frac{0.7 \times 1/50000}{0.01} = 0.0014 .
\end{aligned}
$$

Example 2:

Probability of having Wheezing:

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

P(H)=Having Wheezing=0.0001

P(E/H)= Probability of symptoms given Wheezing=.95

P(E)=Probability of having symptoms cough, cold, fever=.01

Using Bayer's rule we have to find probability wheezing given the symptoms==?--P(H/E)

Dr. Rajeshwari.J
Associate Professor

Being an alcoholic" is the **test** (kind of like a litmus test) for liver disease.

- **A** could mean the event "Patient has liver disease." Past data tells you that 10% of patients entering your clinic have liver disease. P(A) = 0.10.
- **B** could mean the litmus test that "Patient is an alcoholic." Five percent of the clinic's patients are alcoholics. P(B) = 0.05.
- You might also know that among those patients diagnosed with liver disease, 7% are alcoholics. This is your **B|A:** the probability that a patient is alcoholic, given that they have liver disease, is 7%.

Bayes' theorem tells you:

P(A|B) = (0.07 * 0.1)/0.05 = 0.14

In other words, if the patient is an alcoholic, their chances of having liver disease is 0.14 (14%). This is a large increase from the 10% suggested by past data. But it's still unlikely that any particular patient has liver disease.

Q1: Patient has liver disease." Past data tells you that 10% of patients entering your clinic have liver disease. Patient is an alcoholic." Five percent of the clinic's patients are alcoholics. You might also know that among those patients diagnosed with liver disease, 7% are alcoholics. What is the chances of having a liver disease if the patient is alcoholic?

Q2: Example:

- dangerous fires are rare (1%)
- but smoke is fairly common (10%) due to barbecues,
- and 90% of dangerous fires make smoke.

What is the probability of dangerous fire when there is a smoke.

$$(\text{Fire}|\text{Smoke}) = P(\text{Fire})\,P(\text{Smoke}|\text{Fire})\mathbf{P(\text{Smoke})}$$
$$= 1\% \times 90\%\,\mathbf{10\%}$$
$$= 9\%$$

Q3: We will use Rain to mean rain during the day,

- (Rain) is Probability of Rain = 10%
- P(Cloud|Rain) is Probability of Cloud, given that Rain happens = 50%
- P(Cloud) is Probability of Cloud = 40%

Dr. Rajeshwari.J
Associate Professor

P(Rain|Cloud) = *0.1 x 0.5/* **0.4** = .125

Joint probability distribution

- The full joint probability distribution specifies the probability of values to random variables.
- It is usually too large to create or use in its explicit form
- Joint Probability distribution of two variables X and Y are

Joint. Probability	X	X'
Y	0.20	0.12
Y'	0.65	0.03

- Joint probability distribution for n variables requires 2^n entries will all passible combination.

Drawbacks of Joint probability distributions.

→ Large number of variable grow rapidly.

→ Time and space complexity are huge.

→ Alternative to this is Bayesian Network.

(2)

Conditional Probability Table

$P(A) = 0.3$

$P(B) = 0.6$

$P(C|A) = 0.4$

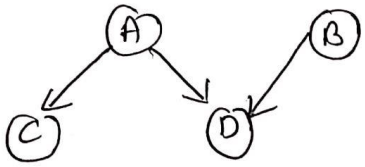
$P(C|\neg A) = 0.2$

$P(D|A,B) = 0.7$

$P(D|A,\neg B) = 0.4$

$P(D|\neg A,B) = 0.2$

$P(D|\neg A,\neg B) = 0.01$



CPT

P(A)	P(B)
0.3	0.6

A	P(C)
T	0.4
F	0.2

A	B	P(D)
T	T	0.7
T	F	0.4
F	T	0.2
F	F	0.01

$P(A,B,C,D) = P(A) * P(B) * P(C|A) * P(D|(A,B))$

(7)

Bayesian Network:

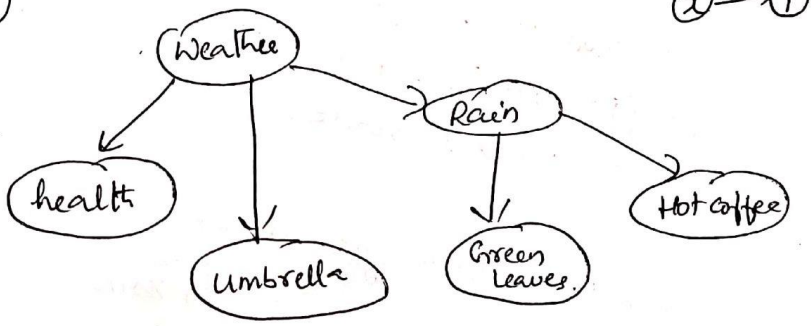
* A Bayesian Belief Network is a probabilistic Graphical model that represents conditional dependencies between random variables through a Directed Acyclic Graph (DAG)

* The graph consists of nodes and Arcs.

* The node represents variables which can be discrete or continuous.

* The arcs represent causal relationship s. If there is an arrow from node X to node Y, then X is a Parent of Y.

Example: (1)

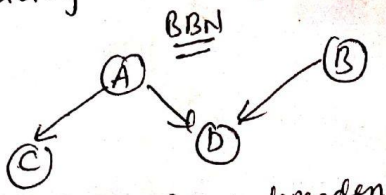


(2) → (Y)

*

→ BBN Enable us to model and reason about uncertainity

Example2:



A & B are unconditional, independent, Evidence & Parent nodes
C & D are conditional, dependent, hypothesis & child nodes.

Two types of Probability comes to our rescue.

(1) Joint Probability.
(2) Conditional Probability.

(3)

Joint Probability:-

P(Weather, Health, Umbrella) → Joint Probability

$= \pi \cdot P(\text{Weather}) * P(\text{Health} \mid \text{Weather}) * P(\text{Umbrella} \mid \text{Weather})$ → Conditional Probability

Generalising:

$$P(x_1, x_2, x_3 \ldots x_n) = \prod_{i=1}^{n} P(x_i \mid \text{Parents}(x_i))$$

denotes Belief

Each node X_i has a conditional probability distribution $P(X_i \mid \text{Parent}(X_i))$ that quantifies the effect of the Parents on that node.

Bayesian Network – Burglar Alarm:-

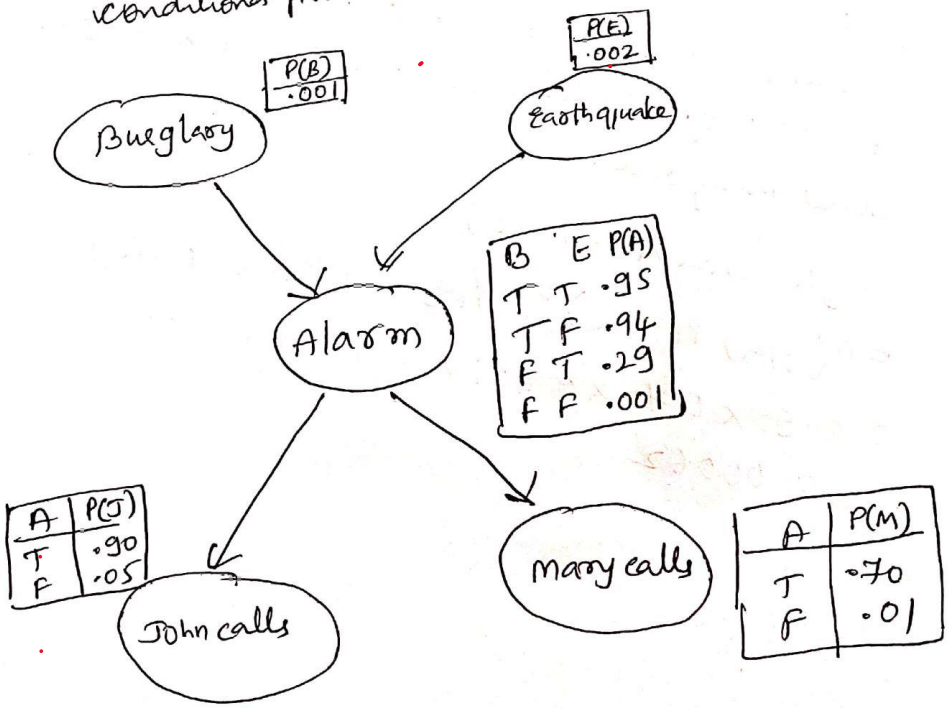
- Installed a new alarm at home.
- It is fairly reliable at detecting a burglary, but also responds on occasion to minor earthquakes.
- You also have two neighbors, John and Mary, who have promised to call you at work when they hear the alarm.

