UNTI – IV

- KNOWLEDGE AND REASONING

Knowledge Representation -Knowledge based agents – The Wumpus world – Propositional Logic - syntax, semantics and knowledge base building - inferences – reasoning patterns in propositional logic – predicate logic – representing facts in logic: Syntax and semantics – Unification – Unification Algorithm - Knowledge representation using rules - Knowledge representation using semantic nets - Knowledge representation using frames inferences - Uncertain Knowledge and reasoning Methods.

PREPARED BY
Dr N Saranya ,AP/CSE

Knowledge and Reasoning

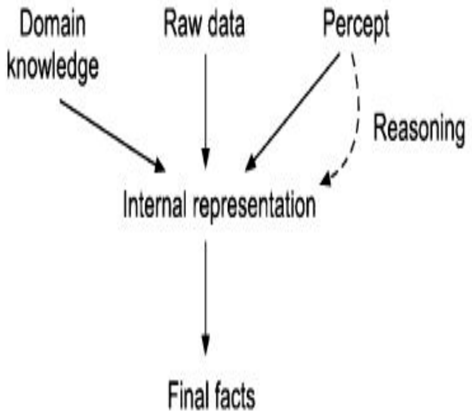
- systematic reasoning process is required to relate the events to the outcomes or to arrive at judgements .
- Reasoning –way to conclude on different aspects of problem based on available knowledge representation

  - Establish relationship among the available data and the final facts

Knowledge Representation:

-knowledge representation is about representation of the facts

Approaches and issues of knowledge Representation:



Approaches

- Adequacy representation
- Adequacy in terms of inferring-knowledge should be represented in such a way that there is way to manipulate the representative data in order to derive the new one
- Property of efficiency in term of acquisition
- Structure:
  - -simple relational knowledge structure

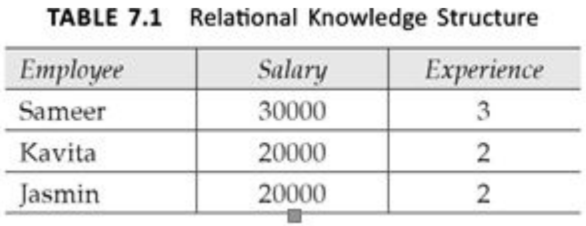


TABLE 7.1 Relational Knowledge Structure

Employee	*Salary*	*Experience*
Sameer	30000	3
Kavita	20000	2
Jasmin	20000	2

  - -inheritable knowledge structure
  - -slot –filler structure
  - Inferential knowledge structure –first order predicate logic
  - Procedural knowledge structure –when we need to have knowledge in detail form

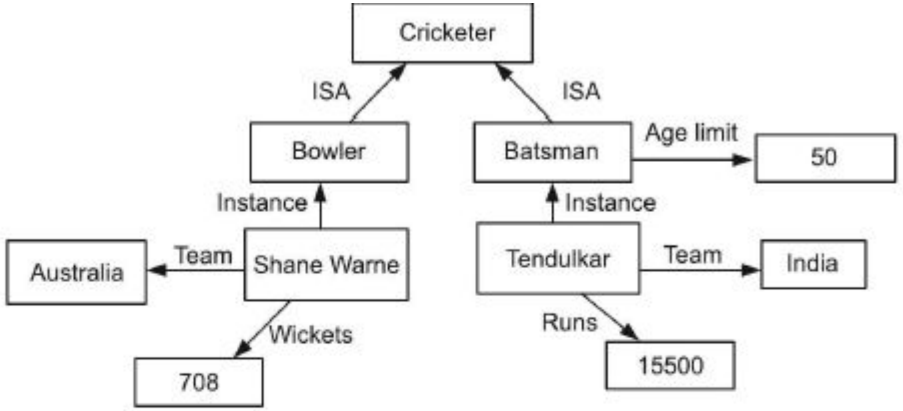


Figure 7.2 Inheritable knowledge.

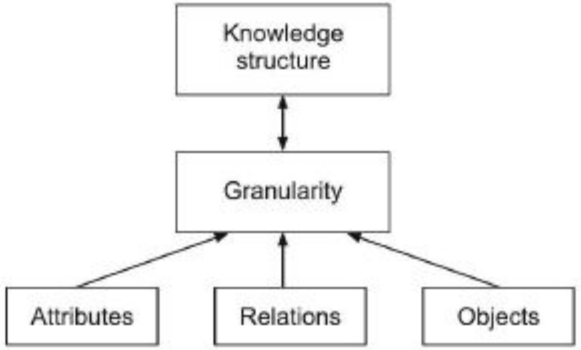
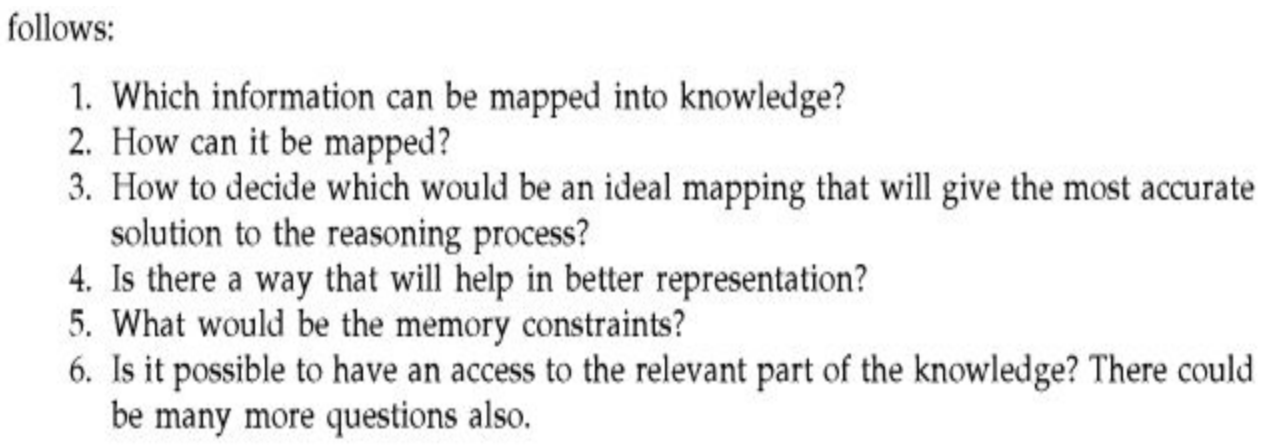


Figure 7.3 Knowledge structure hierarchy.

Issues of knowledge represetation



follows:

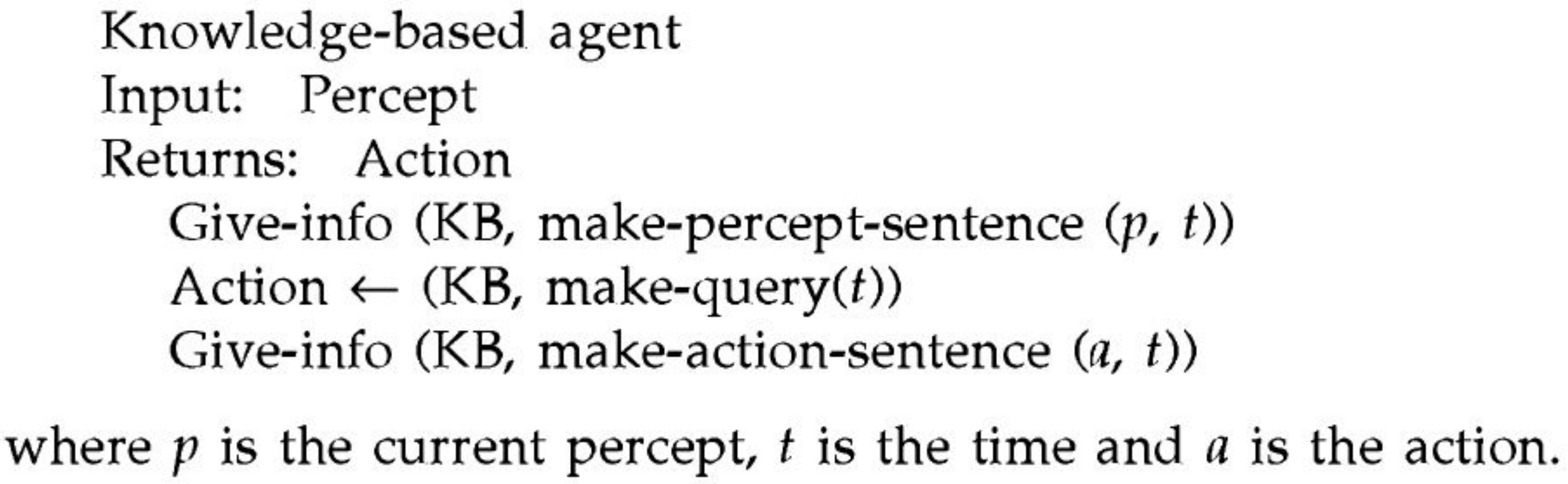
1. Which information can be mapped into knowledge?
2. How can it be mapped?
3. How to decide which would be an ideal mapping that will give the most accurate solution to the reasoning process?
4. Is there a way that will help in better representation?
5. What would be the memory constraints?
6. Is it possible to have an access to the relevant part of the knowledge? There could be many more questions also.

Knowledge representation and methods
1.Attributes
2.Relationship among the attributes
3.Granularity-what depth the mapping of knowledge is to be defined.
4.Reprsentation of the object as sets
5.Selection of correct structure

Knowledge based Agent

- Agent one who acts according to the environment

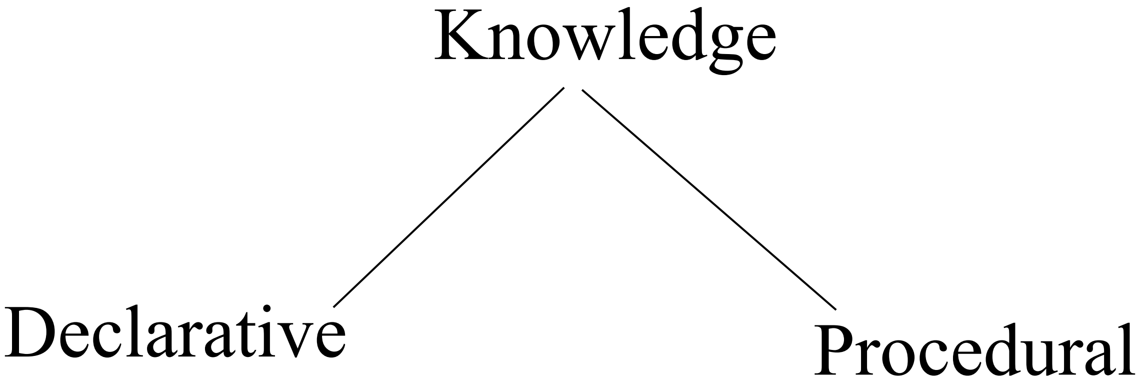
- Knowledge base is nothing but a representation of information role in deciding the actions.
- Action needs to be updated to the knowledge base



```
Knowledge-based agent
Input:    Percept
Returns:    Action
    Give-info (KB, make-percept-sentence ($p$, $t$))
    Action ← (KB, make-query($t$))
    Give-info (KB, make-action-sentence ($a$, $t$))
```

where p is the current percept, t is the time and a is the action.

Knowledge Representation & Reasoning



Knowledge

Declarative

Procedural

- Declarative knowledge deals with factoid questions (what is the capital of India? Etc.)
- Procedural knowledge deals with “How”
- Procedural knowledge can be embedded in declarative knowledge

Planning

Given a set of goals, construct a sequence of actions that achieves those goals:

- often very large search space
- but most parts of the world are independent of most other parts
- often start with goals and connect them to actions
- no necessary connection between order of planning and order of execution
- what happens if the world changes as we execute the plan and/or our actions don’t produce the expected results?

Learning

- If a system is going to act truly appropriately, then it must be able to change its actions in the light of experience:
    - how do we generate new facts from old ?
    - how do we generate new concepts ?
    - how do we learn to distinguish different situations in new environments ?

What is knowledge representation?

- Knowledge representation and reasoning (KR, KRR) is the part of Artificial intelligence which concerned with AI agents thinking and how thinking contributes to intelligent behavior of agents.
- It is responsible for representing information about the real world so that a computer can understand and can utilize this knowledge to solve the complex real world problems such as diagnosis a medical condition or communicating with humans in natural language.
- It is also a way which describes how we can represent knowledge in artificial intelligence. Knowledge representation is not just storing data into some database, but it also enables an intelligent machine to learn from that knowledge and experiences so that it can behave intelligently like a human.

What to Represent?

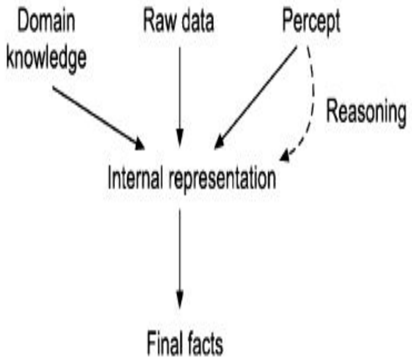
Following are the kind of knowledge which needs to be represented in AI systems:

- **Object:** All the facts about objects in our world domain. E.g., Guitars contains strings, trumpets are brass instruments.
- **Events:** Events are the actions which occur in our world.
- **Performance:** It describe behavior which involves knowledge about how to do things.
- **Meta-knowledge:** It is knowledge about what we know.
- **Facts:** Facts are the truths about the real world and what we represent.
- **Knowledge-Base:** The central component of the knowledge-based agents is the knowledge base. It is represented as KB. The Knowledgebase is a group of the Sentences (Here, sentences are used as a technical term and not identical with the English language).

Approaches to knowledge Representation

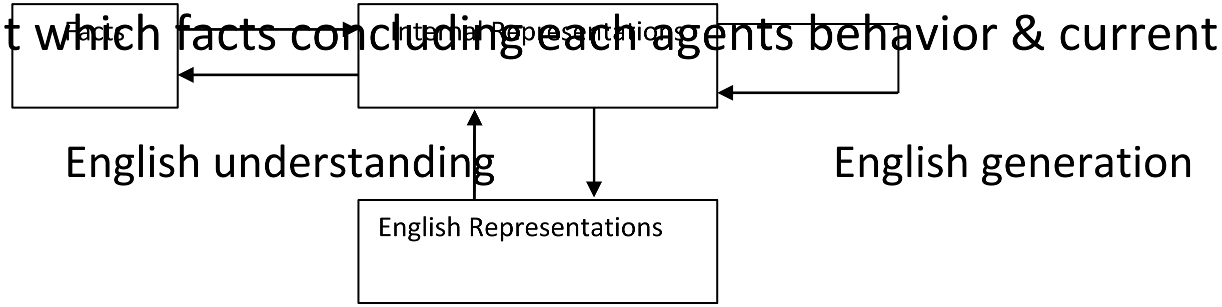
- Representational adequacy the ability to represent all of the kinds of knowledge that are needed in that domain.
- **Inferential Adequacy: -** the ability to manipulate the representation structures in such a way as to derive new structures corresponding to new knowledge inferred from ol.
- **Inferential Efficiency:** - the ability to incorporate into the knowledge structure additional information that can be used to focus the attention of the inference mechanism in the most promising directions.
- **Acquisitioned Efficiency: -** the ability to acquire new information easily. The simplest case involves direct insertion by a person of new knowledge into the database.

Knowledge Building and Representation



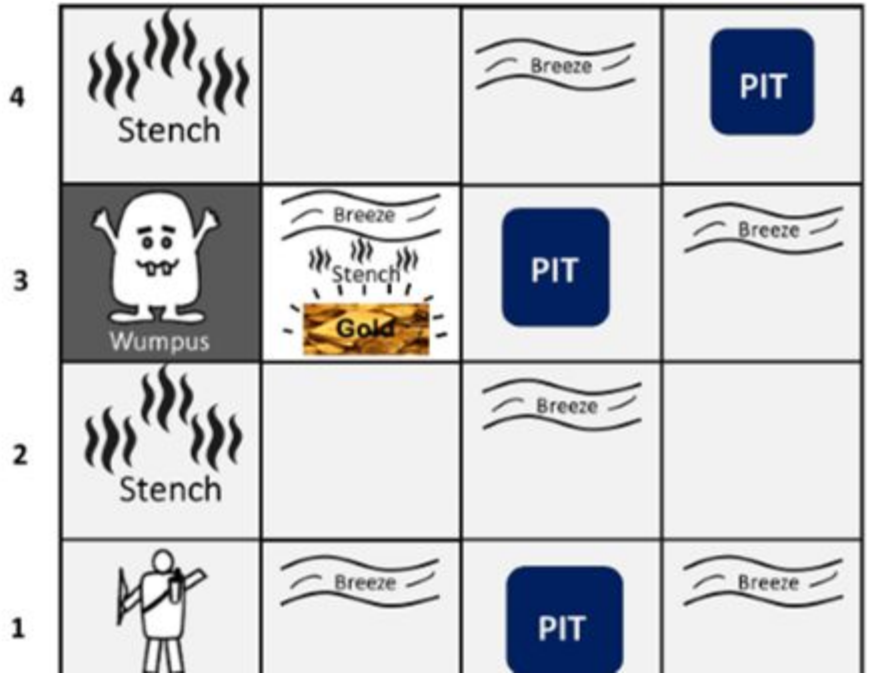
Knowledge Representation Issues

- It becomes clear that particular knowledge representation models allow for more specific more powerful problem solving mechanisms that operate on them.
- Examine specific techniques that can be used for representing & manipulating knowledge within programs.
- **<u>Representation & Mapping</u>**
- Facts :- truths in some relevant world
- These are the things we want to represent.
- Representations of facts in some chosen formalism.
- Things we are actually manipulating. Structuring these entities is as two levels.
- The knowledge level, at which facts concluding each agents behavior & current goals are described.



Wumpus world

The Wumpus world is a cave which has 4/4 rooms connected with passageways. So there are total 16 rooms which are connecte with each other. We have a knowledge-based agent who will go forward in this world. The cave has a room with a beast which i called Wumpus, who eats anyone who enters the room. The Wumpus can be shot by the agent, but the agent has a single arrow In the Wumpus world, there are some Pits rooms which are bottomless, and if agent falls in Pits, then he will be stuck there forever. The exciting thing with this cave is that in one room there is a possibility of finding a heap of gold. So the agent goal is t find the gold and climb out the cave without fallen into Pits or eaten by Wumpus. The agent will get a reward if he comes ou with gold, and he will get a penalty if eaten by Wumpus or falls in the pit.



1.The rooms adjacent to the Wumpus room are smelly, so that it would have some stench.
2.The room adjacent to PITs has a breeze, so if the agent reaches near to PIT, then he will perceive the breeze.
3.There will be glitter in the room if and only if the room has gold.
4.The Wumpus can be killed by the agent if the agent is facing to it, and Wumpus will emit a horrible scream which can be heard anywhere in the cave

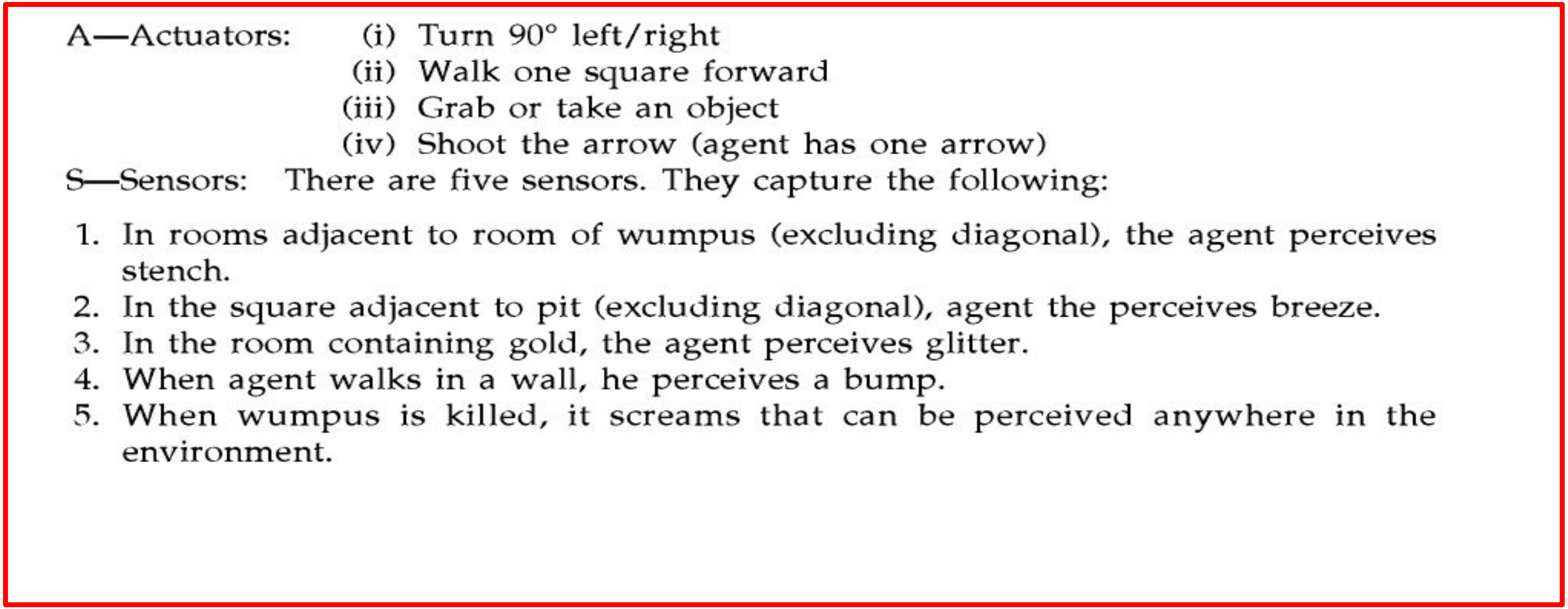
PEAS description of Wumpus world:

Performance measure:

- +1000 reward points if the agent comes out of the cave with the gold.
- -1000 points penalty for being eaten by the Wumpus or falling into the pit.
- -1 for each action, and -10 for using an arrow.
- The game ends if either agent dies or came out of the cave.

Environment:

- A 4*4 grid of rooms.
- The agent initially in room square [1, 1], facing toward the right.
- Location of Wumpus and gold are chosen randomly except the first square [1,1].
- Each square of the cave can be a pit with probability 0.2 except the first square.



A—Actuators: (i) Turn 90° left/right
(ii) Walk one square forward
(iii) Grab or take an object
(iv) Shoot the arrow (agent has one arrow)

S—Sensors: There are five sensors. They capture the following:

1. In rooms adjacent to room of wumpus (excluding diagonal), the agent perceives stench.
2. In the square adjacent to pit (excluding diagonal), agent the perceives breeze.
3. In the room containing gold, the agent perceives glitter.
4. When agent walks in a wall, he perceives a bump.
5. When wumpus is killed, it screams that can be perceived anywhere in the environment.

ARCHITECTURE OF A KNOWLEDGE-BASED AGENT

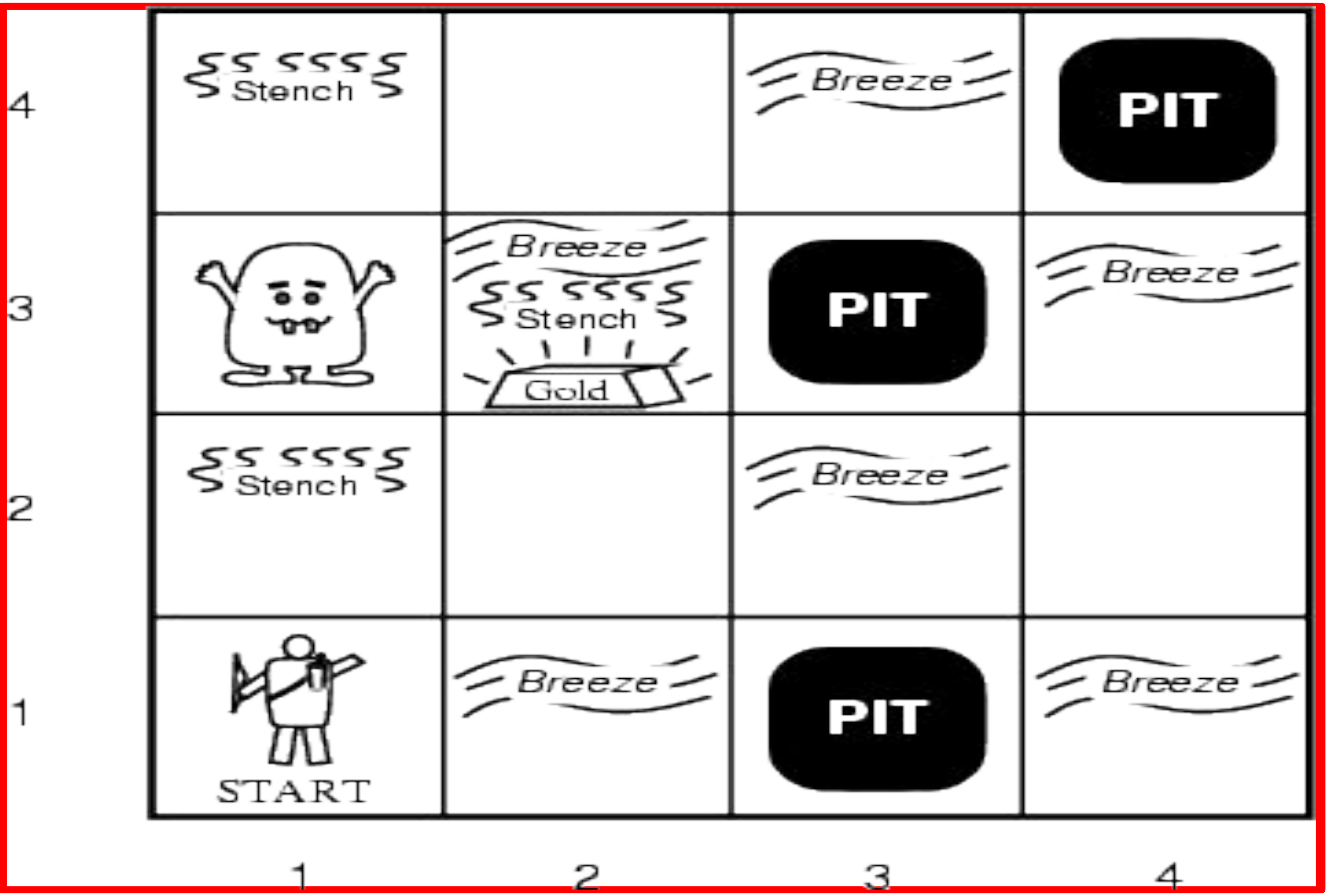
- **Knowledge Level.**
  - The most abstract level: describe agent by saying what it knows.
  - Example: A taxi agent might know that the Golden Gate Bridge connects San Francisco with the Marin County.
- **Logical Level.**
  - The level at which the knowledge is encoded into sentences.
  - Example: Links(GoldenGateBridge, SanFrancisco, MarinCounty).
- **Implementation Level.**
  - The physical representation of the sentences in the logical level.
  - Example: ‘(links goldengatebridge sanfrancisco marincounty)

THE WUMPUS WORLD ENVIRONMENT

- The Wumpus computer game
- The agent explores a cave consisting of rooms connected by passageways.
- Lurking somewhere in the cave is the Wumpus, a beast that eats any agent that enters its room.
- Some rooms contain bottomless pits that trap any agent that wanders into the room.
- Occasionally, there is a heap of gold in a room.
- The goal is to collect the gold and exit the world without being eaten

A TYPICAL WUMPUS WORLD

- The agent always starts in the field [1,1].
- The task of the agent is to find the gold, return to the field [1,1] and climb out of the cave.

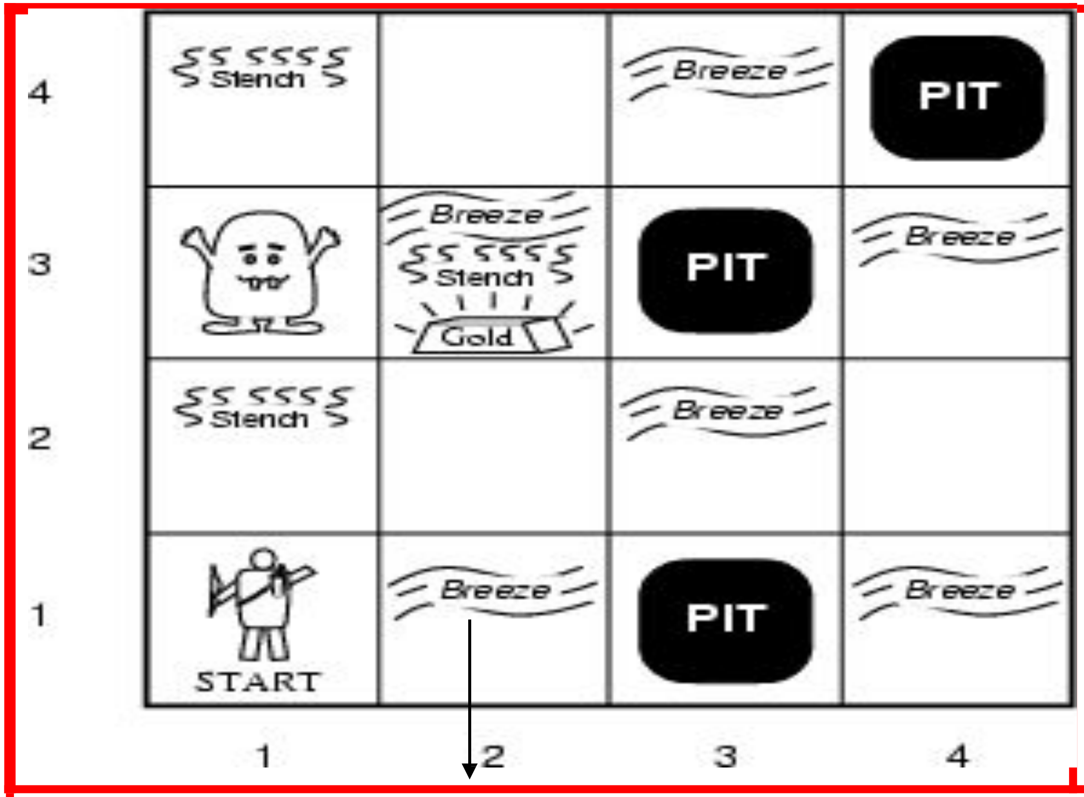


AGENT IN A WUMPUS WORLD: PERCEPTS

- The agent perceives
  - a stench in the square containing the Wumpus and in the adjacent squares (not diagonally)
  - a breeze in the squares adjacent to a pit
  - a glitter in the square where the gold is
  - a bump, if it walks into a wall
  - a woeful scream everywhere in the cave, if the wumpus is killed
- The percepts are given as a five-symbol list. If there is a stench and a breeze, but no glitter, no bump, and no scream, the percept is

  [Stench, Breeze, None, None, None]

EXPLORING A WUMPUS WORLD



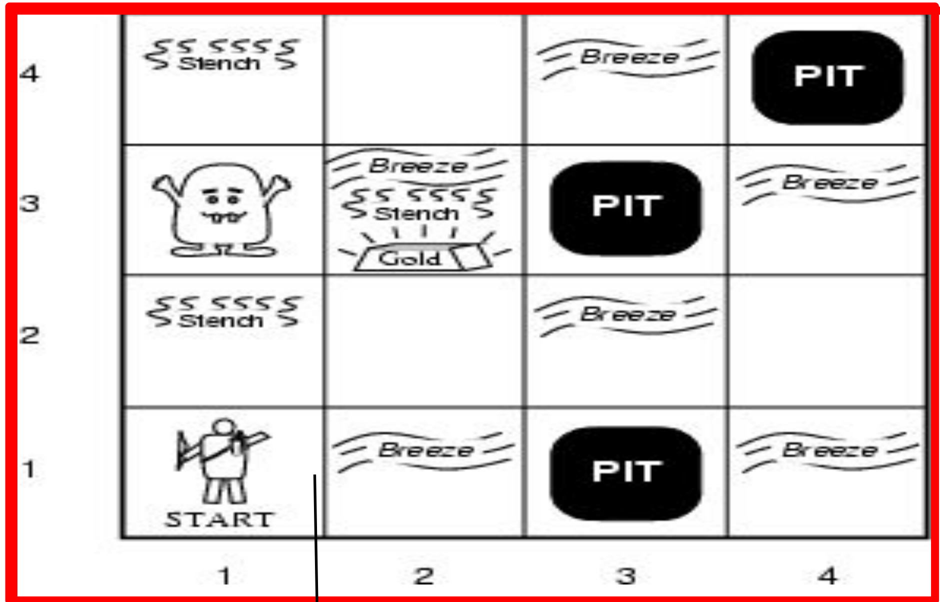
The knowledge base of the agent consists of the rules of the Wumpus world plus the percept “nothing” in [1,1]

Boolean percept feature values: <0, 0, 0, 0, 0>

None, none, none, none, none

Stench, Breeze, Glitter, Bump, Scream

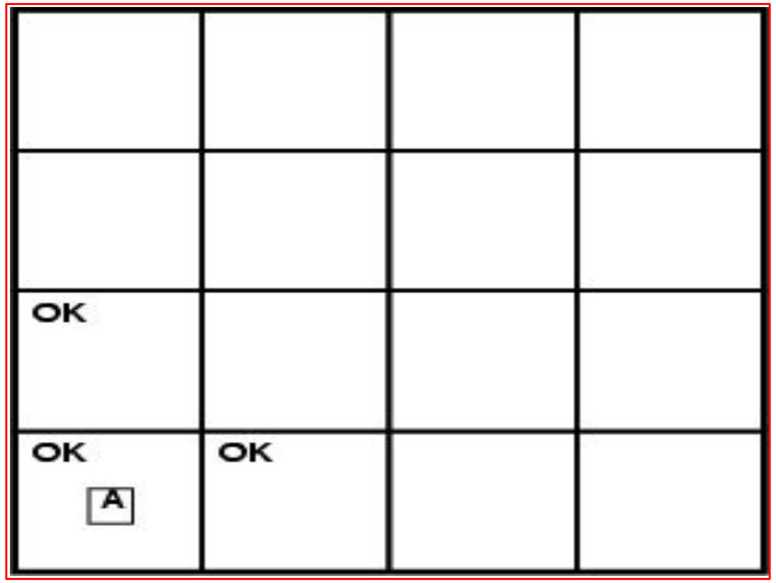
EXPLORING A WUMPUS WORLD



None, none, none, none, none

Stench, Breeze, Glitter, Bump, Scream

World “known” to agent at time = 0.

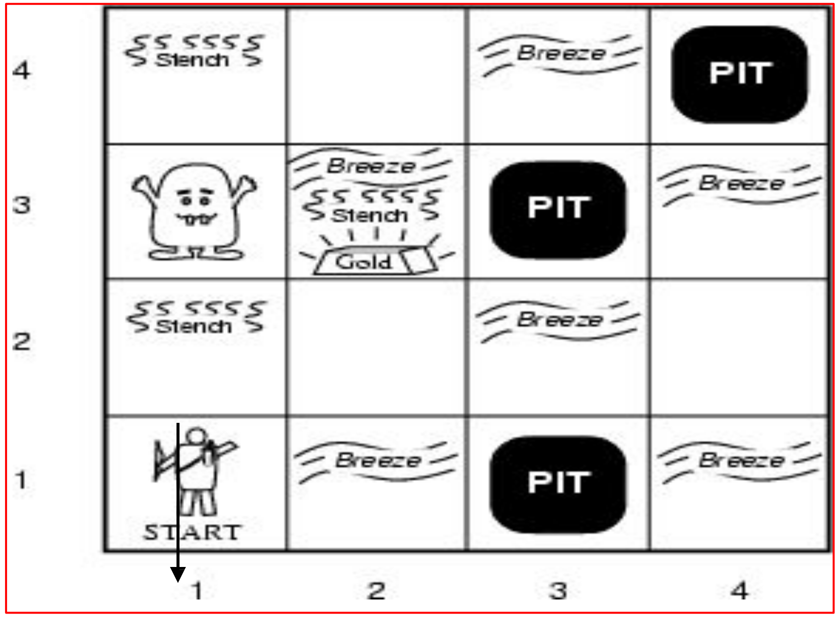


T=0 The KB of the agent consists of the rules of the Wumpus world plus the percept “nothing” in [1,1].

By inference, the agent’s knowledge base also has the information that [2,1] and [1,2] are okay. Added as propositions.

EXPLORING A WUMPUS WORLD

T = 0T = 1



None, none, none, none, none

Stench, Breeze, Glitter, Bump, Scream

Where next?

OK			
OK A	OK		

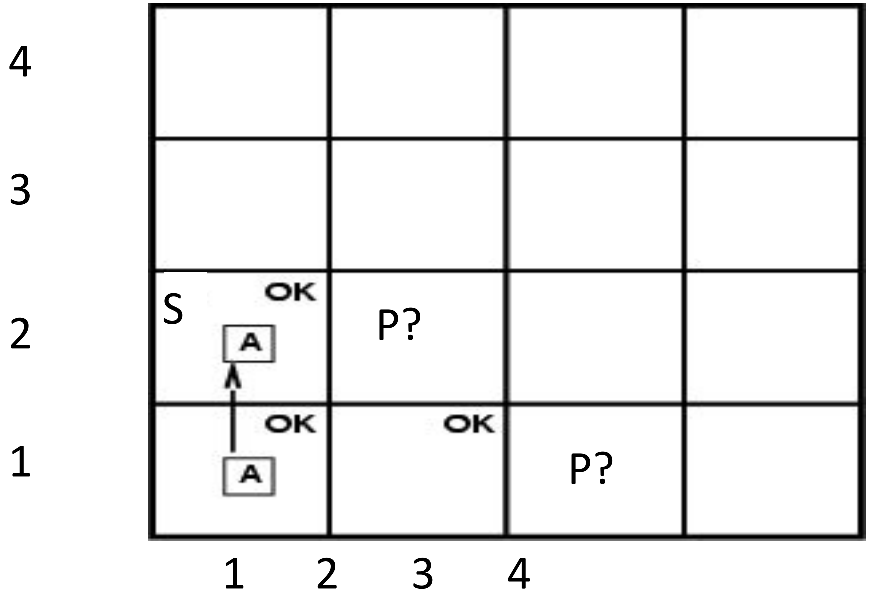
OK	P?		
OK V	OK A/B	P?	

None, breeze, none, none, none

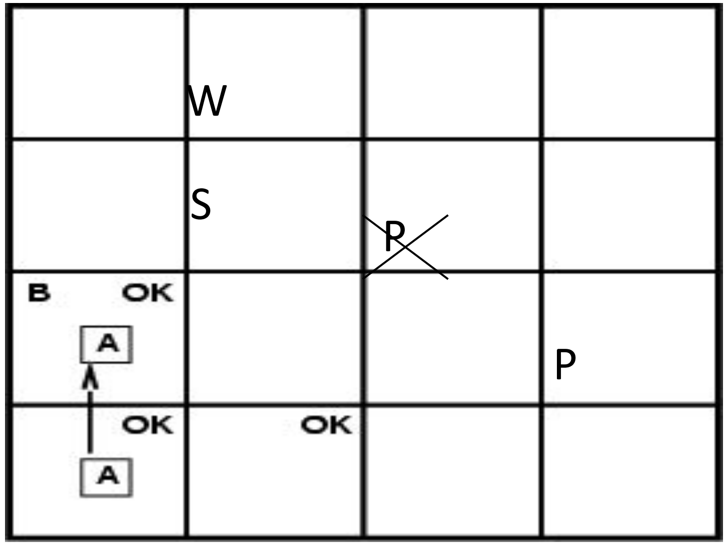
A – agent
V – visited
B - breeze

@ T = 1 What follows?
Pit(2,2) or Pit(3,1)

EXPLORING A WUMPUS WORLD



T=3



Stench, none, none, none, none

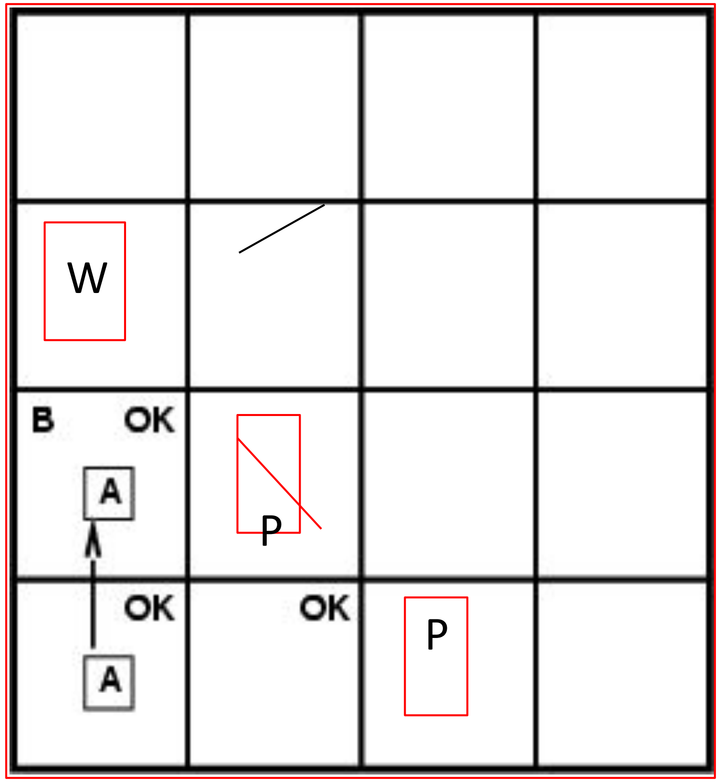
Stench, Breeze, Glitter, Bump, Scream

Where is Wumpus?

Wumpus cannot be in (1,1) or in (2,2) (Why?) Wumpus in (1,3)
Not breeze in (1,2) no pit in (2,2); but we know there is
pit in (2,2) or (3,1) pit in (3,1)

EXPLORING A WUMPUS WORLD

We reasoned about the possible states the Wumpus world can be in, given our percepts and our knowledge of the rules of the Wumpus world.
I.e., the content of KB at T=3.



What follows is what holds true in all those worlds that satisfy what is known at that time T=3 about the particular Wumpus world we are in.