– John always calls when he hears the alarm, but sometimes confuses the telephone singing with the alarm and calls them too.

– Mary on other hand, likes loud music and sometimes misses the alarm altogether.

→ Given the evidence, of who has or has not called, we would like to estimate the probability of a burglary.

Note: • Probability help make an inference.
• Random variables are Boolean.
• Each node is associated with a conditional probability table.



- In burglary network, the topology shows that
  - burglary and Earthquakes directly affect the probability of the alarm.
  - But John and Mary call depends on the alarm.

Suppose mary is currently listening to loud music or Telephone ringing and confusing John.
These factors are summarised in the uncertainty associated with links from Alarm to JohnCalls and MaryCalls.

Example:- we can calculate the probability that the alarm has sounded, but neither a burglary or an earthquake has occured, and both John and Mary call.

$$P(j \wedge m \wedge a \wedge \neg b \wedge \neg e)$$
$$= P(j|a)\, P(m|a)\, P(a|\neg b \wedge \neg e)\, P(\neg b)\, P(\neg e)$$
$$= 0.90 \times 0.70 \times 0.001 \times 0.999 \times 0.998$$
$$= 0.00062$$

Q. $P(J) = ?$

$= P(J/A) * P(A) + P(J/\bar{A}) * P(\bar{A})$

$= .90 \times P(A) + .05 * P(\bar{A})$

$= .90 * .00252 + .05 * .9974$

$= .0521$

$P(A) = P(A/B,E) * P(B \wedge E) +$

$P(A/\bar{B},E) * P(\neg B \wedge E) +$

$P(A/B\bar{E}) * P(B \wedge \neg E) +$

$P(A/\bar{B}\bar{E}) * P(\neg B \wedge \neg E)$

$= 0.95 * P(B) * P(E) +$

$.29 * P(\bar{B}) * P(E) +$

$.95 * P(B) * P(\bar{E}) +$

$.001 * P(\bar{B}) * P(\bar{E})$

$= .95 * 0.001 * 0.002 +$

$.29 * .999 * .002 +$

$.95 * .001 * .998 +$

$.001 * .999 * .998 = .00252.$

$P(\bar{A}) = P(\bar{A}/B,E) \times P(B \wedge E) +$

$P(\bar{A}/\bar{B},E) * P(\neg B \wedge E) +$

$P(\bar{A}/B\bar{E}) * P(B \wedge \neg E) +$

$P(\bar{A}/\bar{B}\bar{E}) * P(\neg B \wedge \neg E)$

$= (1 - 0.95) * 0.001 * 0.02 +$

$(1 - .29) * 0.999 * 0.02 +$

$(1 - .95) * .001 * .998 +$

$(1 - 0.01) * .999 * .998$

$= .9974$

Semantics of Bayesian Network

- An entry in joint distribution is the probability of conjunction of particular assignment to each variable such as

$P(X_1 = x \wedge X_2 = x_2 \wedge \ldots \wedge X_n = x_n)$ is equal to

$$P(x_1, x_2 \ldots x_n) = \prod_{i=1}^{n} P(x_i \mid \text{Parent}(X_i))$$

- Then we repeat the process, reducing each conjunctive probability to a conditional probability and a smaller conjunction. we end up with one big product.

$$P(x_1 \ldots x_n) = P(x_n \mid x_{n-1} \ldots x_1) \cdot P(x_{n-1} \mid x_{n-2} \ldots x_1) \ldots P(x_2 \mid x_1) P(x_1)$$

$$= \prod_{i=1}^{n} P(x_i \mid x_{i-1} \ldots x_1)$$

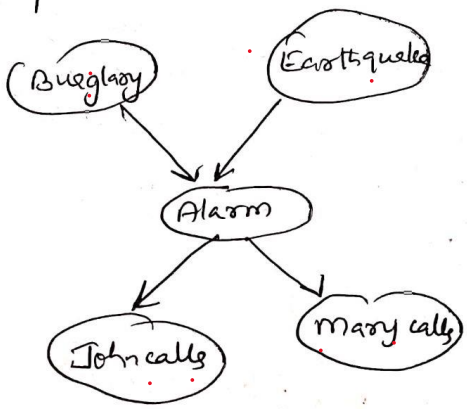
$$P(X_i \mid X_{i-1}, \ldots X_1) = P(X_i \mid \text{Parents}(X_i))$$

Eg $P(\text{Marycalls} \mid \text{Johncalls}, \text{Alarm}, \text{Earthquake}, \text{Burglary})$

$= P(\text{marycalls} \mid \text{Alarm})$

Compactness of Node Ordering:-

- The compactness of Bayesian Network is an example of general property of locally constructed systems.
- In locally constructed system, each subcomponent interacts directly with only a bounded number of other components, regardless of the total number of components.



Eg:-
Johncalls and Marycalls are the subcomponent of the system. Johncalls is not related to Earthquake or Burglary. Marycalls is not related to (communicate) to Burglary or Earthquake. Johncalls and Marycalls react by the Alarm only.

Conditional Independence relations in Bayesian Networks

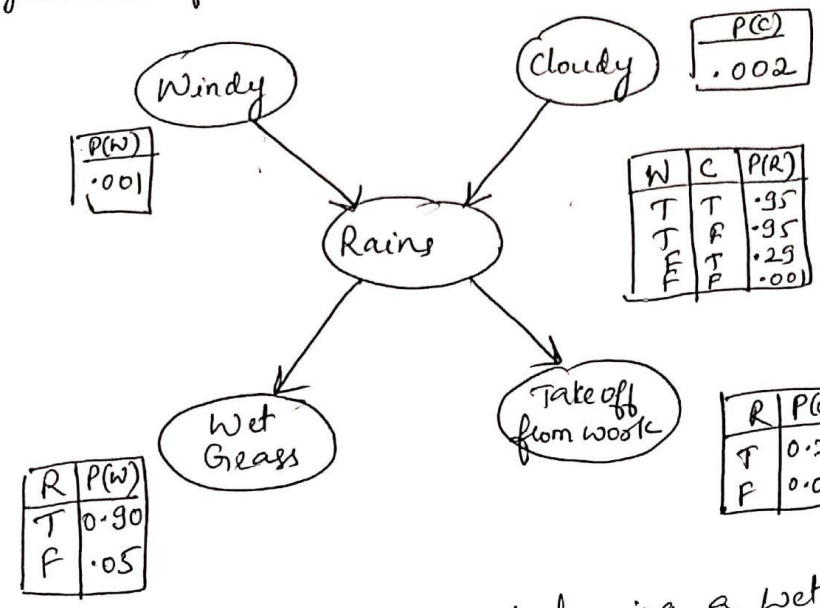
(1) A node is conditionally independent of its nondescendants given its parents.

- For Example: Burglary and Earthquake is independent of JohnCalls and MaryCalls.

(2) A node is conditionally independent of all other nodes in the network, given its Parents.

Eg:- MaryCalls and JohnCalls is independent of Burglary and Earthquake

Bayesian Belief Network



Lets find the Probability of having a wet grass!