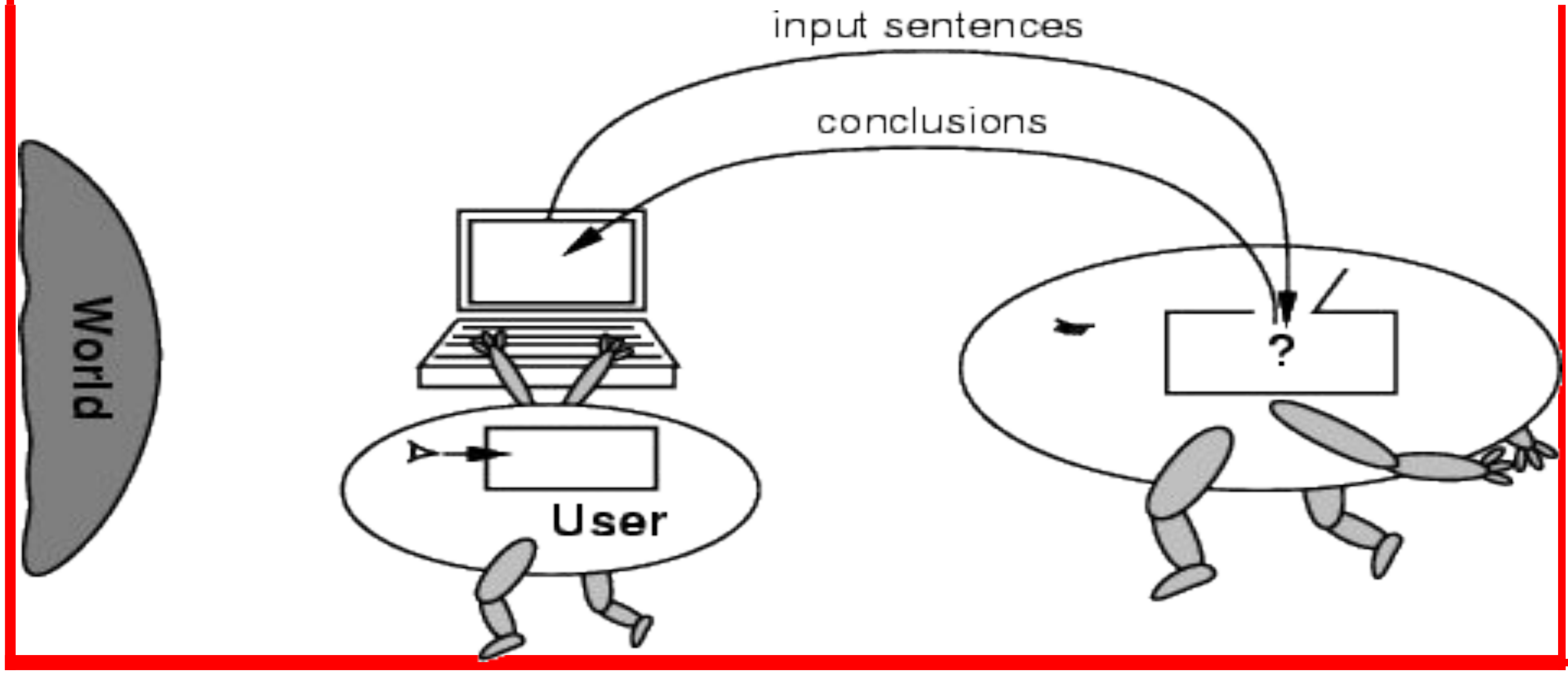
Example property: P_in_(3,1)

Models(KB) $\subseteq$ odels(P_in_(3,1))

Essence of logical reasoning:
Given *all we know*, Pit_in_(3,1) holds.
(“The world cannot be different.”)

NO INDEPENDENT ACCESS TO THE WORLD

- The reasoning agent often gets its knowledge about the facts of the world as a sequence of logical sentences and must draw conclusions only from them without independent access to the world.
- Thus it is very important that the agent's reasoning is sound!

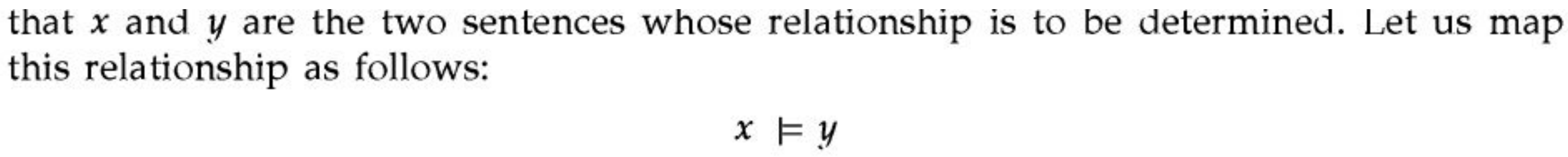


SUMMARY OF KNOWLEDGE BASED AGENTS

- Intelligent agents need knowledge about the world for making good decisions.
- The knowledge of an agent is stored in a knowledge base in the form of **sentences** in a knowledge representation language.
- A knowledge-based agent needs a **knowledge base** and an **inference mechanism**. It operates by storing sentences in its knowledge base, inferring new sentences with the inference mechanism, and using them to deduce which actions to take.
- A **representation language** is defined by its syntax and semantics, which specify the structure of sentences and how they relate to the facts of the world.
- The **interpretation** of a sentence is the fact to which it refers. If this fact is part of the actual world, then the sentence is true.

What is a Logic?

- Logic basically deals with study of principles for reasoning
- How the logic is built or rather how the
- A language with concrete rules
  - No ambiguity in representation (may be other errors!)
  - Allows unambiguous communication and processing
  - Very unlike natural languages e.g. English
- Many ways to translate between languages
  - A statement can be represented in different logics
  - And perhaps differently in same logic
- **Expressiveness** of a logic-How much can we say in this language?
- Not to be confused with logical reasoning



that x and y are the two sentences whose relationship is to be determined. Let us map this relationship as follows:

$$x \models y$$

Logic-wumpus world

Wumpus world ,when the percept are combined with facts or rules ,then the combination constitutes the knowledge base(KB).it cannot be judged to check whether the pit exist in [2,2].

This type of inferring is called model checking.

So in the logical inferring ,there is notation of truth that is to be maintained .Even it needs to have property of completeness.

7.5 Propositional Logic

- It's a mathematical model that provides reasoning regarding the logic value of expressions
- Propositional logic is also called sentential logic. Propositional logic is the fundamental logic
- Syntax
  - Propositions, e.g. "it is wet"
  - Connectives: and, or, not, implies, iff (equivalent)

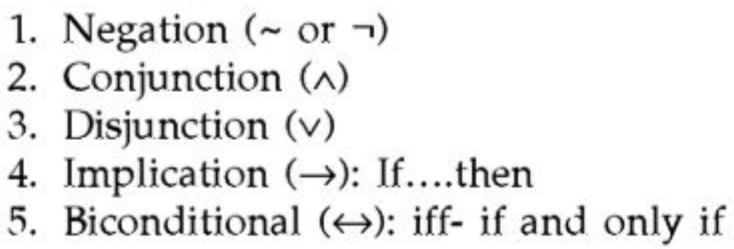
    $\wedge \quad \vee \quad \neg \quad \rightarrow \quad \leftrightarrow$
  - Brackets, T (true) and F (false)
- Semantics (Classical AKA Boolean)
  - Define how connectives affect truth
    - "P and Q" is true if and only if P is true and Q is true
  - Use **truth tables** to work out the truth of statements

7.5 Propositional logic-syntax

- Syntax
  - Rules for constructing legal sentences in the logic
  - Which symbols we can use (English: letters, punctuation)
  - How we are allowed to combine symbols
- Semantics
  - How we interpret (read) sentences in the logic
  - Assigns a meaning to each sentence
- Example: “All lecturers are seven foot tall”
  - A valid sentence (syntax)
  - And we can understand the meaning (semantics)
  - This sentence happens to be false (there is a counterexample)

7.5 Propositional logic-syntax

- Syntax
  - Rules for constructing legal sentences in the logic
  - Which symbols we can use (English: letters, punctuation)
  - How we are allowed to combine symbols
  - 2 types of sentences -1.simple 2.compound
  - Atomic sentences consists of single propositional symbol
  - Symbols essentially can be true or false
  - The five operators are

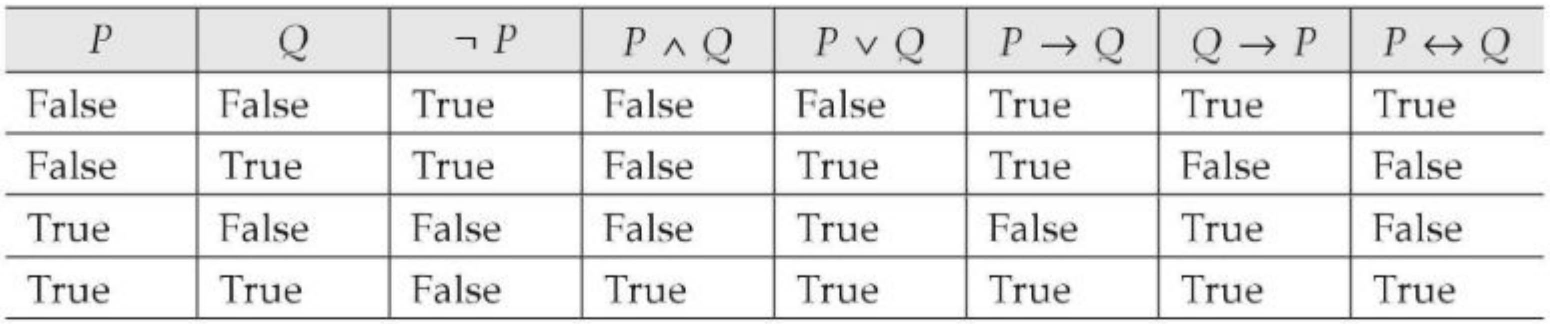


1. Negation (~ or ¬)
2. Conjunction (∧)
3. Disjunction (∨)
4. Implication (→): If....then
5. Biconditional (↔): iff- if and only if

7.5 Propositional logic-Semantics

- Semantics
  - It tells about the rules to determine the truth of a sentences
  - Assigns a meaning to each sentence
  - Things are simple when it comes to simple sentences .
  - So with 2 propositional symbols playing role ,models will be 2^2

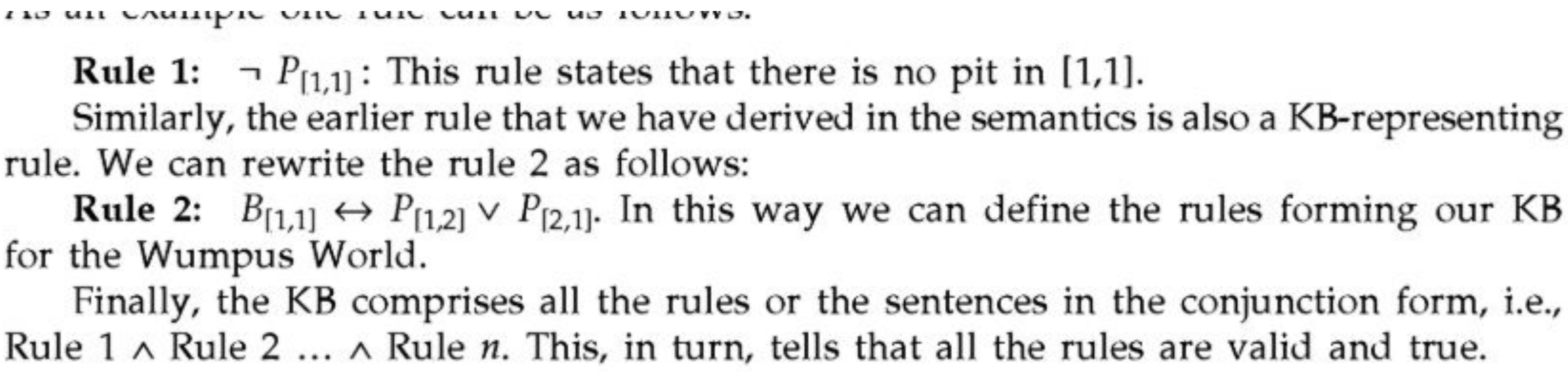
TABLE 7.2 Truth Tables



P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \rightarrow Q$	$Q \rightarrow P$	$P \leftrightarrow Q$
False	False	True	False	False	True	True	True
False	True	True	False	True	True	False	False
True	False	False	False	True	False	True	False
True	True	False	True	True	True	True	True

7.5 Building a knowledge base (KB)

- Example:Wumpus world
- First step is to decide the propositions. Then construct the rule with the operators and the true /false operators .
- Consider the i,j represent he room or grid value .
- P[I,j]=proposition that is true when there is pit in i,j
- b[I,j]=proposition that is true when there is breeze in i,j
- The KB comprises the rules that are built with propositions and operators



As an example one rule can be as follows.

Rule 1: $\neg P_{[1,1]}$: This rule states that there is no pit in [1,1].

Similarly, the earlier rule that we have derived in the semantics is also a KB-representing rule. We can rewrite the rule 2 as follows:

Rule 2: $B_{[1,1]} \leftrightarrow P_{[1,2]} \vee P_{[2,1]}$. In this way we can define the rules forming our KB for the Wumpus World.

Finally, the KB comprises all the rules or the sentences in the conjunction form, i.e., Rule 1 $\wedge$ Rule 2 … $\wedge$ Rule *n*. This, in turn, tells that all the rules are valid and true.

7.5 Inference-tautology,contradiction

After KB represented decide upon the inference

Different values which the propositions take in the compound statement .so inferring we should decide whether KB

Recursive enumeration

Given :KB list of symbols in KB and x

Check -1 .if symbols are empty then

check if model is consistent with KB

If true then check if x evaluates to true

Else it is inconsistent

2 Else

recursively construct conjunction

For partial models with symbols in KB and x

Some concepts:Tautology,Contradiction and satisfiability

1.Tautology:A tautology states that the sentences is always true in all models. Its also called validity (pV⌐p)

2.Contradiction:proposition always false.(p^ ⌐p)

3.Satisfiability:A sentences or proposition is satisfiable if it is true for some model .

7.5 Refutation

Explore all the models and check them till we get the one that satisfies the sentences .understanding the relation between validity and satisfiability can be explained as follows

X |=y iff sentence (x ^ ¬y) is unsatisfiable

Proving y from x by checking the unsatisfiability as proof by refutation or contradiction.

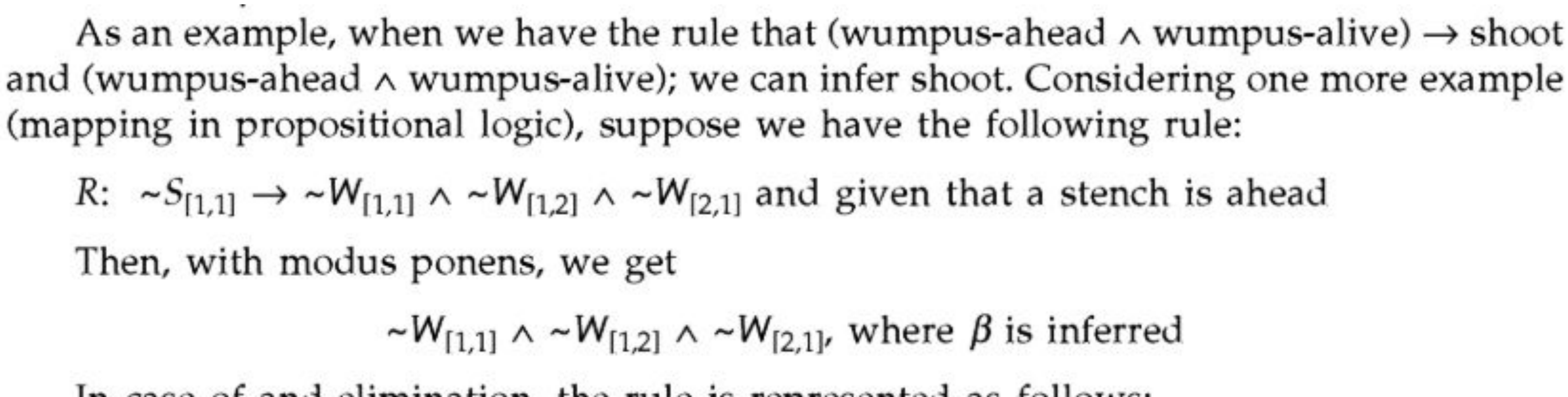
Reasoning patterns in propositional logic:

Reasoning patterns we use and apply the basic rules in order to derive chains of conclusions.

These rules are also called patterns of inference .

Two most commonly used rules are modus ponens and elimination

$\alpha \rightarrow \beta,\ \alpha \mapsto \beta$



As an example, when we have the rule that (wumpus-ahead ∧ wumpus-alive) → shoot and (wumpus-ahead ∧ wumpus-alive); we can infer shoot. Considering one more example (mapping in propositional logic), suppose we have the following rule:

R: $\sim S_{[1,1]} \rightarrow \sim W_{[1,1]} \wedge \sim W_{[1,2]} \wedge \sim W_{[2,1]}$ and given that a stench is ahead

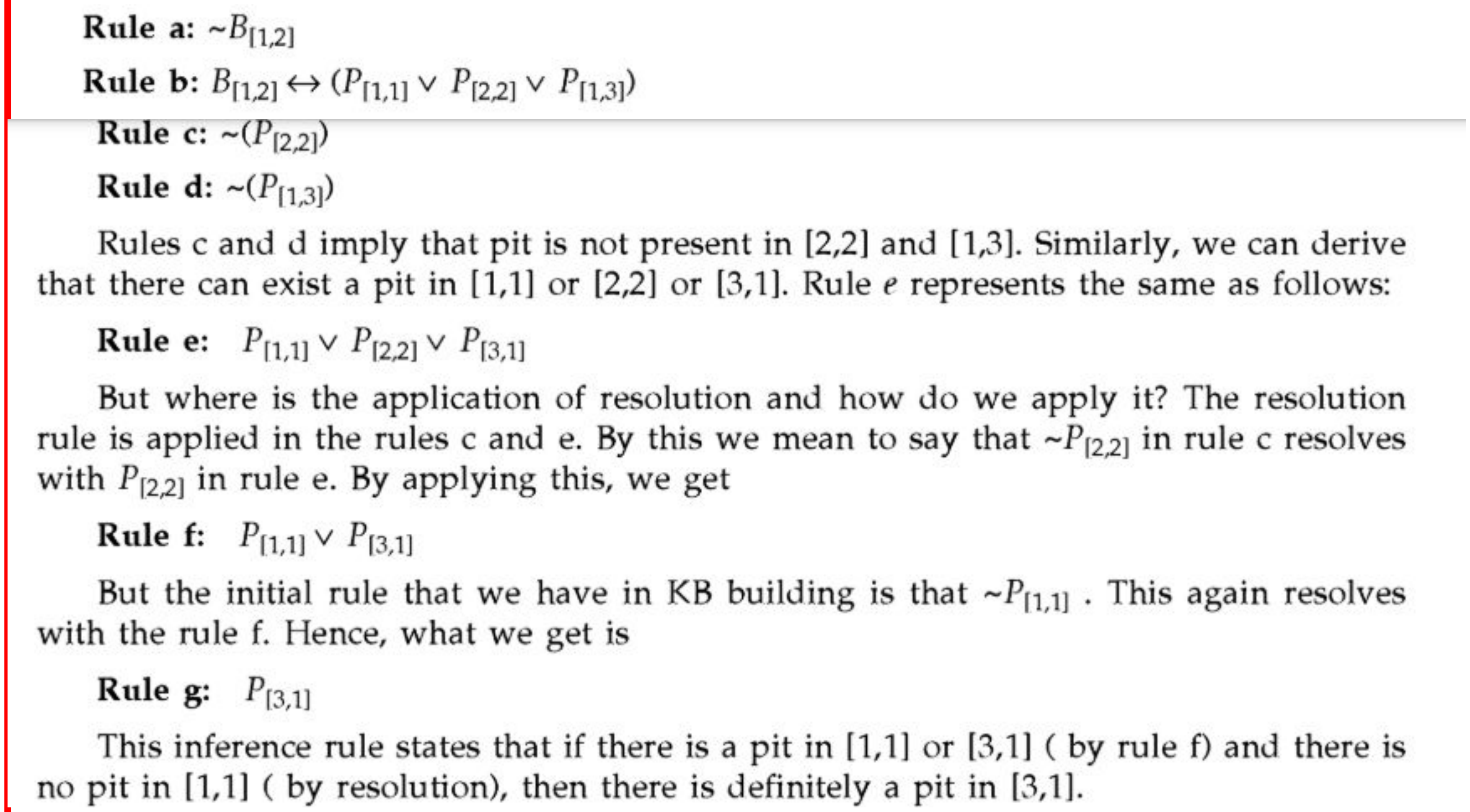
Then, with modus ponens, we get

$$\sim W_{[1,1]} \wedge \sim W_{[1,2]} \wedge \sim W_{[2,1]},\ \text{where } \beta \text{ is inferred}$$

In case of and elimination, the rule is represented as follows:

7.5 Resolution

Single inference rule which gives a complete inference algorithm



Rule a: $\sim B_{[1,2]}$

Rule b: $B_{[1,2]} \leftrightarrow (P_{[1,1]} \vee P_{[2,2]} \vee P_{[1,3]})$

Rule c: $\sim(P_{[2,2]})$

Rule d: $\sim(P_{[1,3]})$

Rules c and d imply that pit is not present in [2,2] and [1,3]. Similarly, we can derive that there can exist a pit in [1,1] or [2,2] or [3,1]. Rule *e* represents the same as follows:

Rule e: $P_{[1,1]} \vee P_{[2,2]} \vee P_{[3,1]}$

But where is the application of resolution and how do we apply it? The resolution rule is applied in the rules c and e. By this we mean to say that $\sim P_{[2,2]}$ in rule c resolves with $P_{[2,2]}$ in rule e. By applying this, we get

Rule f: $P_{[1,1]} \vee P_{[3,1]}$

But the initial rule that we have in KB building is that $\sim P_{[1,1]}$. This again resolves with the rule f. Hence, what we get is

Rule g: $P_{[3,1]}$

This inference rule states that if there is a pit in [1,1] or [3,1] (by rule f) and there is no pit in [1,1] (by resolution), then there is definitely a pit in [3,1].

7.5 Conjective normal form(CNF)

Resolution algorithm

1.Convert (KB ^ -y)into CNF

2.We get some resulting clauses

3.The resolution rule is to be applied to each clause

4.The complementing pairs,are resolved to generate a new rule or a clause.

5.Add this into to the KB if not already present

6.Goto step 3,till any of the following two conditions occur:

(i) No new clauses can be added in which KB doesn't entail à

(ii)applying the resolution yields an empty rule, indicating KB entails à

7.5 Forward and backward chaining

Forward and backward chaining are the algorithms that are used for inferring.

Forward chaining :

Process start from the known facts

This process carried out till we reach the query

The approach is mapped to an AND-OR graph .

1.Start with the known facts

2.if the premises of the implications in the clause are known ,then add conclusion

3.Go to step 2,

i.we infer the values as true or false for the query

ii.No inference can occur

Backward chaining:

Process starts with the query.so move from goal to infer the facts .This method is also called goal-driven method

PREDICATE LOGIC

Its also called first order logic.

It allows to describe the objects involved and their relationships.

Consider the following example .

All   kids are naughty

Suzy is a kid

Then ,Suzy is naughty.

Its an powerful tool .

Its represented using the following ways

- Repersenting facts in logic:syntax and semantics
- Instance representation and ISA Relationship
- Comparision of predicate and propositional logic

Representing facts in logic: syntax and semantics

Any sentences has a subject and a predicate.

Ex:The car is red
Subject:the car
Predicate:red
We can represent in the following way:
Atomic sentences->predicate(terms)
The connectives and the quantifiers are used in the formulation of the sentences
Connectives:
Quantifiers:
Simple sentences:sita is the mother of Rohan
Representation:Mother (Sita,Rohan)

Complex sentences: Reeta uncle and Rohan’s daddy booked a flat
Representation:Booked(uncle(Reeta),Daddy(Rohan))

First Order Logic

- More expressive logic than propositional
  - Used in this course (Lecture 6 on representation in FOL)
- **Constants** are objects: john, apples
- **Predicates** are properties and relations:
  - likes(john, apples)
- **Functions** transform objects:
  - likes(john, fruit_of(apple_tree))
- **Variables** represent any object: likes(X, apples)
- **Quantifiers** qualify values of variables
  - True for all objects (Universal): ∀X. likes(X, apples)
  - Exists at least one object (Existential): ∃X. likes(X, apples)

First Order Logic

- More expressive logic than propositional
  - Used in this course (Lecture 6 on representation in FOL)
- **Constants** are objects: john, apples
- **Predicates** are properties and relations:
  - likes(john, apples)
- **Functions** transform objects:
  - likes(john, fruit_of(apple_tree))
- **Variables** represent any object: likes(X, apples)
- **Quantifiers** qualify values of variables
  - True for all objects (Universal): ∀X. likes(X, apples)
  - Exists at least one object (Existential): ∃X. likes(X, apples)

First Order Logic

- More expressive logic than propositional
  - Used in this course (Lecture 6 on representation in FOL)
- **Constants** are objects: john, apples
- **Predicates** are properties and relations:
  - likes(john, apples)
- **Functions** transform objects:
  - likes(john, fruit_of(apple_tree))
- **Variables** represent any object: likes(X, apples)
- **Quantifiers** qualify values of variables
  - True for all objects (Universal): ∀X. likes(X, apples)
  - Exists at least one object (Existential): ∃X. likes(X, apples)

Example: FOL Sentence

- “Every rose has a thorn”

$$\forall X.(rose(X) \rightarrow \exists Y.(has(X,Y) \wedge thorn(Y)))$$

- For all X
  - if (X is a rose)
  - then there exists Y
    - (X has Y) and (Y is a thorn)

Example: FOL Sentence

- “On Mondays and Wednesdays I go to John’s house for dinner”

$$\forall X.((is\_mon(X) \vee is\_wed(X)) \longrightarrow eat\_meal(me, houseOf(john), X))$$

- Note the change from “and” to “or”
  - Translating is problematic

Higher Order Logic

- More expressive than first order
- Functions and predicates are also objects
  - Described by predicates: binary(addition)
  - Transformed by functions: differentiate(square)
  - Can quantify over both
- E.g. define red functions as having zero at 17

$$\forall F.(red(F) \leftrightarrow F(0) = 17)$$

- Much harder to reason with

Beyond True and

- Multi-valued logics
  - More than two truth values
  - e.g., true, false & unknown
  - **Fuzzy logic** uses probabilities, truth value in [0,1]
- Modal logics
  - Modal operators define mode for propositions
  - **Epistemic logics** (belief)
    - e.g. □p (necessarily p), ◊p (possibly p), …
  - **Temporal logics** (time)
    - e.g. □p (always p), ◊p (eventually p), …

Propositional logic

- Propositional logic consists of:
  - The logical values true and false (T and F)
  - Propositions: "Sentences," which
    - Are atomic (that is, they must be treated as indivisible units, with no internal structure), and
    - Have a single logical value, either true or false
  - Operators, both unary and binary; when applied to logical values, yield logical values
    - The usual operators are and, or, not, and implies

Truth tables

- Logic, like arithmetic, has operators, which apply to one, two, or more values (operands)
- A truth table lists the results for each possible arrangement of operands
  - Order is important: ***x op y*** may or may not give the same result as ***y op x***
- The rows in a truth table list all possible sequences of truth values for n operands, and specify a result for each sequence
  - Hence, there are 2^n rows in a truth table for n operands

Unary operators

- There are four possible unary operators:

X	Constant true, (T)
T	T
F	T

X	Identity, (X)
T	T
F	F

X	Constant false, (F)
T	F
F	F

X	**Negation,** ¬X
T	F
F	T

- Only the last of these (negation) is widely used (and has a symbol,¬ ,for the operation

Combined tables for unary operators

X	Constant T	Constant F	Identity	¬X
T	T	F	T	F
F	T	F	F	T

Binary operators

- There are sixteen possible binary operators:

X	Y																
T	T	T	T	T	T	T	T	T	T	F	F	F	F	F	F	F	F
T	F	T	T	T	T	F	F	F	F	T	T	T	T	F	F	F	F
F	T	T	T	F	F	T	T	F	F	T	T	F	F	T	T	F	F
F	F	T	F	T	F	T	F	T	F	T	F	T	F	T	F	T	F

- All these operators have names, but I haven’t tried to fit them in
- Only a few of these operators are normally used in logic

Useful binary operators

- Here are the binary operators that are traditionally used:

X	Y	AND X ∧ Y	OR X ∨ Y	IMPLIES X ⇒ Y	BICONDITIONAL X ⇔ Y
T	T	T	T	T	T
T	F	F	T	F	F
F	T	F	T	T	F
F	F	F	F	T	T

- Notice in particular that material implication (⇒) only approximately means the same as the English word "implies"
- All the other operators can be constructed from a combination of these (along with unary not ¬)

Logical expressions

- All logical expressions can be computed with some combination of and ($\land$),
  or ($\lor$), and not ($\neg$) operators

- For exar y:

X	Y	$\neg X$	$\neg X \lor Y$	$X \Rightarrow Y$
T	T	F	T	T
T	F	F	F	F
F	T	T	T	T
F	F	T	T	T

- Notice that $\neg X \lor Y$ is equivalent to $X \Rightarrow Y$

Another example

- Exclusive or (xor) is true if exactly one of its operands is true

X	Y	¬X	¬Y	¬X ∧ Y	X ∧ ¬Y	(¬X∧Y)∨(X∧¬Y)	X xor Y
T	T	F	F	F	F	F	F
T	F	F	T	F	T	T	T
F	T	T	F	T	F	T	T
F	F	T	T	F	F	F	F

- Notice that (¬X∧Y)∨(X∧¬Y) is equivalent to X xor Y

World

- A world is a collection of prepositions and logical expressions relating those prepositions
- Example:
  - Propositions: JohnLovesMary, MaryIsFemale, MaryIsRich
  - Expressions:

    MaryIsFemale ∧ MaryIsRich ⇒ JohnLovesMary
- A proposition "says something" about the world, but since it is atomic (you can't look inside it to see component parts), propositions tend to be very specialized and inflexible

Models

A model is an assignment of a truth value to each proposition, for example:

- JohnLovesMary: T, MaryIsFemale: T, MaryIsRich: F

- An expression is satisfiable if there is a model for which the expression is true
  - For example, the above model satisfies the expression

    MaryIsFemale ∧ MaryIsRich ⇒ JohnLovesMary

- An expression is valid if it is satisfied by *every* model
  - This expression is *not* valid:

    MaryIsFemale ∧ MaryIsRich ⇒ JohnLovesMary

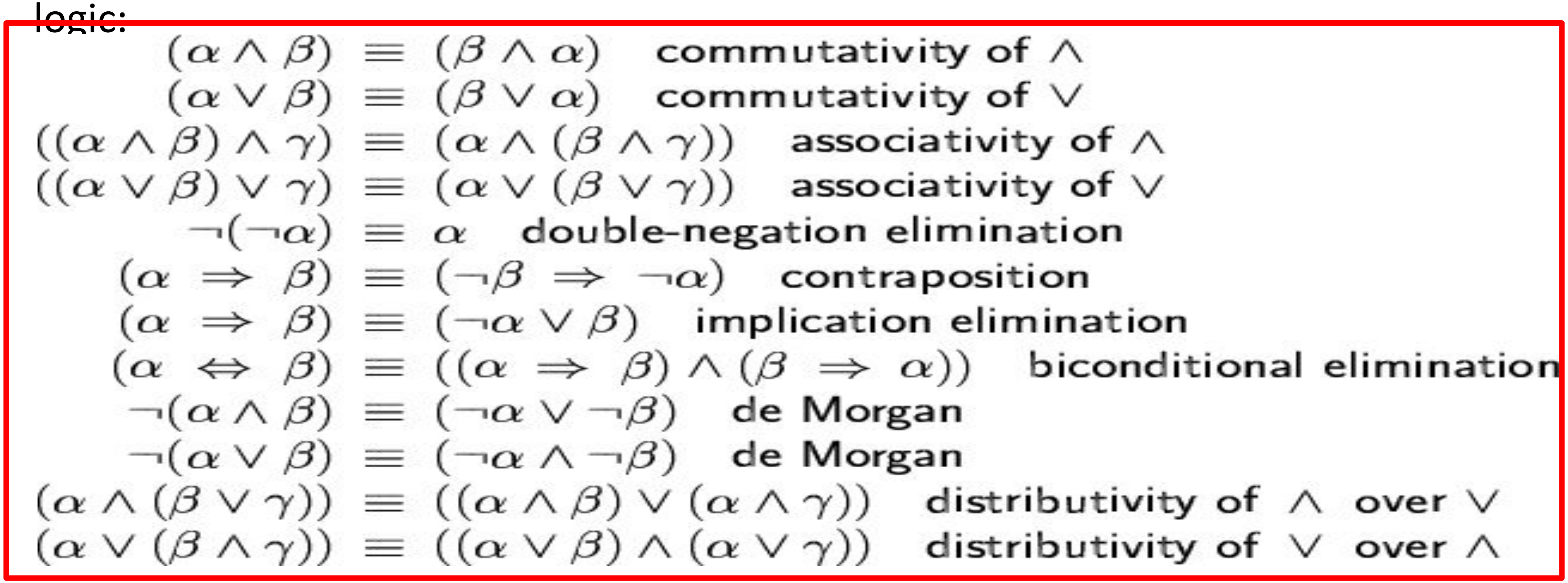
    because it is not satisfied by this model:

    JohnLovesMary: F, MaryIsFemale: T, MaryIsRich: T
  - But this expression *is* valid:

    MaryIsFemale ∧ MaryIsRich ⇒ MaryIsFemale

Inference rules in propositional logic

- Here are just a few of the rules you can apply when reasoning in propositional logic:



$(\alpha \wedge \beta)$	$\equiv$	$(\beta \wedge \alpha)$	commutativity of $\wedge$
$(\alpha \vee \beta)$	$\equiv$	$(\beta \vee \alpha)$	commutativity of $\vee$
$((\alpha \wedge \beta) \wedge \gamma)$	$\equiv$	$(\alpha \wedge (\beta \wedge \gamma))$	associativity of $\wedge$
$((\alpha \vee \beta) \vee \gamma)$	$\equiv$	$(\alpha \vee (\beta \vee \gamma))$	associativity of $\vee$
$\neg(\neg\alpha)$	$\equiv$	α	double-negation elimination
$(\alpha \Rightarrow \beta)$	$\equiv$	$(\neg\beta \Rightarrow \neg\alpha)$	contraposition
$(\alpha \Rightarrow \beta)$	$\equiv$	$(\neg\alpha \vee \beta)$	implication elimination
$(\alpha \Leftrightarrow \beta)$	$\equiv$	$((\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha))$	biconditional elimination
$\neg(\alpha \wedge \beta)$	$\equiv$	$(\neg\alpha \vee \neg\beta)$	de Morgan
$\neg(\alpha \vee \beta)$	$\equiv$	$(\neg\alpha \wedge \neg\beta)$	de Morgan
$(\alpha \wedge (\beta \vee \gamma))$	$\equiv$	$((\alpha \wedge \beta) \vee (\alpha \wedge \gamma))$	distributivity of $\wedge$ over $\vee$
$(\alpha \vee (\beta \wedge \gamma))$	$\equiv$	$((\alpha \vee \beta) \wedge (\alpha \vee \gamma))$	distributivity of $\vee$ over $\wedge$

Implication elimination

- A particularly important rule allows you to get rid of the implication operator, ⇒ :
  - $X \Rightarrow Y \equiv \neg X \vee Y$
- We will use this later on as a necessary tool for simplifying logical expressions
- The symbol ≡ means “is logically equivalent to”

Conjunction elimination

- Another important rule for simplifying logical expressions allows you to get rid of the conjunction (**and**) operator, ∧ :
- This rule simply says that if you have an **and** operator at the top level of a fact (logical expression), you can break the expression up into two separate facts:
  - MaryIsFemale ∧ MaryIsRich
    - becomes:
  - MaryIsFemale
  - MaryIsRich

Inference by computer

- To do inference (reasoning) by computer is basically a *search* process, taking logical expressions and applying inference rules to them
  - Which logical expressions to use?
  - Which inference rules to apply?
- Usually you are trying to “prove” some particular statement
- Example:
  - it_is_raining ∨ it_is_sunny
  - it_is_sunny ⇒ I_stay_dry
  - it_is_rainy ⇒ I_take_umbrella
  - I_take_umbrella ⇒ I_stay_dry
  - To prove:

I_stay_dry

7.8 REPRESENTING KNOWLEDGE USING RULES

- Declarative and procedural knowledge
- Logic programming
- Forward and backward reasoning
- Matching-indexing

  -matching with variable

  -approximate and complex matching

  -conflict resolution

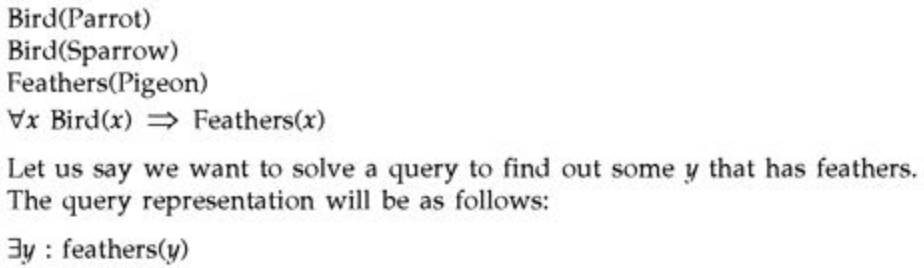
Declarative and procedural knowledge

there are 2 types of representations

declarations –Knowledge is specified

Procedural- Knowledge is embedded.

It can have heuristics too to have generated.

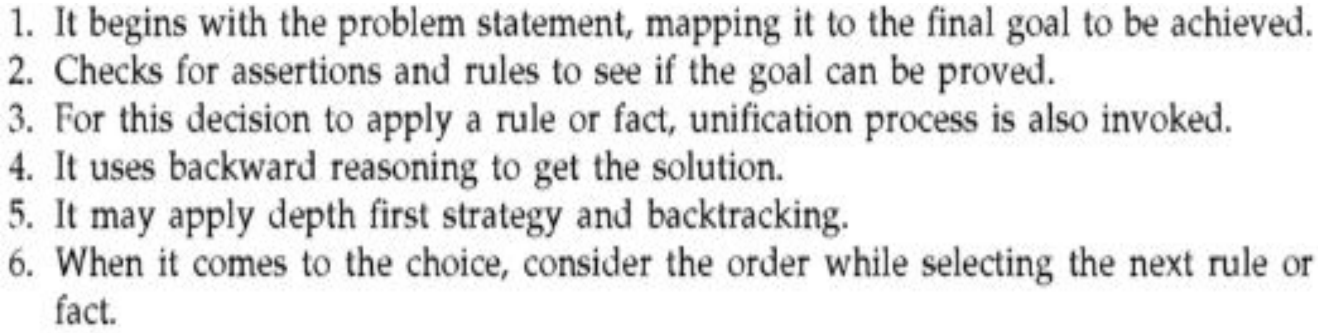


Bird(Parrot)
Bird(Sparrow)
Feathers(Pigeon)
$\forall x\ \text{Bird}(x) \Rightarrow \text{Feathers}(x)$

Let us say we want to solve a query to find out some y that has feathers. The query representation will be as follows:

$\exists y : \text{feathers}(y)$

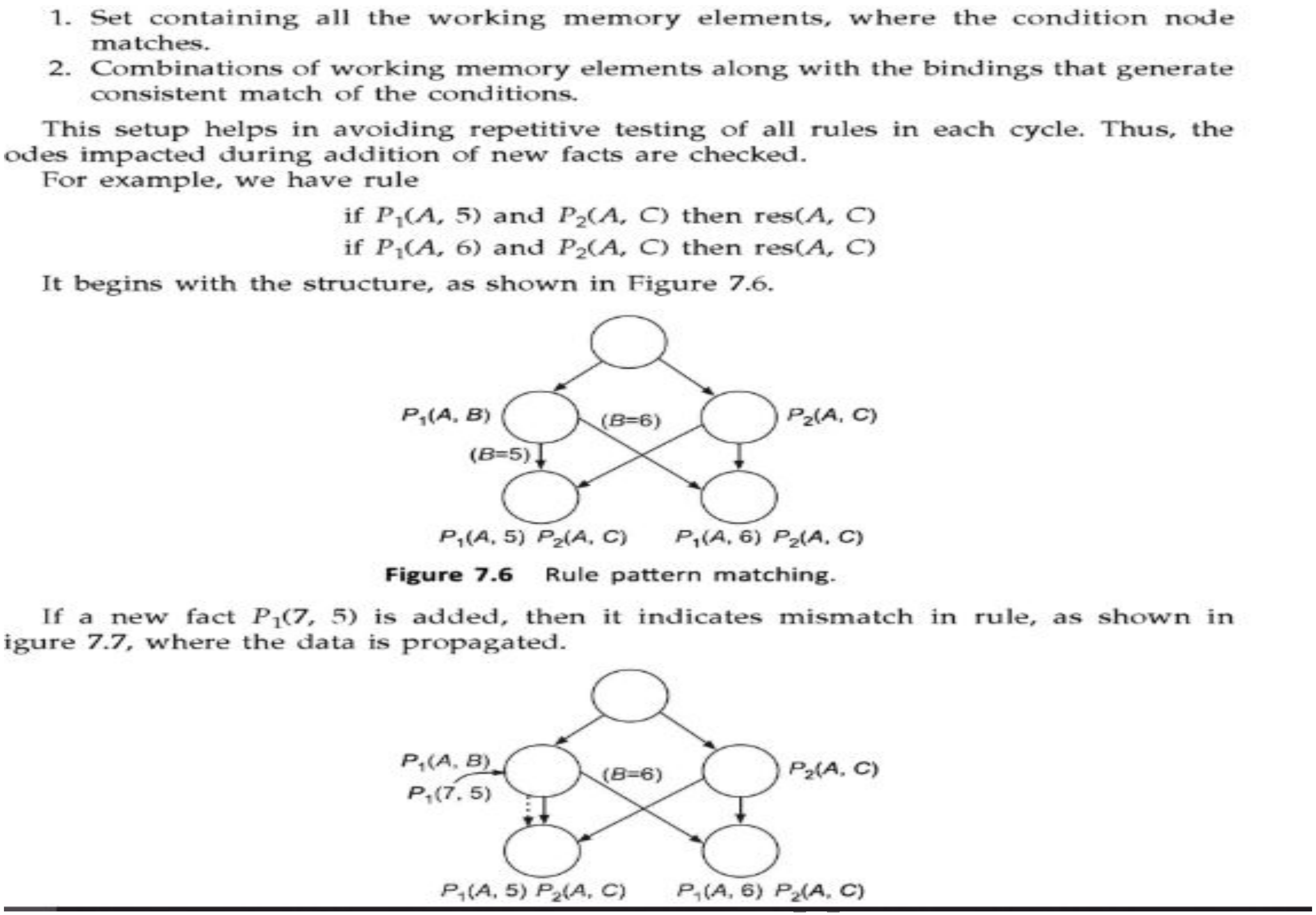
- Logic programming
  - It’s a programming paradigm which assertions are views as programs
  - Its comprises facts and rules.
  - Whether the axioms are sufficient to determine the truth of the goal.



1. It begins with the problem statement, mapping it to the final goal to be achieved.
2. Checks for assertions and rules to see if the goal can be proved.
3. For this decision to apply a rule or fact, unification process is also invoked.
4. It uses backward reasoning to get the solution.
5. It may apply depth first strategy and backtracking.
6. When it comes to the choice, consider the order while selecting the next rule or fact.

-Forward and backward reasoning

Forward reasoning :a tree is built considering the different moves or intermediate steps.
At each point of time ,the next level is generated with the selection rule.
The process is left to right evaluation of rule.
-Backward Reasoning
Process starts with the goal. The rule whose right matches with the root are considered
Prolog and MYCIN are the two systems under this category.
System directed by goal is backward chaining
Forward chaining is very simple to process
MATCHING
Perform matching that is carried out between current state and preconditioning of the rules
Indexing:
Rules can be indexed by pieces and their positions
Matching with variable
Generally unification algorithm used to solve this .
The algorithm maintains the network of the rule and it uses state description to determine which rule
Can be applicable.



1. Set containing all the working memory elements, where the condition node matches.
2. Combinations of working memory elements along with the bindings that generate consistent match of the conditions.

This setup helps in avoiding repetitive testing of all rules in each cycle. Thus, the odes impacted during addition of new facts are checked.

For example, we have rule

$$\text{if } P_1(A, 5) \text{ and } P_2(A, C) \text{ then res}(A, C)$$

$$\text{if } P_1(A, 6) \text{ and } P_2(A, C) \text{ then res}(A, C)$$

It begins with the structure, as shown in Figure 7.6.

Figure 7.6 Rule pattern matching.

If a new fact $P_1(7, 5)$ is added, then it indicates mismatch in rule, as shown in igure 7.7, where the data is propagated.

Appropriate and complex Matching:

Preconditions describe the properties that are not explicitly mentioned in the current state .

ELIZA-very old AI program that simulated behavior used this matching

Women: I need your help

ELIZA: How can I help You?

Conflict Resolution

The rule need to be matched and is equally necessary to decide the order in which this is to carried out .

This stage of matching is referred to as conflict resolution.