$P(W) = P(W/R) * P(R) + P(W/\bar{R}) * P(\bar{R})$ — (1)

$P(W) = 0.9 * P(R) + 0.05 * P(\bar{R})$

$P(W) = 0.9 * 0.00252 + 0.05 * 0.99744 = 0.0521$

$P(R) = P(R/W,C) * P(W \wedge C) + P(R/\bar{W}C) * P(\neg W \wedge C) +$
$P(R/W\bar{C}) * P(W \wedge \neg C) + P(R/\bar{W}\bar{C}) * P(\bar{W} \wedge \bar{C})$

$= 0.95 * 0.001 * 0.002 + 0.29 * 0.999 * 0.002 +$
$0.95 * 0.001 * 0.998 + 0.001 * 0.999 * 0.0998$

$P(R) = 0.00252$

$P(\bar{R}) = P(R/W,C) * P(W \wedge C) + P(\bar{R}/\bar{W}C) * P(\bar{W} \wedge C)$
$+ P(\bar{R}/W\bar{C}) * P(W \wedge \bar{C}) + P(\bar{R}/\bar{W}\bar{C}) * P(\bar{W} \wedge \bar{C})$

$= (1-0.95) * 0.001 * 0.002 + (1-0.29) * 0.999 * 0.002 +$
$(1-0.95) * 0.001 * 0.998 + (1-0.001) * 0.999 * 0.998$
$= 0.99744$

Inference in Bayesian Belief Network

- Compute Posterior Probability Distribution for a set of query variables, given some observed events.
- Inference: Calculating some useful quantity from a joint probability distribution.

Example :- Posterior probability

$$P(Q \mid E_1 = e_1, \ldots E_k = e_k)$$

Types of Inferences
1) Inference by Enumeration
2) Variable elimination

General case :- // [Inference by Enumeration]

- Evidence variable $E_1 \ldots E_k = e_1 \ldots e_k$
- Query variable Q
- Hidden variables $H_1 \ldots H_r$

More specially a query $P(Q/E)$ can be answered by

$$P(Q/E) = \alpha P(Q, E) = \alpha \sum_H P(Q, E, H)$$

Consider $P(\text{Burglary} \mid \text{Johncalls} = \text{true}, \text{Marycalls} = \text{true})$

- Burglary – Query Variable (X)
- Johncalls – Evidence Variable (E_1)
- Marycalls – Evidence Variable (E_2)

The hidden variables of the query are Earthquake and Alarm.

Inference by Enumeration

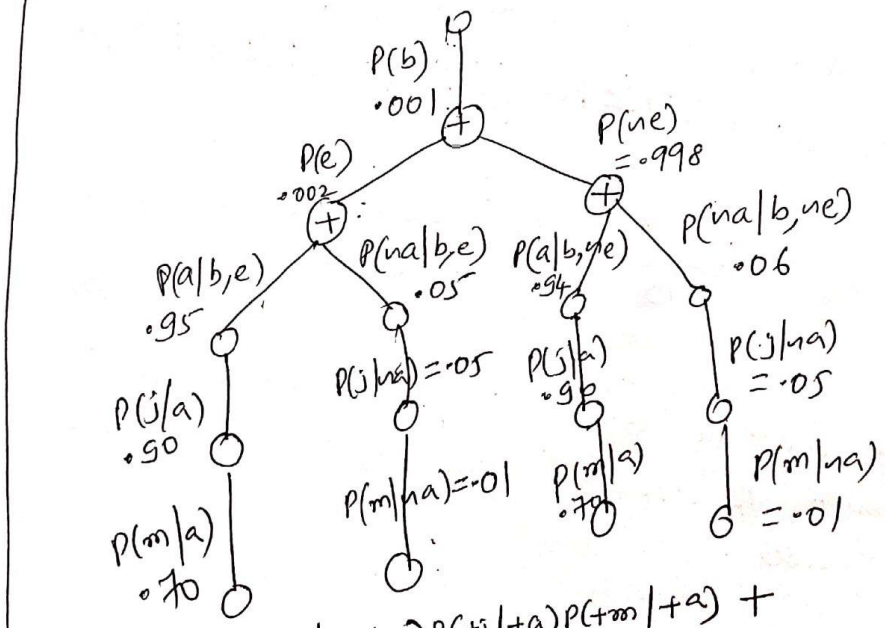
$$P(B|j,m) = \alpha P(B,j,m) = \alpha \sum_e \sum_a P(B,e,a,j,m)$$