1.Rule that is matched

2.Objects that are matched .

3.Actions that the matched rule performed

7.9 Semantic network

•Basic idea behind knowledge is better understood as a set of concepts

•The semantic network comprises nodes that are connected to each other by arcs

•COMMION SEMANTIC RALATIONS

- 1.instances
- 2.ISA
- 3.Haspart

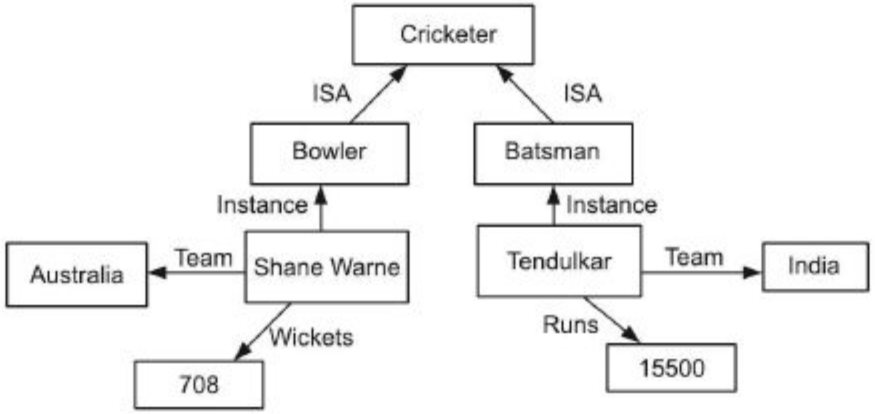
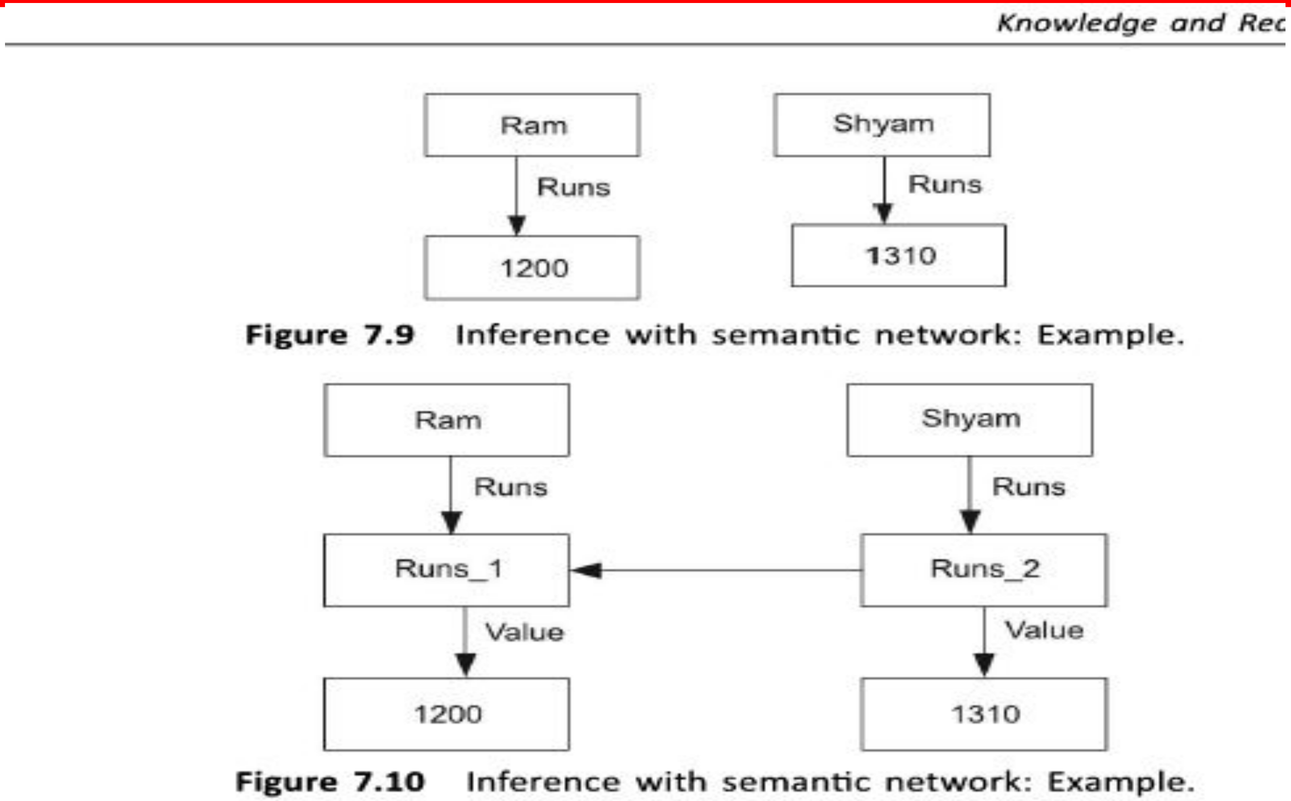


Figure 7.8 Semantic network.

Inference in Sematic Network

2 mechanisms are followed intersection search and inheritance

Intersection search::intersection between the nodes is found in the relations too

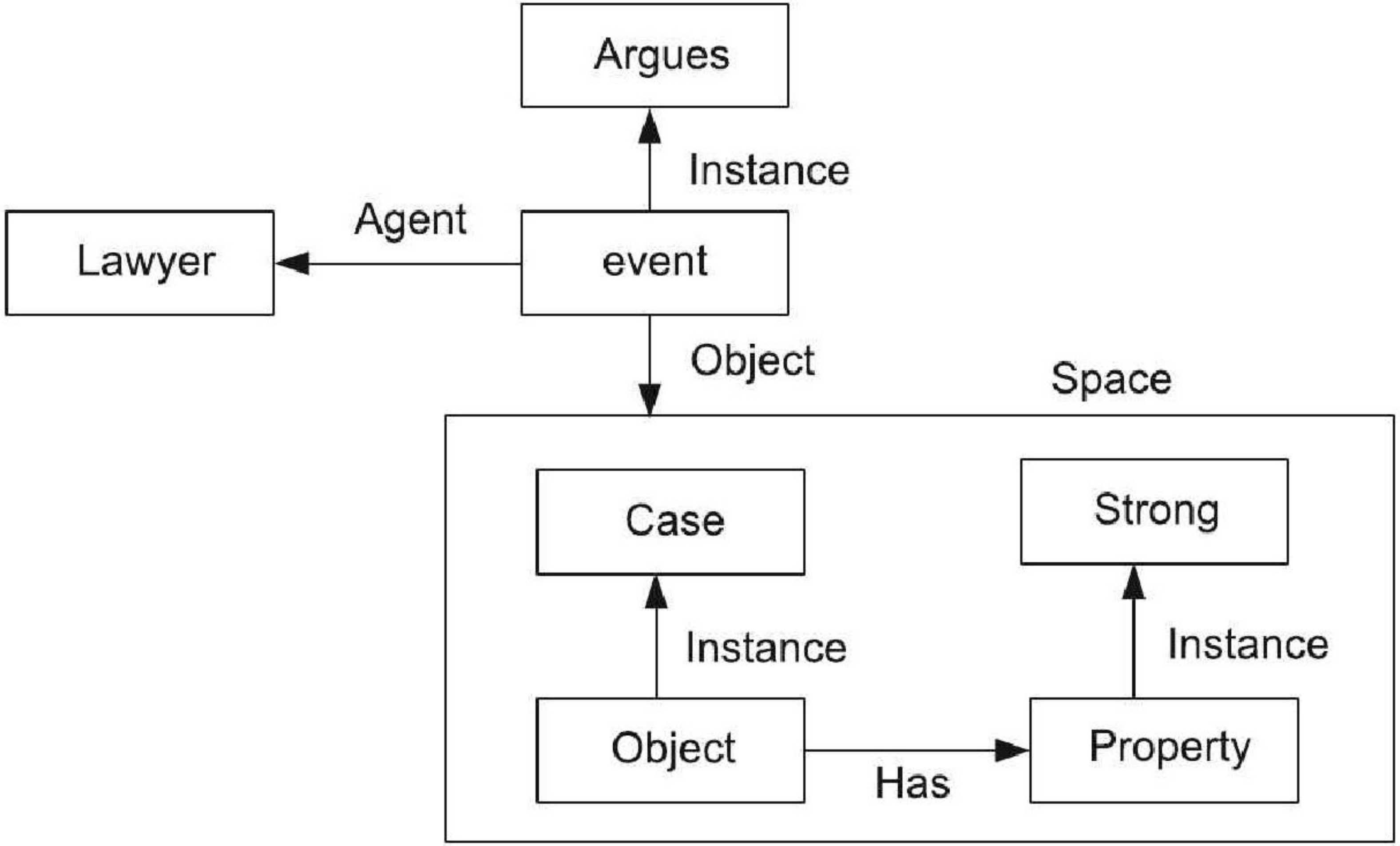


Knowledge and Rec

Figure 7.9 Inference with semantic network: Example.

Figure 7.10 Inference with semantic network: Example.

Partitioned semantic networks: extension of semantic networks.is forms more sophisticated logic structure . The network is spilt into spaces that consist of nodes and arcs.

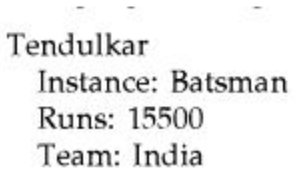

Argues
Instance
Agent
Lawyer
event
Object
Space
Case
Strong
Instance
Instance
Object
Has
Property

I stay dry

7.10 Frame systems

Frame based systems are knowledge representation systems that uses frames.
Frame is collection of attributes or slots.
Its concerned with labelled representations.it requires structured representation
Its structure to represent that have some associated value in the real world.
Minskys Frame:

These frame capture the concepts by clustering relevant information.
So large amount of procedurally expressive knowledge should be the part of frames
Representation of Frames:



Tendulkar
  Instance: Batsman
  Runs: 15500
  Team: India

Uncertain knowledge and reasoning

- In real life, it is not always possible to determine the state of the environment as it might not be clear. Due to partially observable or non-deterministic environments, agents may need to handle uncertainty and deal with it.
- Uncertain data: Data that is missing, unreliable, inconsistent or noisy
- Uncertain knowledge: When the available knowledge has multiple causes leading to multiple effects or incomplete knowledge of causality in the domain
- Uncertain knowledge representation: The representations which provides a restricted model of the real system, or has limited expressiveness
- Inference: In case of incomplete or default reasoning methods, conclusions drawn might not be completely accurate. Let’s understand this better with the help of an example.
- IF primary infection is bacteria cea
- AND site of infection is sterile
- AND entry point is gastrointestinal tract
- THEN organism is bacteriod (0.7).
- In such uncertain situations, the agent does not guarantee a solution but acts on its own assumptions and probabilities and gives some degree of belief that it will reach the required solution.

Uncertain knowledge and reasoning

- For example, In case of Medical diagnosis consider the rule Toothache = Cavity. This is not complete as not all patients having toothache have cavities. So we can write a more generalized rule Toothache = Cavity V Gum problems V Abscess… To make this rule complete, we will have to list all the possible causes of toothache. But this is not feasible due to the following rules:
- *Laziness- It will require a lot of effort to list the complete set of antecedents and consequents to make the rules complete.*
- *Theoretical ignorance- Medical science does not have complete theory for the domain*
- *Practical ignorance- It might not be practical that all tests have been or can be conducted for the patients.*
- Such uncertain situations can be dealt with using

Probability theory

Truth Maintenance

systems Fuzzy logic.

Uncertain knowledge and reasoning

Probability

- Probability is the degree of likeliness that an event will occur. It provides a certain degree of belief in case of uncertain situations. It is defined over a set of events U and assigns value P(e) i.e. probability of occurrence of event e in the range [0,1]. Here each sentence is labeled with a real number in the range of 0 to 1, 0 means the sentence is false and 1 means it is true.
- Conditional Probability or Posterior Probability is the probability of event A given that B has already occurred.
- $P(A|B) = (P(B|A) * P(A)) / P(B)$
- For example, P(It will rain tomorrow| It is raining today) represents conditional probability of it raining tomorrow as it is raining today.
- $P(A|B) + P(NOT(A)|B) = 1$
- Joint probability is the probability of 2 independent events happening simultaneously like rolling two dice or tossing two coins together. For example, Probability of getting 2 on one dice and 6 on the other is equal to 1/36. Joint probability has a wide use in various fields such as physics, astronomy, and comes into play when there are two independent events. The full joint probability distribution specifies the probability of each complete assignment of values to random variables.

Uncertain knowledge and reasoning

Bayes Theorem

- It is based on the principle that every pair of features being classified is independent of each other. It calculates probability P(A|B) where A is class of possible outcomes and B is given instance which has to be classified.
- P(A|B) = P(B|A) * P(A) / P(B)
- P(A|B) = Probability that A is happening, given that B has occurred (posterior probability)
- P(A) = prior probability of class
- P(B) = prior probability of predictor
- P(B|A) = likelihood

Uncertain knowledge and reasoning

Consider the following data. Depending on the weather (sunny, rainy or overcast), the children will play(Y) or not play(N).

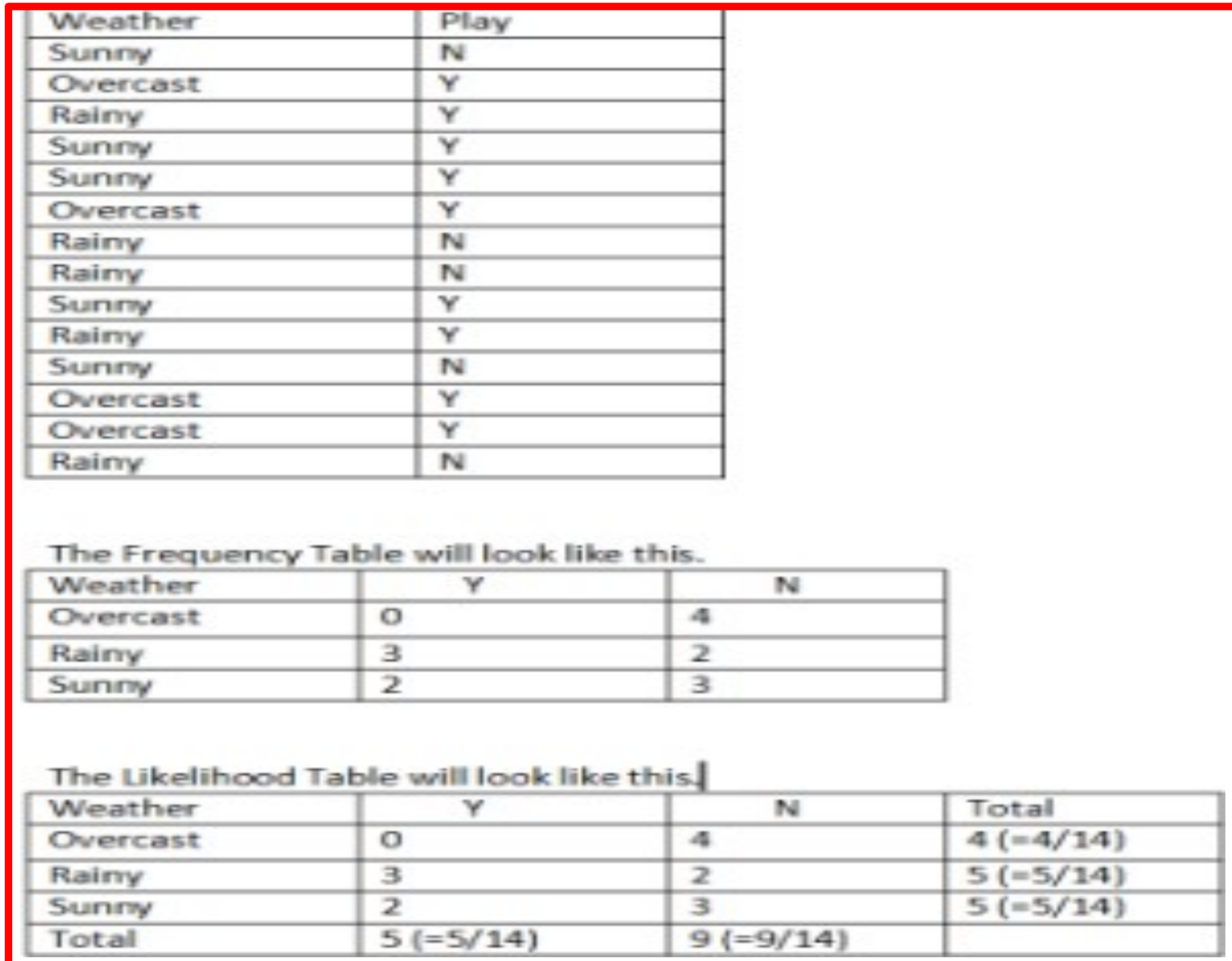
Here, the total number of observations = 14

Probability that children will play given that weather is sunny :

P(Yes| Sunny) = P(Sunny | Yes) * P(Yes) / P(Sunny)

= 0.33 * 0.64 / 0.36

= 0.59



Weather	Play
Sunny	N
Overcast	Y
Rainy	Y
Sunny	Y
Sunny	Y
Overcast	Y
Rainy	N
Rainy	N
Sunny	Y
Rainy	Y
Sunny	N
Overcast	Y
Overcast	Y
Rainy	N

The Frequency Table will look like this.

Weather	Y	N
Overcast	0	4
Rainy	3	2
Sunny	2	3

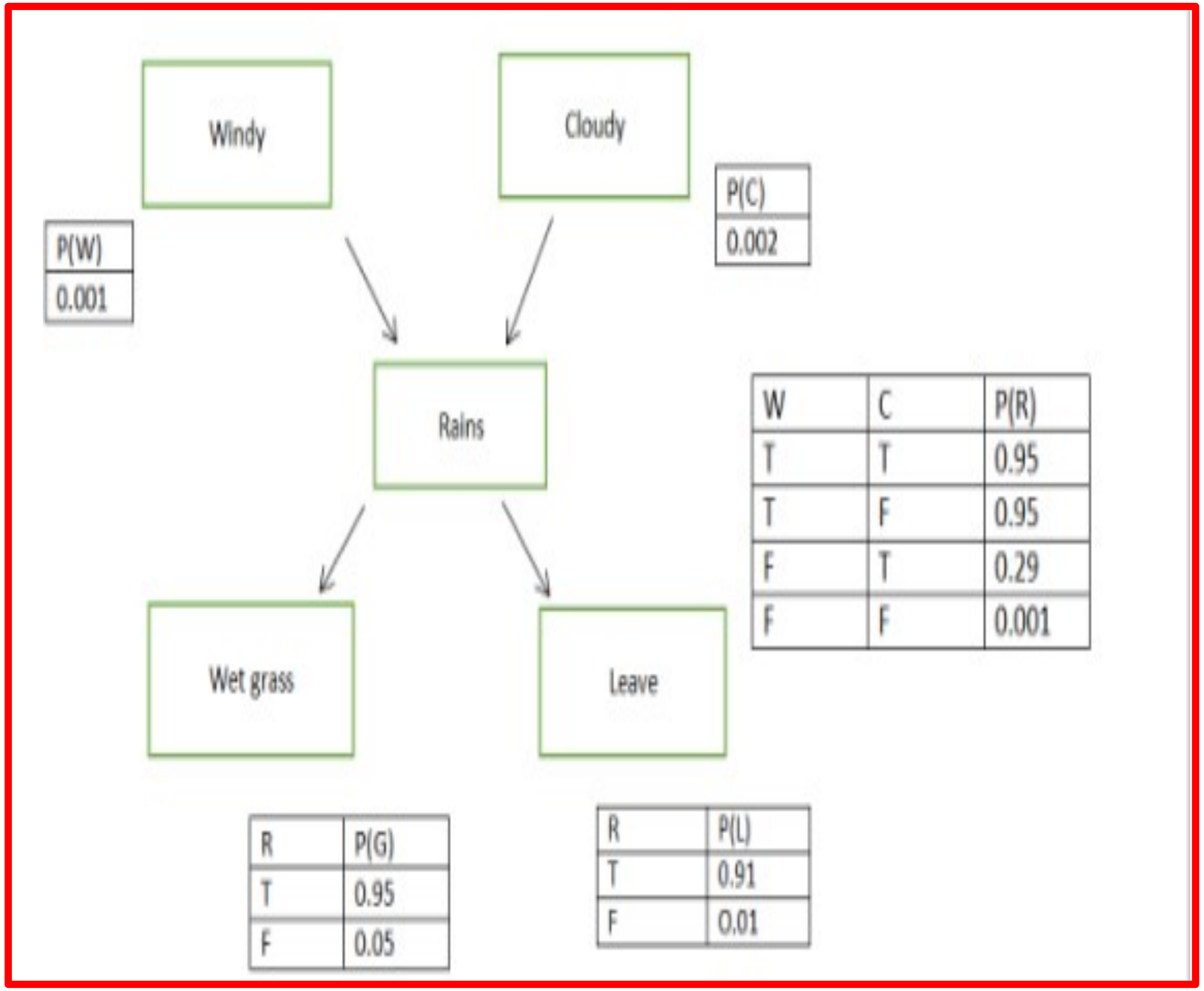
The Likelihood Table will look like this.

Weather	Y	N	Total
Overcast	0	4	4 (=4/14)
Rainy	3	2	5 (=5/14)
Sunny	2	3	5 (=5/14)
Total	5 (=5/14)	9 (=9/14)	

Uncertain knowledge and reasoning

It is a probabilistic graphical model for representing uncertain domain and to reason under uncertainty. It consists of nodes representing variables, arcs representing direct connections between them, called causal correlations. It represents conditional dependencies between random variables through a Directed Acyclic Graph (DAG). A belief network consist of:

1. A DAG with nodes labeled with variable names,
2. Domain for each random variable,
3. Set of conditional probability tables for each variable given its parents, including prior probability for nodes with no parents.



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(G)
T	0.95
F	0.05

R	P(L)
T	0.91
F	0.01

Uncertain knowledge and reasoning

- Let’s have a look at the steps followed.

1. Identify nodes which are the random variables and the possible values they can have from the probability domain. The nodes can be boolean (True/ False), have ordered values or integral values.
2. Structure- It is used to represent causal relationships between the variables. Two nodes are connected if one affects or causes the other and the arc points towards the effect. For instance, if it is windy or cloudy, it rains. There is a direct link from Windy/Cloudy to Rains. Similarly, from Rains to Wet grass and Leave, i.e., if it rains, grass will be wet and leave is taken from work.

Uncertain knowledge and reasoning

3. Probability- Quantifying relationship between nodes.
   Conditional Probability:

- P(A^B) = P(A|B) * P(B)
- P(A|B) = P(B|A) * P(A)
- P(B|A) = P(A|B) * P(B) / P(A)
- Joint probability:

4. Markov property- Bayesian Belied Networks require assumption of Markov property, i.e., all direct dependencies are shown by using arcs. Here there is no direct connection between it being Cloudy and Taking a leave. But there is one via Rains. Belief Networks which have Markov property are also called independence maps.

Uncertain knowledge and reasoning

Inference in Belief Networks

- Bayesian Networks provide various types of representations of probability distribution over their variables. They can be conditioned over any subset of their variables, using any direction of reasoning.
- For example, one can perform diagnostic reasoning, i.e. when it Rains, one can update his belief about the grass being wet or if leave is taken from work. In this case reasoning occurs in the opposite direction to the network arcs. Or one can perform predictive reasoning, i.e., reasoning from new information about causes to new beliefs about effects, following direction of the arcs. For example, if the grass is already wet, then the user knows that it has rained and it might have been cloudy or windy. Another form of reasoning involves reasoning about mutual causes of a common effect. This is called inter causal reasoning.
- There are two possible causes of an effect, represented in the form of a 'V'. For example, the common effect 'Rains' can be caused by two reasons 'Windy' and 'Cloudy.' Initially, the two causes are independent of each other but if it rains, it will increase the probability of both the causes. Assume that we know it was windy. This information explains the reasons for the rainfall and lowers probability that it was cloudy.

Knowledge and Reasoning Table of

- Knowledge and reasoning-Approaches and issues of knowledge reasoning-Knowledge base agents
- Logic Basics-Logic-Propositional logic-syntax ,semantics and inferences-Propositional logic- Reasoning patterns
- Unification and Resolution-Knowledge representation using rules-Knowledge representation using semantic nets
- Knowledge representation using frames-Inferences-
- Uncertain Knowledge and reasoning-Methods-Bayesian probability and belief network
- Probabilistic reasoning-Probabilistic reasoning over time
- Other uncertain techniques-Data mining-Fuzzy logic-Dempster -shafer theory

Bayesian probability and belief network

- Bayesian belief network is key computer technology for dealing with probabilistic events and to solve a problem which has uncertainty. We can define a Bayesian network as:
- "A Bayesian network is a probabilistic graphical model which represents a set of variables and their conditional dependencies using a directed acyclic graph."
- It is also called a **Bayes network, belief network, decision network**, or **Bayesian model**.

Bayesian probability and belief network

- Bayesian networks are probabilistic, because these networks are built from a **probability distribution**, and also use probability theory for prediction and anomaly detection.
- Real world applications are probabilistic in nature, and to represent the relationship between multiple events, we need a Bayesian network. It can also be used in various tasks including **prediction, anomaly detection, diagnostics, automated insight, reasoning, time series prediction**, and **decision making under uncertainty**.
- Bayesian Network can be used for building models from data and experts opinions, and it consists of two parts:

Directed Acyclic Graph

Table of conditional probabilities.

- The generalized form of Bayesian network that represents and solve decision problems under uncertain knowledge is known as an **Influence diagram**.

Bayesian probability and belief network

Directed Acyclic Graph

A Bayesian network graph is made up of nodes and Arcs (directed links), where:

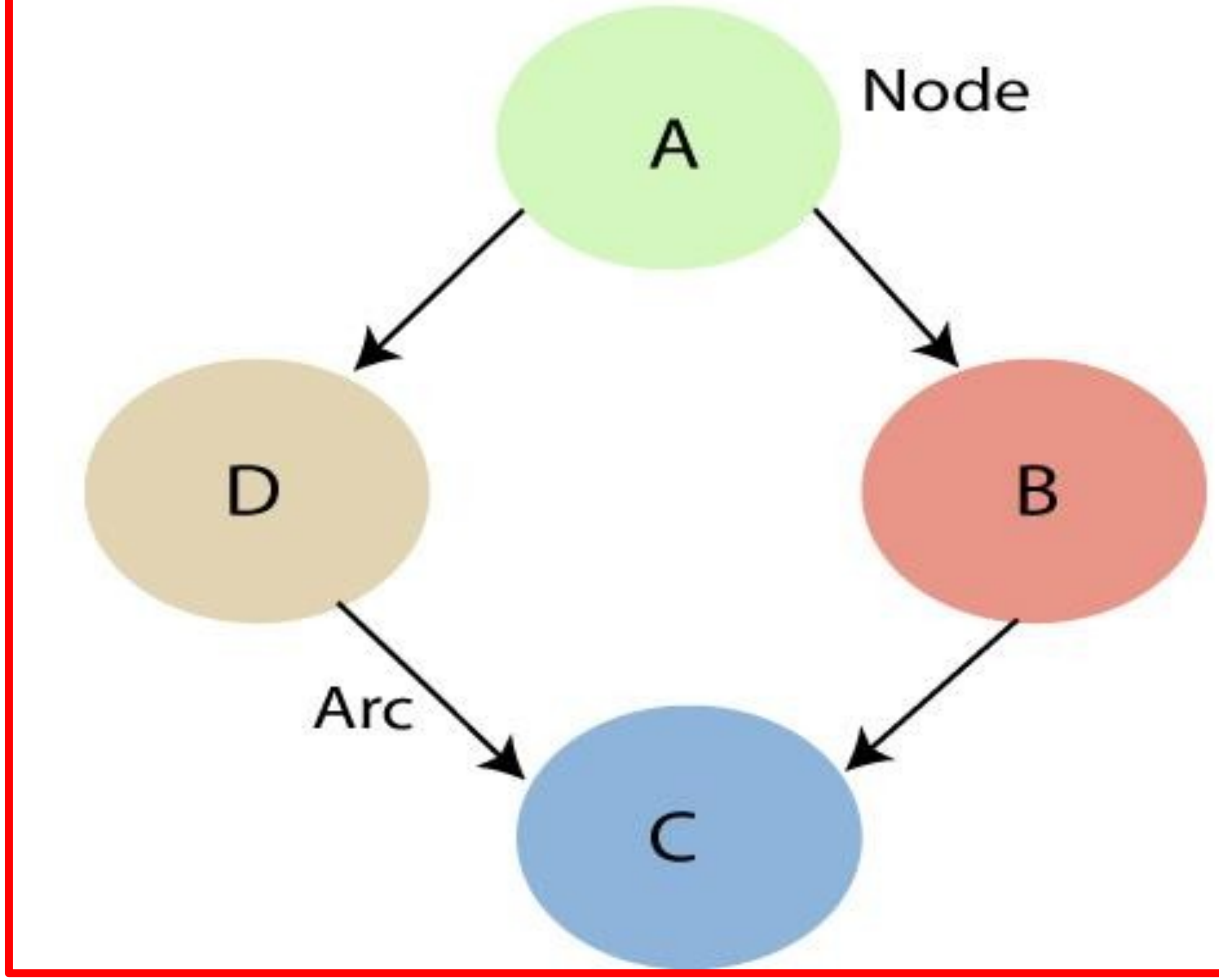
Each **node** corresponds to the random variables, and a variable can be **continuous** or **discrete**.

Arc or directed arrows represent the causal relationship or conditional probabilities between random variables. These directed links or arrows connect the pair of nodes in the graph.
These links represent that one node directly influence the other node, and if there is no directed link that means that nodes are independent with each other

In the above diagram, A, B, C, and D are random variables represented by the nodes of the network graph.

If we are considering node B, which is connected with node A by a directed arrow, then node A is called the parent of Node B.

Node C is independent of node A.



Bayesian probability and belief network

CONDITIONAL PROBABILITY

- The Bayesian network has mainly two components:

Causal Component

Actual numbers

- Each node in the Bayesian network has condition probability distribution $P(X_i \mid Parent(X_i))$, which determines the effect of the parent on that node.
- Bayesian network is based on Joint probability distribution and conditional probability. So let's first understand the joint probability distribution:

Bayesian probability and belief network

Joint probability distribution:

- If we have variables x1, x2, x3,....., xn, then the probabilities of a different combination of x1, x2, x3.. xn, are known as Joint probability distribution.
- **$P[x_1, x_2, x_3,....., x_n]$**, it can be written as the following way in terms of the joint probability distribution.
- **$= P[x_1| x_2, x_3,....., x_n]P[x_2, x_3,....., x_n]$**
- **$= P[x_1| x_2, x_3,....., x_n]P[x_2|x_3,....., x_n]....P[x_{n-1}|x_n]P[x_n]$.**
- In general for each variable Xi, we can write the equation as:
- $P(X_i|X_{i-1},........., X_1) = P(X_i |Parents(X_i))$

Bayesian probability and belief network

Explanation of Bayesian network:

- Let's understand the Bayesian network through an example by creating a directed acyclic graph:

Example: Harry installed a new burglar alarm at his home to detect burglary. The alarm reliably responds at detecting a burglary but also responds for minor earthquakes. Harry has two neighbors David and Sophia, who have taken a responsibility to inform Harry at work when they hear the alarm. David always calls Harry when he hears the alarm, but sometimes he got confused with the phone ringing and calls at that time too. On the other hand, Sophia likes to listen to high music, so sometimes she misses to hear the alarm. Here we would like to compute the probability of Burglary Alarm.

Bayesian probability and belief network

Problem:

- **Calculate the probability that alarm has sounded, but there is neither a burglary, nor an earthquake occurred, and David and Sophia both called the Harry.**

Solution:

- The Bayesian network for the above problem is given below. The network structure is showing that burglary and earthquake is the parent node of the alarm and directly affecting the probability of alarm's going off, but David and Sophia's calls depend on alarm probability.
- The network is representing that our assumptions do not directly perceive the burglary and also do not notice the minor earthquake, and they also not confer before calling.
- The conditional distributions for each node are given as conditional probabilities table or CPT.
- Each row in the CPT must be sum to 1 because all the entries in the table represent an exhaustive set of cases for the variable.
- In CPT, a boolean variable with k boolean parents contains 2^K probabilities. Hence, if there are two parents, then CPT will contain 4 probability values

Bayesian probability and belief network

List of all events occurring in this network:

- **Burglary (B)**
- **Earthquake(E)**
- **Alarm(A)**
- **David Calls(D)**
- **Sophia calls(S)**

We can write the events of problem statement in the form of probability: **P[D, S, A, B, E]**, can rewrite the above probability statement using joint probability distribution:

- **P[D, S, A, B, E]= P[D | S, A, B, E]. P[S, A, B, E]**
- **=P[D | S, A, B, E]. P[S | A, B, E]. P[A, B, E]**
- **= P [D| A]. P [S| A, B, E]. P[A, B, E]**
- **= P[D | A]. P[S | A]. P[A| B, E]. P[B, E]**
- **= P[D | A]. P[S | A]. P[A| B, E]. P[B |E]. P[E]**

Bayesian probability and belief network

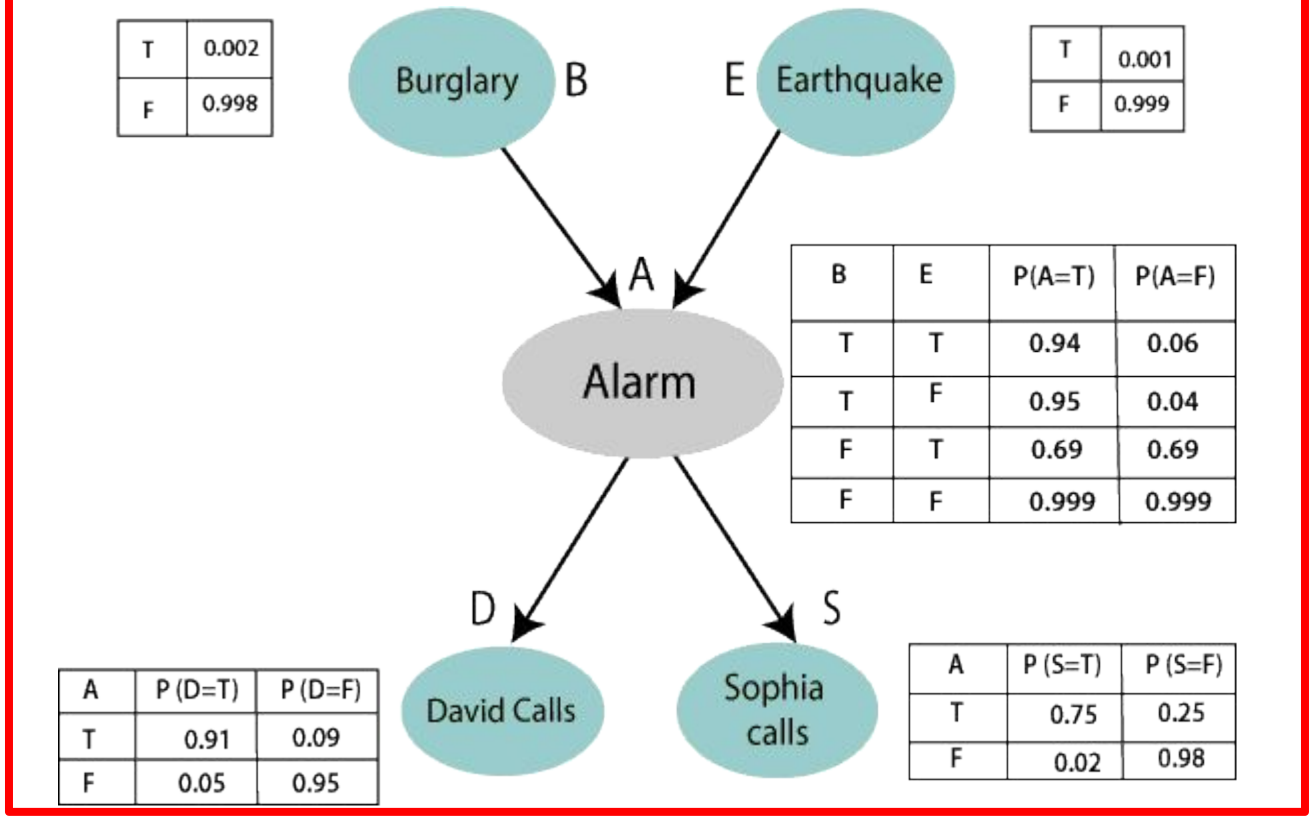
Let's take the observed probability for the Burglary and earthquake component:

P(B=True) = 0.002, which is the probability of burglary.

P(B=False)= 0.998, which is the probability of no burglary.

P(E=True)= 0.001, which is the probability of a minor earthquake

P(E=False)= 0.999, Which is the probability that an earthquake not occurred.



T	0.002
F	0.998

T	0.001
F	0.999

B	E	P(A=T)	P(A=F)
T	T	0.94	0.06
T	F	0.95	0.04
F	T	0.69	0.69
F	F	0.999	0.999

A	P (D=T)	P (D=F)
T	0.91	0.09
F	0.05	0.95

A	P (S=T)	P (S=F)
T	0.75	0.25
F	0.02	0.98

Bayesian probability and belief network

We can provide the conditional probabilities as per the below tables:

Conditional probability table for Alarm A:

The Conditional probability of Alarm A depends on Burglar and earthquake:

B	E	P(A= True)	P(A= False)
True	True	0.94	0.06
True	False	0.95	0.04
False	True	0.31	0.69
False	False	0.001	0.999

Bayesian probability and belief network

Conditional probability table for David Calls:

The Conditional probability of David that he will call depends on the probability of Alarm.

A	P(D= True)	P(D= False)
True	0.91	0.09
False	0.05	0.95

Bayesian probability and belief network

Conditional probability table for Sophia Calls:

The Conditional probability of Sophia that she calls is depending on its Parent Node "Alarm."

A	P(S= True)	P(S= False)
True	0.75	0.25
False	0.02	0.98

Bayesian probability and belief network

- From the formula of joint distribution, we can write the problem statement in the form of probability distribution:
- **P(S, D, A, ¬B, ¬E) = P (S|A) *P (D|A)*P (A|¬B ^ ¬E) *P (¬B) *P (¬E).**

= 0.75* 0.91* 0.001* 0.998*0.999

= 0.00068045.

Hence, a Bayesian network can answer any query about the domain by using Joint distribution.

- **The semantics of Bayesian Network:**
- There are two ways to understand the semantics of the Bayesian network, which is given below:

1. To understand the network as the representation of the Joint probability distribution.

- It is helpful to understand how to construct the network.

2. To understand the network as an encoding of a collection of conditional independence statements.

- It is helpful in designing inference procedure.

Bayes' theorem in Artificial intelligence

Bayes' theorem:

- Bayes' theorem is also known as **Bayes' rule, Bayes' law**, or **Bayesian reasoning**, which determines the probability of an event with uncertain knowledge.
- In probability theory, it relates the conditional probability and marginal probabilities of two random events.
- Bayes' theorem was named after the British mathematician **Thomas Bayes**. The **Bayesian inference** is an application of Bayes' theorem, which is fundamental to Bayesian statistics.
- It is a way to calculate the value of P(B|A) with the knowledge of P(A|B).
- Bayes' theorem allows updating the probability prediction of an event by observing new information of the real world.

Bayes' theorem in Artificial intelligence

Example: If cancer corresponds to one's age then by using Bayes' theorem, we can determine the probability of cancer more accurately with the help of age.

- Bayes' theorem can be derived using product rule and conditional probability of event A with known event B:
- As from product rule we can write:
- P(A ∧ B)= P(A|B) P(B) or
- Similarly, the probability of event B with known event A:
- P(A ∧ B)= P(B|A) P(A)
- Equating right hand side of both the equations, we will get:

The above equation (a) is called as **Bayes' rule** or **Bayes' theorem**. This equation is basic of most modern AI systems for **probabilistic inference**.

- It shows the simple relationship between joint and conditional probabilities. Here,
- P(A|B) is known as **posterior**, which we need to calculate, and it will be read as Probability of hypothesis A when we have occurred an evidence B.
- P(B|A) is called the likelihood, in which we consider that hypothesis is true, then we calculate the probability of evidence.
- P(A) is called the **prior probability**, probability of hypothesis before considering the evidence
- P(B) is called **marginal probability**, pure probability of an evidence.
- In the equation (a), in general, we can write P (B) = P(A)*P(B|Ai), hence the Bayes' rule can be written as:
- We can write if A_1, A_2, A_3,........, A_n is a set of mutually exclusive and exhaustive events.

Applying Bayes' theorem in Artificial intelligence rule:

- Bayes' rule allows us to compute the single term P(B|A) in terms of P(A|B), P(*B*), and P(A). This is very useful in cases where we have a good probability of these three terms and want to determine the fourth one. Suppose we want to perceive the effect of some unknown cause, and want to compute that cause, then the Bayes' rule becomes:

Example-1:

Question: what is the probability that a patient has diseases meningitis with a stiff neck?

- **Given Data:**

A doctor is aware that disease meningitis causes a patient to have a stiff neck, and it occurs 80% of the time. He is also aware of some more facts, which are given as follows:

The Known probability that a patient has meningitis disease is 1/30,000.

The Known probability that a patient has a stiff neck is 2%.

Let a be the proposition that patient has stiff neck and b be the proposition that patient has meningitis. , so we can calculate the following as:

P(a|b) = 0.8

P(b) = 1/30000

P(a)= .02

- Hence, we can assume that 1 patient out of 750 patients has meningitis disease with a stiff neck.

Applying Bayes' theorem in Artificial intelligence

Example-2:

Question: From a standard deck of playing cards, a single card is drawn. The probability that the card is king is 4/52, then calculate posterior probability P(King|Face), which means the drawn face card is a king card.

Solution:

$$\mathbf{P(king|face)} = \frac{P(Face|king) * P(King)}{P(Face)} \quad ....(i)$$

P(king): probability that the card is King= 4/52= 1/13

P(face): probability that a card is a face card= 3/13

P(Face|King): probability of face card when we assume it is a king =

1 Putting all values in equation (i) we will get:

$$P(king|face) = \frac{1 * (\frac{1}{13})}{(\frac{3}{13})} = \mathbf{1/3}, \text{ it is a probability that a face card is a king card.}$$

-

Application of Bayes' theorem in Artificial intelligence

.

Following are some applications of Bayes' theorem:

- It is used to calculate the next step of the robot when the already executed step is given.
- Bayes' theorem is helpful in weather forecasting.
- It can solve the Monty Hall problem.

Knowledge and Reasoning Table of

Probabilistic reasoning

Uncertainty:

- Till now, we have learned knowledge representation using first-order logic and propositional logic with certainty, which means we were sure about the predicates. With this knowledge representation, we might write A→B, which means if A is true then B is true, but consider a situation where we are not sure about whether A is true or not then we cannot express this statement, this situation is called uncertainty.
- So to represent uncertain knowledge, where we are not sure about the predicates, we need uncertain reasoning or probabilistic reasoning.

Causes of uncertainty:

Following are some leading causes of uncertainty to occur in the real world.

- Information occurred from unreliable sources.
- Experimental Errors
- Equipment fault
- Temperature variation
- Climate change.

Probabilistic reasoning

Probabilistic reasoning:

- Probabilistic reasoning is a way of knowledge representation where we apply the concept of probability to indicate the uncertainty in knowledge. In probabilistic reasoning, we combine probability theory with logic to handle the uncertainty.
- We use probability in probabilistic reasoning because it provides a way to handle the uncertainty that is the result of someone's laziness and ignorance.
- In the real world, there are lots of scenarios, where the certainty of something is not confirmed, such as "It will rain today," "behavior of someone for some situations," "A match between two teams or two players." These are probable sentences for which we can assume that it will happen but not sure about it, so here we use probabilistic reasoning.