- Even we have to consider hidden variables e and a (Evidence and alarm)

- The semantics of Bayesian Network give us an expression, in terms of CPT entries, for simplicity we do this just for Burglary = True.

$$P(b|j,m) = \alpha \sum_e \sum_a P(b)\,P(e)\,P(a|b,e)\,P(j|a)\,P(m|a)$$

burglary=true and johncalls = marycalls=true.



$$= P(B)\,P(+e)\,P(+a|B,+e)\,P(+j|+a)\,P(+m|+a) +$$
$$P(B)\,P(+e)\,P(-a|B,+e)\,P(+j|-a)\,P(+m|-a) +$$
$$P(B)\,P(-e)\,P(+a|B,-e)\,P(+j|+a)\,P(+m|+a) +$$
$$P(B)\,P(-e)\,P(-a|B,-e)\,P(+j|-a)\,P(+m|-a)$$

Inference by enumeration is slow because we join up the whole joint distribution before sum out the hidden variables.

Inference by Enumeration Example:

Random Variables

- R : Raining
- T : Traffic
- L : Late for class

$$P(L) = ?$$

$$= \sum_{r,t} P(r,t,L)$$

$$= \sum_{r,t} P(r)\,P(t|r)\,P(L|t)$$

P(R)

+r	0.1
−r	0.9

P(T|R)

+r	+t	0.8
+r	−t	0.2
−r	+t	0.1
−r	−t	0.9

P(L|T)

+t	+l	0.3
+t	−l	0.7
−t	+l	0.1
−t	−l	0.9

- First Basic operation : Joining factors
- Combining factors: Get all factors over the joining variable. Build a new factor, over the union of variable involved.
- Example : Join on R.

P(R) × P(T|R) ⟹ P(R,T)

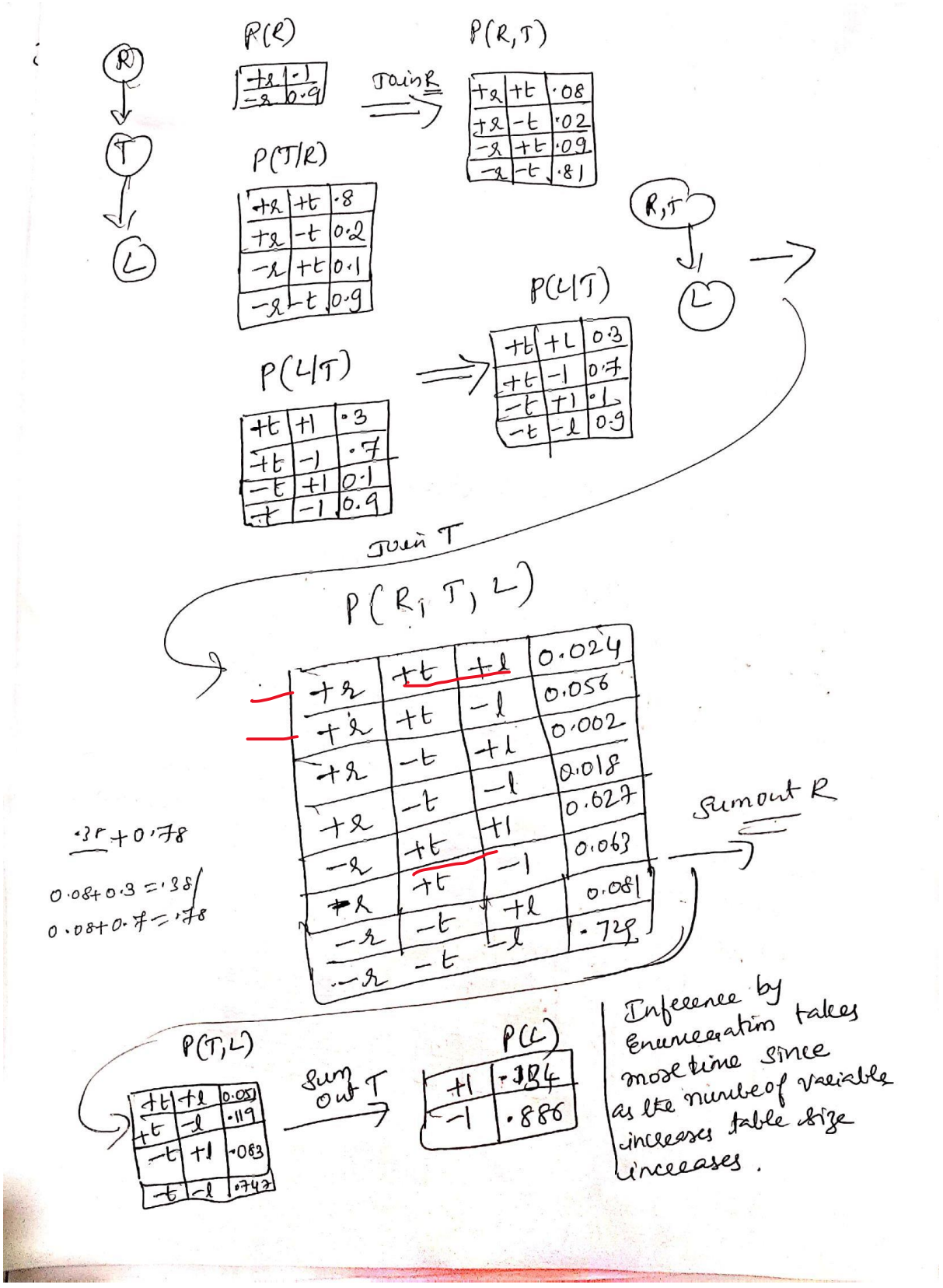
+r	0.1
−r	0.9

×

+r	+t	0.8
+r	−t	0.2
−r	+t	0.1
−r	−t	0.9

=

+r	+t	0.08
+r	−t	0.02
−r	+t	0.09
−r	−t	0.81


P(R)
P(R,T)
Join R
P(T/R)
P(L/T)
P(L/T)
R,T
Join T
P(R, T, L)
Sum out R
.38+0.378
0.08+0.3 = .38
0.08+0.7 = .78
P(T,L)
Sum out T
P(L)
Inference by Enumeration takes more time since as the number of variable increases table size increases.