Need of probabilistic reasoning in AI:

- When there are unpredictable outcomes.
- When specifications or possibilities of predicates becomes too large to handle.
- When an unknown error occurs during an experiment.

In probabilistic reasoning, there are two ways to solve problems with uncertain knowledge:

- **Bayes' rule**
- **Bayesian Statistics**

Probabilistic reasoning

As probabilistic reasoning uses probability and related terms, so before understanding probabilistic reasoning, let's understand some common terms:

Probability: Probability can be defined as a chance that an uncertain event will occur. It is the numerical measure of the likelihood that an event will occur. The value of probability always remains between 0 and 1 that represent ideal uncertainties.

$0 \leq P(A) \leq 1$, where P(A) is the probability of an event A.

P(A) = 0, indicates total uncertainty in an event A.

P(A) =1, indicates total certainty in an event A.

We can find the probability of an uncertain event by using the below formula.

P(¬A) = probability of a not happening event. $P(\neg A) + P(A) = 1$.

- **Event:** Each possible outcome of a variable is called an event.
- **Sample space:** The collection of all possible events is called sample space.
- **Random variables:** Random variables are used to represent the events and objects in the real world.
- **Prior probability:** The prior probability of an event is probability computed before observing new information.
- **Posterior Probability:** The probability that is calculated after all evidence or information has taken into account. It is a combination of prior probability and new information.

Probabilistic reasoning

Conditional probability:

- Conditional probability is a probability of occurring an event when another event has already happened.

Let's suppose, we want to calculate the event A when event B has already occurred, "the probability of A under the conditions of B", it can be written as:

$P(A/B)=P(A \wedge B)/P(B)$

- **Where $P(A \wedge B)$= Joint probability of a and B**
- **P(B)= Marginal probability of B.**
- If the probability of A is given and we need to find the probability of B, then it will be given as: $P(B/A)=P(A \wedge B)/P(A)$
- It can be explained by using the below Venn diagram, where B is occurred event, so sample space will be reduced to set B, and now we can only calculate event A when event B is already occurred by dividing the probability of **$P(A \wedge B)$ by P(B)**.

Probabilistic reasoning

Example:

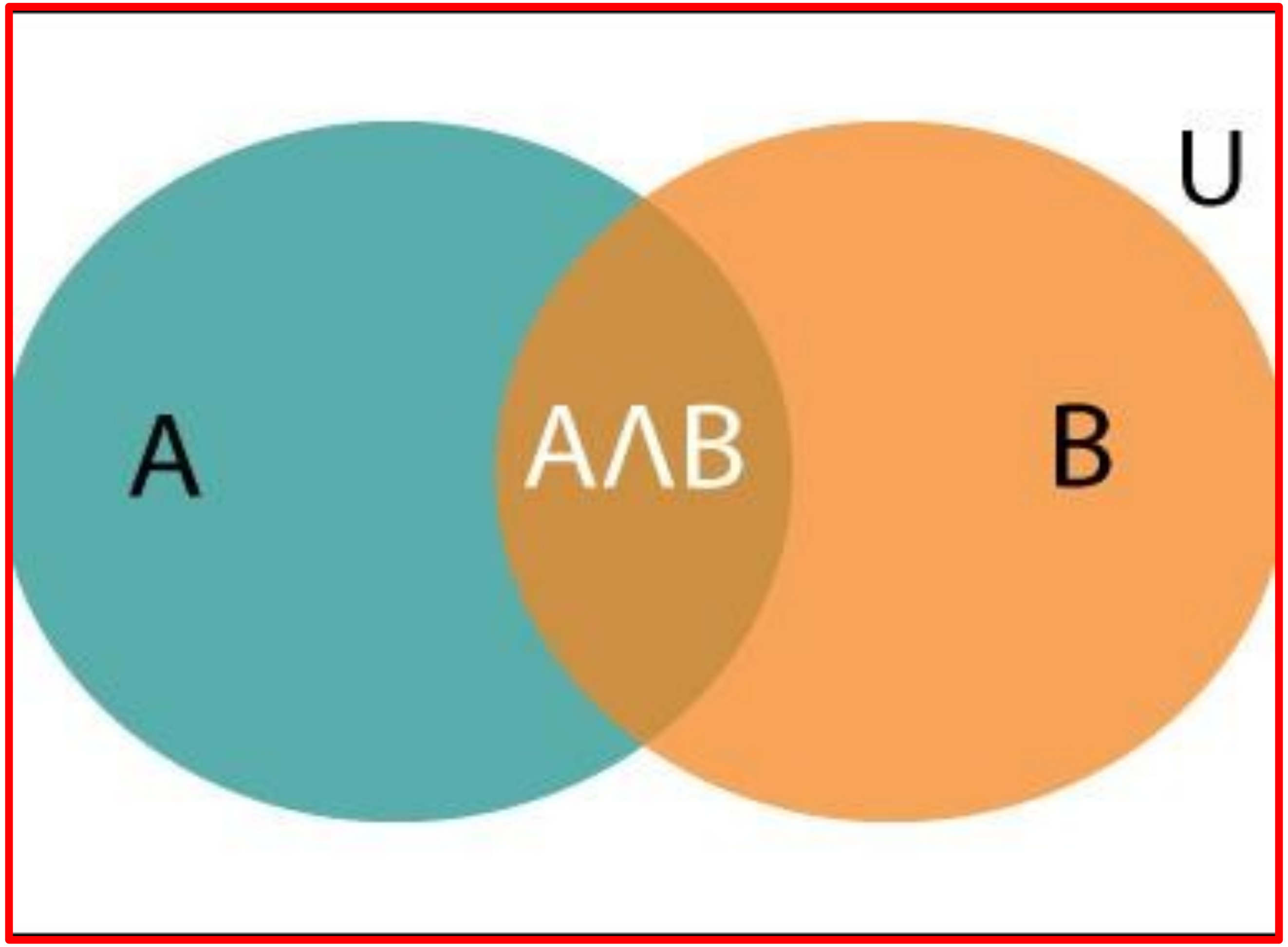
In a class, there are 70% of the students who like English and 40% of the students who likes English and mathematics, and then what is the percent of students those who like English also like mathematics?

Solution:

Let, A is an event that a student likes Mathematics

B is an event that a student likes English.

Hence, 57% are the students who like English also like Mathematics.



Knowledge and Reasoning Table of

- Knowledge and reasoning-Approaches and issues of knowledge reasoning-Knowledge base agents
- Logic Basics-Logic-Propositional logic-syntax ,semantics and inferences-Propositional logic- Reasoning patterns
- Unification and Resolution-Knowledge representation using rules-Knowledge representation using semantic nets
- Knowledge representation using frames-Inferences-
- Uncertain Knowledge and reasoning-Methods-Bayesian probability and belief network
- Probabilistic reasoning-Probabilistic reasoning over time
- Other uncertain techniques-Data mining-Fuzzy logic-Dempster -shafer theory

Probabilistic reasoning over time

Definition

- Probabilistic reasoning is the representation of knowledge where the concept of probability is applied to indicate the uncertainty in knowledge.

Reasons to use Probabilistic Reasoning in AI

- Some reasons to use this way of representing knowledge is given below:
- When we are unsure of the predicates.
- When the possibilities of predicates become too large to list down.
- When during an experiment, it is proven that an error occurs.
- **Probability** of a given event = Chances of that event occurring / Total number of Events.

Notations and Properties

- Consider the statement S: March will be cold.
- Probability is often denoted as P(predicate).
- Considering the chances of March being cold is only 30%, therefore, **P(S) = 0.3**
- Probability always takes a value between 0 and 1. If the probability is 0, then the event will never occur and if it is 1, then it will occur for sure.
- Then, $P(\neg S) = 0.7$
- This means, the probability of March not being cold is 70%.
- Property 1: $P(S) + P(\neg S) = 1$

Probabilistic reasoning over time

Consider the statement T: April will be cold.

- Then, P(SAT) means Probability of S AND T, i.e., Probability of March and April being cold.
- P(S∨T) means Probability of S OR T, i.e., Probability of March or April being cold.
- Property 2: P(S∨T) = P(S) + P(T) - P(SAT)
- Proofs for the properties are not given here and you can work them out by yourselves using Venn Diagrams.

Conditional Property

- Conditional Property is defined as the probability of a given event given another event. It is denoted by **P(B|A)** and is read as: ''Probability of B given probability of A.''
- Property 3: P(B|A) = P(BAA) / P(A).

Bayes' Theorem

- Given P(A), P(B) and P(A|B), then
- P(B|A) = P(A|B) x P(B) / P(A)

Probabilistic reasoning over time

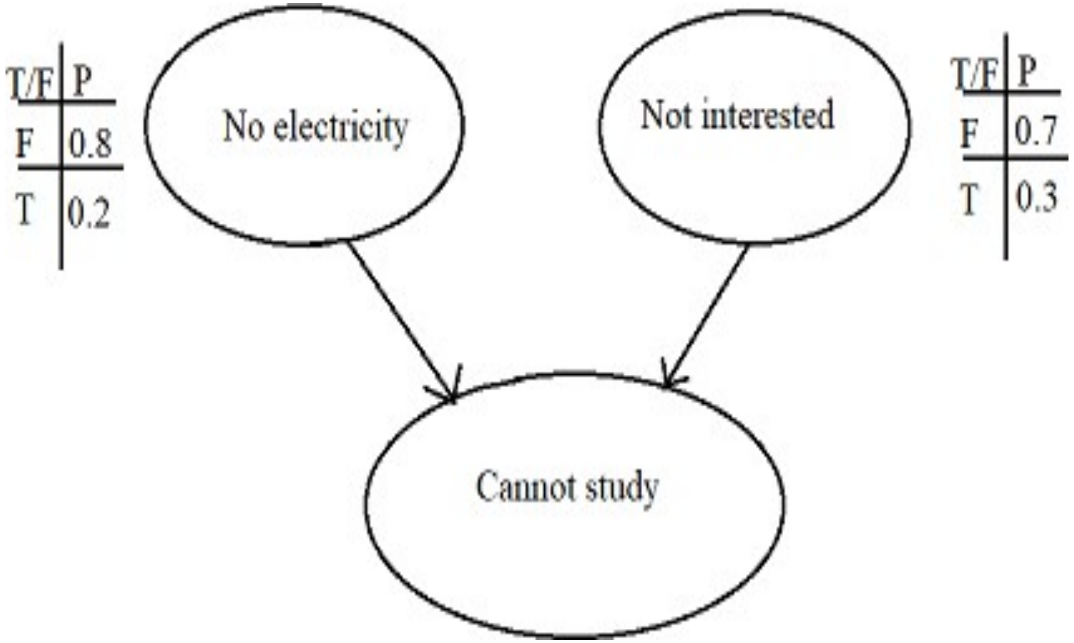
Bayesian Network

When designing a Bayesian Network, we keep the **local probability table** at each node.

Bayesian Network - Example

Consider a Bayesian Network as given below:

This Bayesian Network tells us the reason a particular person cannot study. It may be either because of no electricity or because of his lack of interest. The corresponding probabilities are written in front of the causes.



T/F	P
F	0.8
T	0.2

T/F	P
F	0.7
T	0.3

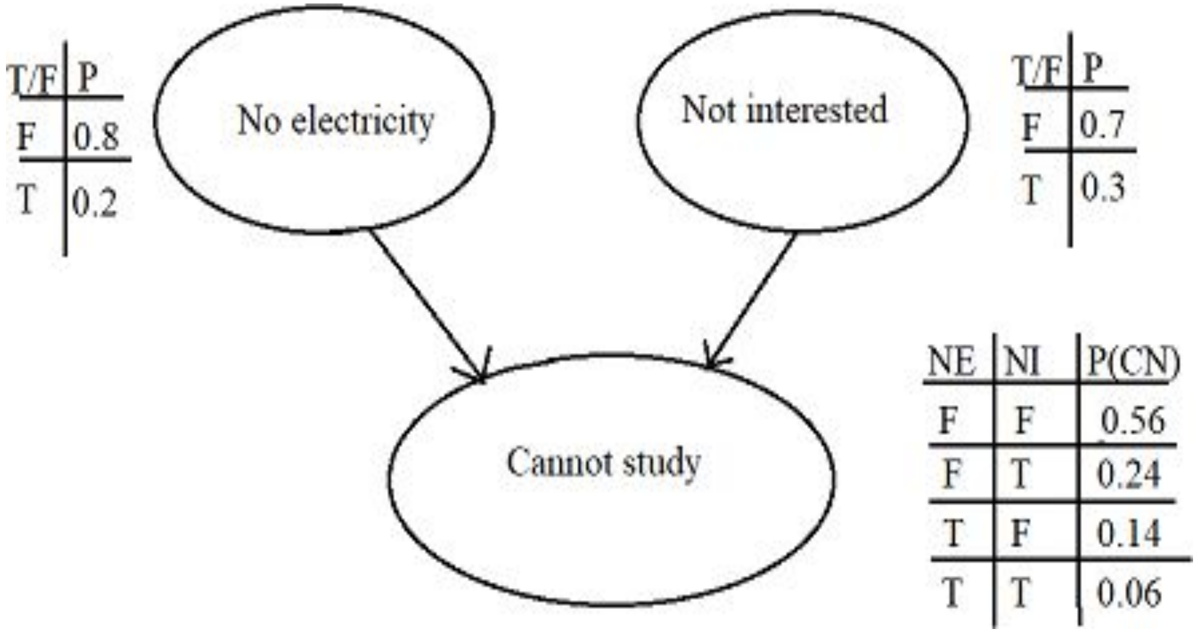
Probabilistic reasoning overtime

Now, as you can see no cause is dependent on each other and they directly contribute to the person's inability to study. To plot the third table, we consider four cases. Since, the causes are independent, their corresponding probabilities can be multiplied directly.

No Electricity	Not interested	P(Cannot Study)
F	F	P(No electricity = F) x P(Not Interested = F) = 0.8 x 0.7 = 0.56
F	T	P(No electricity = F) x P(Not Interested = T) = 0.8 x 0.3 = 0.24
T	F	P(No electricity = T) x P(Not Interested = F) = 0.2 x 0.7 = 0.14
T	T	P(No electricity = T) x P(Not Interested = T) = 0.2 x 0.3 = 0.06

Probabilistic reasoning over time

The updated Bayesian Network is:



T/F	P
F	0.8
T	0.2

T/F	P
F	0.7
T	0.3

NE	NI	P(CN)
F	F	0.56
F	T	0.24
T	F	0.14
T	T	0.06

Knowledge and Reasoning Table of

- Knowledge and reasoning-Approaches and issues of knowledge reasoning-Knowledge base agents
- Logic Basics-Logic-Propositional logic-syntax ,semantics and inferences-Propositional logic- Reasoning patterns
- Unification and Resolution-Knowledge representation using rules-Knowledge representation using semantic nets
- Knowledge representation using frames-Inferences-
- Uncertain Knowledge and reasoning-Methods-Bayesian probability and belief network
- Probabilistic reasoning-Probabilistic reasoning over time
- Other uncertain techniques-Data mining-Fuzzy logic-Dempster -shafer theory

Data Mining

- In artificial intelligence and machine learning, data mining, or knowledge discovery in databases, is the nontrivial extraction of implicit, previously unknown and potentially useful information from data. Statistical methods are used that enable trends and other relationships to be identified in large databases.
- The major reason that data mining has attracted attention is due to the wide availability of vast amounts of data, and the need for turning such data into useful information and knowledge. The knowledge gained can be used for applications ranging from risk monitoring, business management, production control, market analysis, engineering, and science exploration.

In general, three types of data mining techniques are used: association, regression, and classification.

Association analysis

- Association analysis is the discovery of association rules showing attribute-value conditions that occur frequently together in a given set of data. Association analysis is widely used to identify the correlation of individual products within shopping carts.

Regression analysis

- Regression analysis creates models that explain dependent variables through the analysis of independent variables. As an example, the prediction for a product's sales performance can be created by correlating the product price and the average customer income level.

Classification and prediction

- Classification is the process of designing a set of models to predict the class of objects whose class label is unknown. The derived model may be represented in various forms, such as if-then rules, decision trees, or mathematical formulas.

Data Mining

- A decision tree is a flow-chart-like tree structure where each node denotes a test on an attribute value, each branch represents an outcome of the test, and each tree leaf represents a class or class distribution. Decision trees can be converted to classification rules.
- Classification can be used for predicting the class label of data objects. Prediction encompasses the identification of distribution trends based on the available data.

The data mining process consists of an iterative sequence of the following steps:

- Data coherence and cleaning to remove noise and inconsistent data
- Data integration such that multiple data sources may be combined
- Data selection where data relevant to the analysis are retrieved
- Data transformation where data are consolidated into forms appropriate for mining
- Pattern recognition and statistical techniques are applied to extract patterns
- Pattern evaluation to identify interesting patterns representing knowledge
- Visualization techniques are used to present mined knowledge to users

Data Mining

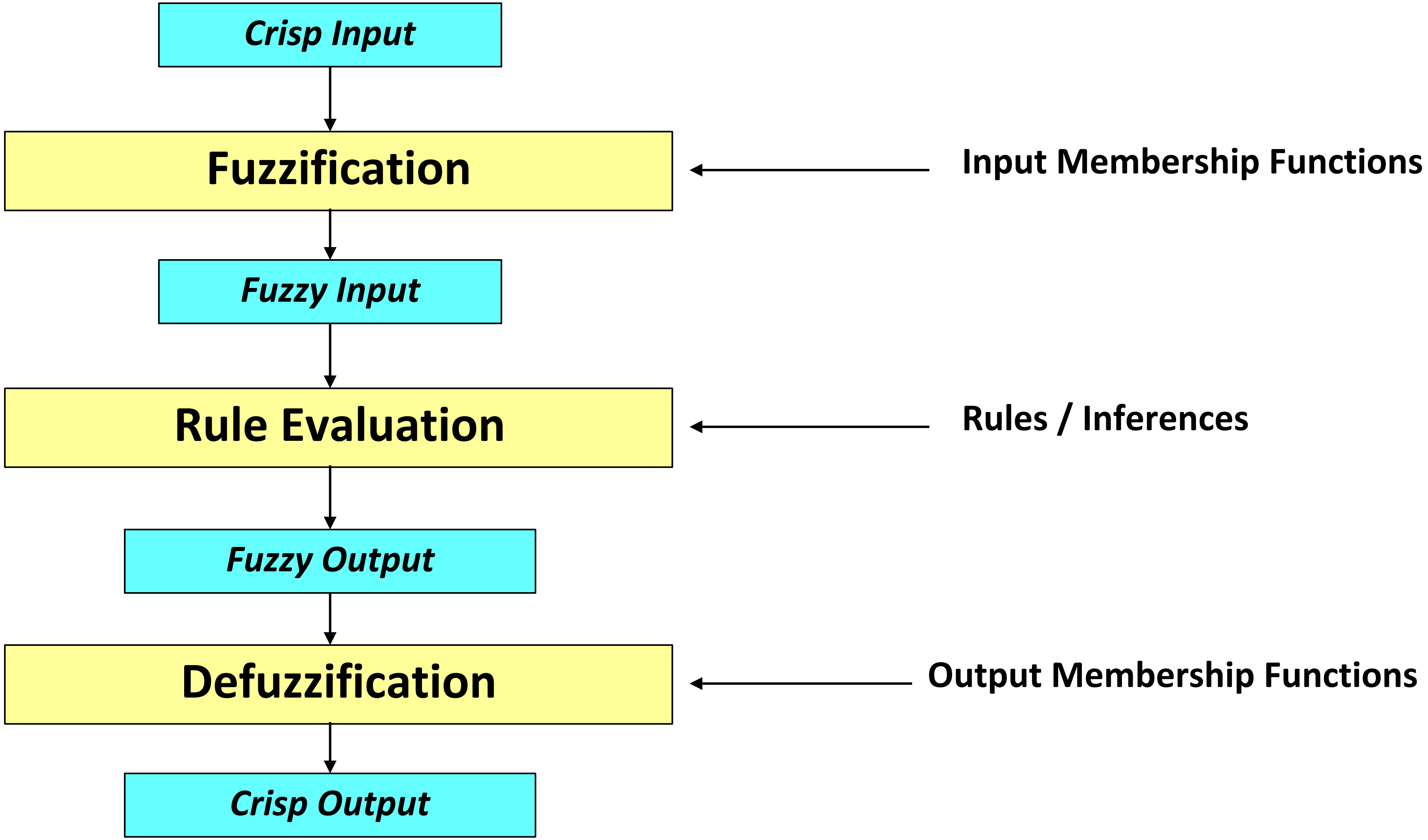
Limits of Data Mining

- GIGO (garbage in garbage out) is almost always referenced with respect to data mining, as the quality of the knowledge gained through data mining is dependent on the quality of the historical data. We know data inconsistencies and dealing with multiple data sources represent large problems in data management.
- Data cleaning techniques are used to deal with detecting and removing errors and inconsistencies to improve data quality; however, detecting these inconsistencies is extremely difficult. How can we identify a transaction that is incorrectly labeled as suspicious? Learning from incorrect data leads to inaccurate models.
- Another limitation of data mining is that it only extracts knowledge limited to the specific set of historical data, and answers can only be obtained and interpreted with regards to previous trends learned from the data.
- This limits one's ability to benefit from new trends. Because the decision tree is trained specifically on the historical data set, it does not account for personalization within the tree. Additionally, data mining (decision trees, rules, clusters) are non-incremental and do not adapt while in production.

Knowledge and Reasoning Table of

- Knowledge and reasoning-Approaches and issues of knowledge reasoning-Knowledge base agents
- Logic Basics-Logic-Propositional logic-syntax ,semantics and inferences-Propositional logic- Reasoning patterns
- Unification and Resolution-Knowledgerepresentation using rules-Knowledge representation using semantic nets
- Knowledge representation using frames-Inferences-
- Uncertain Knowledge and reasoning-Methods-Bayesian probability and belief network
- Probabilistic reasoning-Probabilistic reasoning over time
- Other uncertain techniques-Data mining-Fuzzy logic-Dempster -shafer theory

Operation of Fuzzy System



Crisp Input

Fuzzification

Input Membership Functions

Fuzzy Input

Rule Evaluation

Rules / Inferences

Fuzzy Output

Defuzzification

Output Membership Functions

Crisp Output

Building Fuzzy Systems

- Fuzzification
- Inference
- Composition
- Defuzzification

Fuzzification

- Establishes the fact base of the fuzzy system. It identifies the input and output of the system, defines appropriate IF THEN rules, and uses raw data to derive a membership function.
- Consider an air conditioning system that determine the best circulation level by sampling temperature and moisture levels. The inputs are the current temperature and moisture level. The fuzzy system outputs the best air circulation level: “none”, “low”, or “high”. The following fuzzy rules are used:

1. If the room is hot, circulate the air a lot.
2. If the room is cool, do not circulate the air.
3. If the room is cool and moist, circulate the air slightly.

- A knowledge engineer determines membership functions that map temperatures to fuzzy values and map moisture measurements to fuzzy values.

Inference

- Evaluates all rules and determines their truth values. If an input does not precisely correspond to an IF THEN rule, partial matching of the input data is used to interpolate an answer.
- Continuing the example, suppose that the system has measured temperature and moisture levels and mapped them to the fuzzy values of .7 and .1 respectively. The system now infers the truth of each fuzzy rule.
- To do this a simple method called MAX-MIN is used. This method sets the fuzzy value of the THEN clause to the fuzzy value of the IF clause. Thus, the method infers fuzzy values of 0.7, 0.1, and 0.1 for rules 1, 2, and 3 respectively.

Composition

- Combines all fuzzy conclusions obtained by inference into a single conclusion. Since different fuzzy rules might have different conclusions, consider all rules.
- Continuing the example, each inference suggests a different action
  - rule 1 suggests a "high" circulation level
  - rule 2 suggests turning off air circulation
  - rule 3 suggests a "low" circulation level.
- A simple MAX-MIN method of selection is used where the maximum fuzzy value of the inferences is used as the final conclusion. So, composition selects a fuzzy value of 0.7 since this was the highest fuzzy value associated with the inference conclusions.

Defuzzification

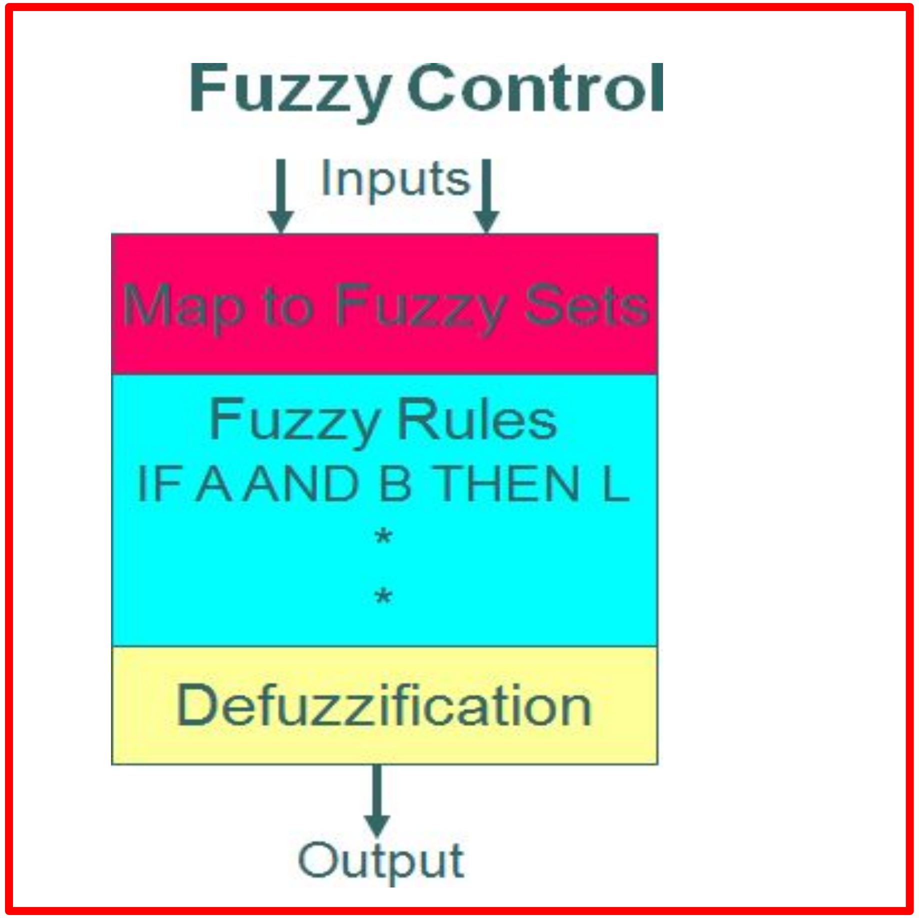
- Convert the fuzzy value obtained from composition into a “crisp” value. This process is often complex since the fuzzy set might not translate directly into a crisp value.Defuzzification is necessary, since controllers of physical systems require discrete signals.
- Continuing the example, composition outputs a fuzzy value of 0.7. This imprecise value is not directly useful since the air circulation levels are “none”, “low”, and “high”. The defuzzification process converts the fuzzy output of

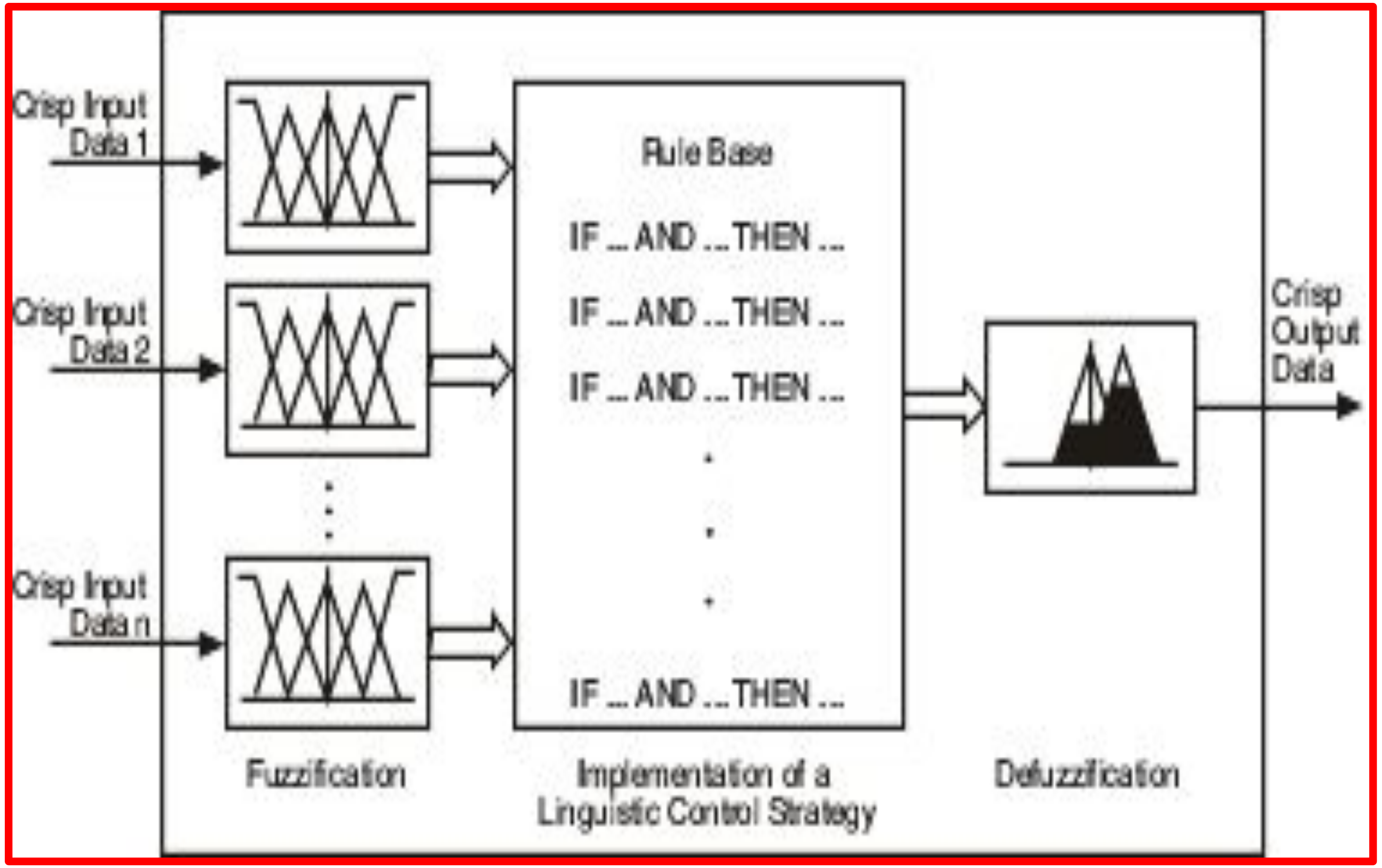
  0.7 into one of the air circulation levels. In this case it is clear that a fuzzy output of 0.7 indicates that the circulation should be set to “high”.

Defuzzification

- There are many defuzzification methods. Two of the more common techniques are the centroid and maximum methods.
- In the centroid method, the crisp value of the output variable is computed by finding the variable value of the center of gravity of the membership function for the fuzzy value.
- In the maximum method, one of the variable values at which the fuzzy subset has its maximum truth value is chosen as the crisp value for the output variable.

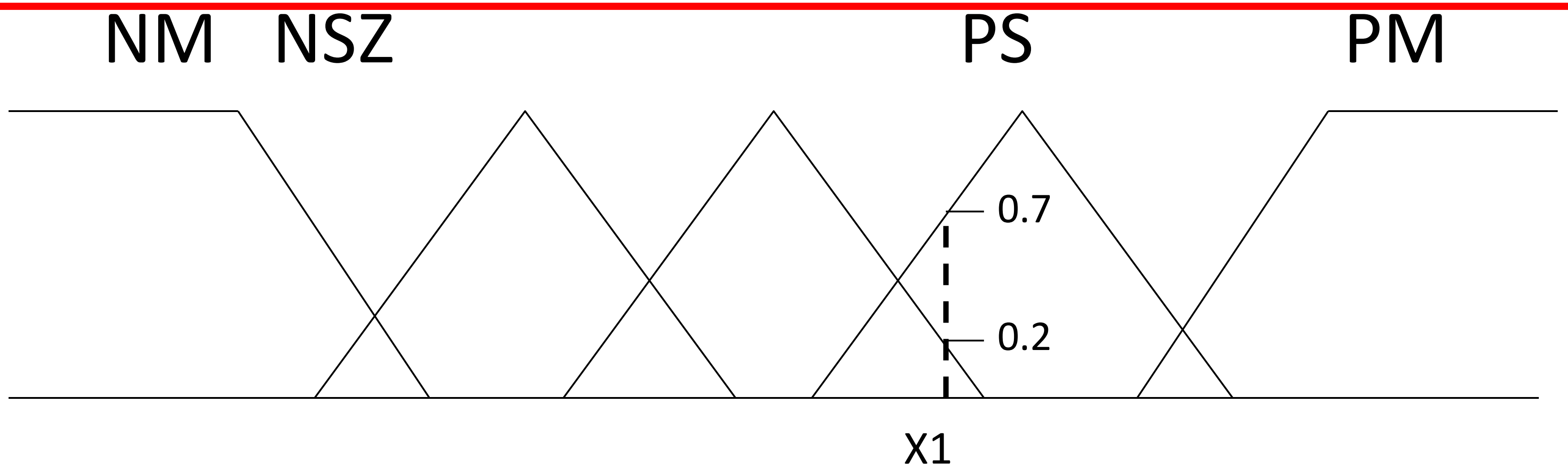
Example: Design of Fuzzy Expert System – Washing Machine





Fuzzification

Given inputs x1 and x2, find the weight values associated with each input membership function.



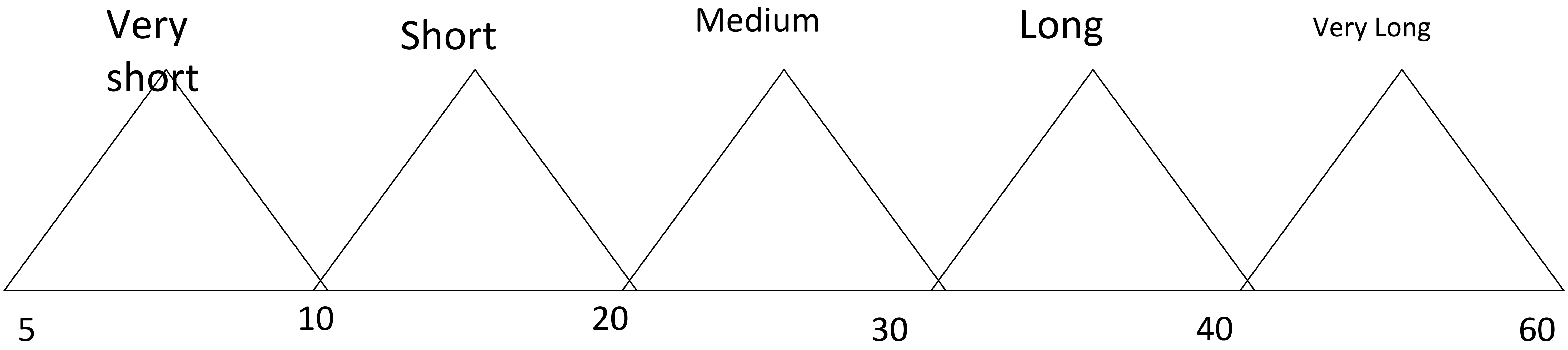
W = [0, 0, 0.2, 0.7, 0]

Fuzzy Rules

	Not Greasy	Medium	Greasy
Small Dirt	Time= Vshort	Medium	Long
Medium Dirt	Short	Medium	Long
Large Dirt	Medium	Long	Very Long

DeFuzzification

Washing Time Long = (Y- 30)/(40-30)
Washing Time Medium = (Y- 20)/(30-20)



X1 and X2 = 0.5

(Y – 20)/(30-20) = 0.5
Y – 20 = 0.5* 10 = 5
Y = 25 Mins

Knowledge and Reasoning Table of

- Knowledge andreasoning-Approachesandissues of knowledge reasoning-Knowledge base agents
- Logic Basics-Logic-Propositional logic-syntax ,semantics andinferences-Propositional logic- Reasoning patterns
- Unification andResolution-Knowledge representation using rules-Knowledge representation using semantic nets
- Knowledge representation using frames-Inferences-
- Uncertain Knowledge andreasoning-Methods-Bayesian probability andbelief network
- Probabilistic reasoning-Probabilistic reasoning over time
- Other uncertain techniques-Data mining-Fuzzy logic-Dempster -shafer theory

Dempster Shafer Theory

- **Dempster Shafer Theory** is given by Arthure P.Dempster in 1967 and his student Glenn Shafer in 1976. This theory is being released because of following reason:-
- Bayesian theory is only concerned about single evidences.
- Bayesian probability cannot describe ignorance.
- DST is an evidence theory, it combines all possible outcomes of the problem. Hence it is used to solve problems where there may be a chance that a different evidence will lead to some different result.

The **uncertainity in this model** is given by:-

- Consider all possible outcomes.
- Belief will lead to believe in some possibility by bringing out some evidence.
- Plausibility will make evidence compatibility with possible outcomes.

Dempster Shafer Theory

For eg:-

let us consider a room where four person are presented A, B, C, D(lets say) And suddenly lights out and when the lights come back B has been died due to stabbing in his back with the help of a knife. No one came into the room and no one has leaved the room and B has not committed suicide. Then we have to find out who is the murdrer?

To solve these there are the **following possibilities**:

- Either {A} or{C} or {D} has killed him.
- Either {A, C} or {C, D} or {A, C} have killed him.
- Or the three of them kill him i.e; {A, C, D}
- None of the kill him {o}(let us say).

These will be the possible evidences by which we can find the murderer by measure of plausibIlity.

Using the above example we can say :

Set of possible conclusion (P): {p1, p2….pn}

where P is set of possible conclusion and cannot be exhaustive means at least one (p)i must be true.
(p)i must be mutually exclusive.
Power Set will contain 2^n elements where n is number of elements in the possible set.
For eg:-
If P = { a, b, c}, then Power set is given as

{o, {a}, {b}, {c}, {a, b}, {b, c}, {a, c}, {a, b, c}}= 2^3 elements.

Dempster Shafer Theory

- **Mass function m(K):** It is an interpretation of m({K or B}) i.e; it means there is evidence for {K or B} which cannot be divided among more specific beliefs for K and B.
- **Belief in K:** The belief in element K of Power Set is the sum of masses of element which are subsets of K. This can be explained through an example
  Lets say K = {a, b, c}
  Bel(K) = m(a) + m(b) + m(c) + m(a, b) + m(a, c) + m(b, c) + m(a, b, c)
- **Plaausiblity in K:** It is the sum of masses of set that intersects with K.
  i.e; Pl(K) = m(a) + m(b) + m(c) + m(a, b) + m(b, c) + m(a, c) + m(a, b, c)

Characteristics of Dempster Shafer Theory:

- It will ignorance part such that probability of all events aggregate to 1.
- Ignorance is reduced in this theory by adding more and more evidences.
- Combination rule is used to combine various types of possibllities.

Dempster Shafer Theory

Advantages:

- As we add more information, uncertainty interval reduces.
- DST has much lower level of ignorance.
- Diagnose Hierarchies can be represented using this.
- Person dealing with such problems is free to think about evidences.

Disadvantages:

- In this computation effort is high, as we have to deal with 2^n of sets.

•P(

) = ({

Dempster Shafer Problem

Example: 4 people (B, J, S and K) are locked in a room when light goes out .

When light comes on, K is dead, staffed whit a knife.

Not suicide (staffed in the back)

No one entered room.

Assume only one killer

$\theta = \{B,J,S\}$

P(ϴ) = ({ $\emptyset, \{B\}, \{J\}, \{S\}, \{B,J\}, \{B,S\}, \{J,S\}, \{B,J,S\})$

Detectives after receiving the crime scene, assign mass probabilities to various elements of the power set:

Event	Mass
No one is guilty	0
B is guilty	0.1
J is guilty	0.2
S is guilty	0.1
Either B or J is guilty	0.1
Either B or S is guilty	0.1
Either S or J is guilty	0.3
One of the 3 is guilty	0.1

Dempster Shafer Problem

Belief in A:

The belief in an element A of the power set is the sum of the masses of elements which are subsets of A (including A itself)

Ex: Given A= {q1, q2, q3}

Bet (A)

={m(q1)+m(q2)+m(q3)+m(q1,q2)+m(q2,q3),m(q1,q3)+m(q1,q2,q3)}

Ex: Given the above mass assignments,

Bel(B) = m(B) =0.1

Bel (B,J) = m(B)+m(J)+m(B,J) = 0.1+0.2=0.1 0.4

RESULT:

A	{B}	{J}	{S}	{B,J}	{B,S}	{S,J}	{B,J,S}
M(A)	0.1	0.2	0.1	0.1	0.1	0.3	0.1
Bel (A)	0.1	0.2	0.1	0.4	0.3	0.6	1.0

Dempster Shafer Problem

Plausibility of A: pl(A)

The plausibility of an element A, pl(a), is the sum of all the masses of the sets that instruct with the

Set A : Ex:Pl (B,J) =M(B) +m(J)+M(B,J)+M(B,S)+M(J,S)+M(B,J,S)=0.9

~~All Result:~~

A	{B}	{J}	{S}	{B,J}	{B,S}	{S,J}	{B,J,S}
M(A)	0.1	0.2	0.1	0.1	0.1	0.3	0.1
Pl (A)	0.4	0.7	0.6	0.9	0.8	0.9	1.0

Dempster Shafer Problem

Disbelief (or Doubt) in A: dis (A)

The disbelief in A is simply bel (7A)

It is calculated by summing all masses of elements which do not intersect with

A D is (A) = 1-pl (A)

Or

Pl (A) = 1-Dis (A)

A	{B}	{J}	{S}	{B,J}	{B,S}	{S,J}	{B,J,S}
Pl(A)	0.4	0.7	0.6	0.9	0.8	0.9	1.0
Dis(A)	0.6	0.3	0.4	0.1	0.2	0.1	0.0

Belief Interval of A:

The certainty associated with a give subset A is defined by the belief interval: [bel(A)

p(A)] Ex . The belief interval of (B,S) IS [0.3,08]

A	{B}	{J}	{S}	{B,J}	{B,S}	{S,J}	{B,J,S}
M(A)	0.1	0.2	0.1	0.1	0.1	0.3	0.1
Bel (A)	0.1	0.2	0.1	0.4	0.3	0.6	1.0
Pl(A)	0.4	0.7	0.6	0.9	0.8	0.9	1.0

Dempster Shafer Problem

P(A) represents the maximum share of the evidence. We could possibly have, if for all its that intouects with A, the part that intracts actually valid. So, Pl(A) is the max possible value of prof(A).

<u>Belief intervals and Probability</u>

The probability in A falls some ware between bel (A) and pl(A).

-bel (A) represents the evidence. We have for a directly So proof (A) cannot be less than this value.

- PL(A) represents the maximum share of the evidence we could possibly have. If, for all sets that intersects with A, the part that intersects is actually valid. So, PL(A) is the max possible value of proof(A).

Belief intervals allow Dempster, Shaffer theory to reason about the degree of certainity or certainity of our beliefs.

A small difference between belief and plausibility shows that we are curtain about our belief.

A large difference shows that we are uncertain about our belief.

however, even with a 'O' interval, this does not mean we know which conclusion is right.

A	{B}	{J}	{S}	{B,J}	{B,S}	{S,J}	{B,J,S}
M(A)	0.1	0.2	0.1	0.1	0.1	0.3	0.1
Bel (A)	0.1	0.2	0.1	0.4	0.3	0.6	1.0
Pl(A)	0.4	0.7	0.6	0.9	0.8	0.9	1.0
Belief interval	{0.1,0.4}	{0.2,0.7}	{0.1,0.6}	{0.4,0.9}	{0.3,0.8}	{0.6,0.9}	{1,1}

Thank You