• Inference by Enumeration

$$= \sum_t \sum_r P(r)\, P(t/r)\, P(L/t)$$

Join on r

Join on t

Eliminate r

Eliminate t

Variable Elimination

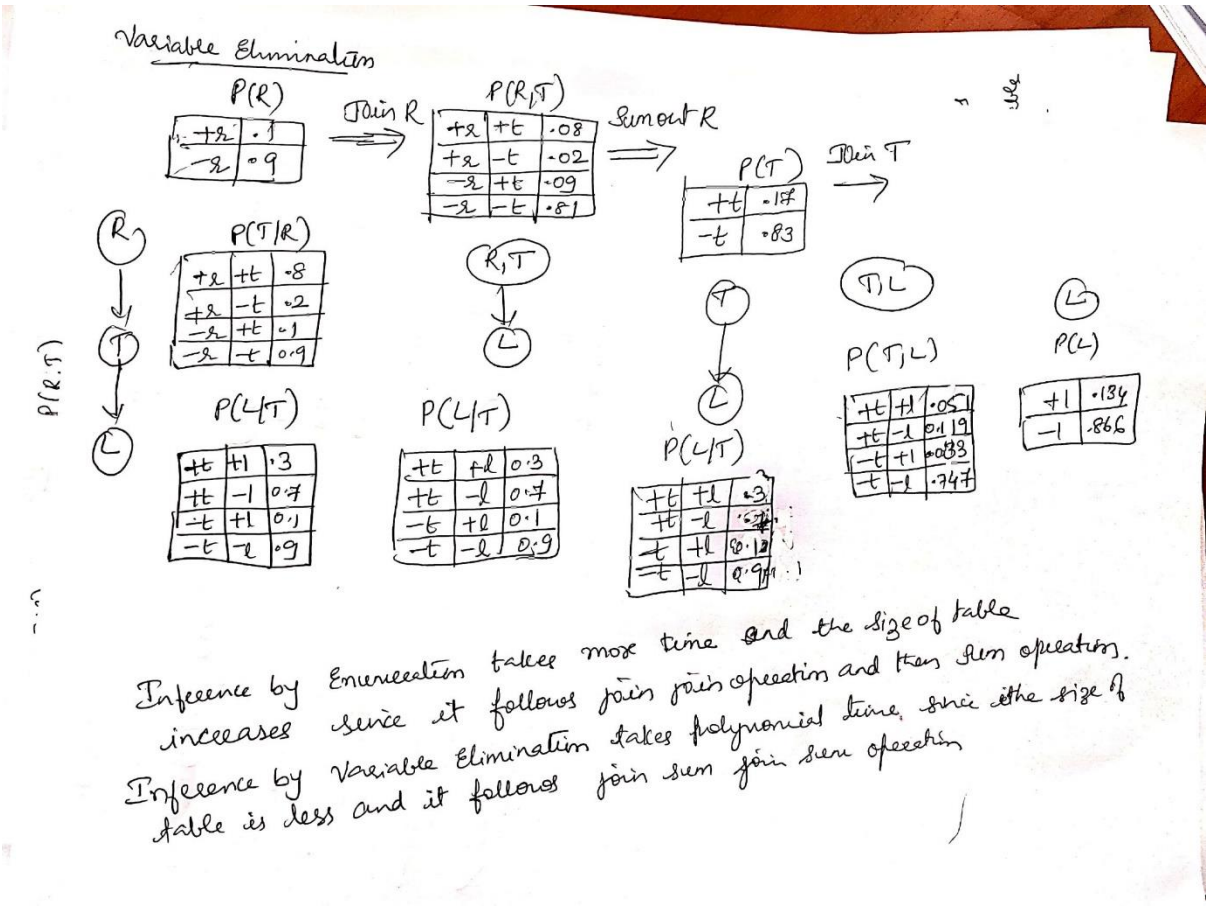
$$= \sum_t P(L/t) \sum_r P(r)\, P(t/r)$$

Join on r

Eliminate r

Join on t

Eliminate t



Inference by Enumeration takes more time and the size of table increases since it follows join join operation and then sum operation.

Inference by Variable Elimination takes polynomial time, since the size of table is less and it follows join sum join sum operation

Inference by Variable Elimination :-
(Inference by Variable Removal):

- The enumeration algorithm can be improved substantially by elimination repeated calculations
- The idea is simple: Do the calculation once and Solve the result for later use. This is a form of dynamic programming.
- previous equation

$$P(b|j,m) = \alpha \sum_e \sum_a P(b)P(e)P(a|b,e)P(j|a)P(m|a)$$

From this repeated variables are seperated:

$$P(b|j,m) = \alpha P(b) \sum_e P(e) \sum_a P(a|b,e) P(j|a) P(m|a)$$

$$= \underbrace{\alpha}_{B}\ \underbrace{P(b)}_{B}\ \underbrace{\sum_e}_{E}\ \underbrace{P(e)}_{E}\ \underbrace{\sum_a}_{A}\ \underbrace{P(a|b,e)}_{A}\ \underbrace{P(j|a)}_{J}\ \underbrace{P(m|a)}_{M}$$

we have annotated each part of the expression with the name of the associated variable, these parts are called factors.