AI

UNIT 4

KNOWLEDGE AND REASONING

Prepared by: Dr. Nilima Kulkarni

Acknowledgement

This Ppt's are prepared using

1. Stuart Russell and Peter Norvig (1995), "Articial Intelligence: A Modern Approach", Third edition, Pearson, 2003.

2. Parag Kulkarni and Prachi Joshi, "Artificial Intelligence Building Intelligent Systems", PHI learning Pvt. Ltd., ISBN 978-81-203-5046-5, 2015. 5.

3. Online source

Outline

- Knowledge-based agents
- Wumpus world
- Logic in general
- Propositional and first-order logic
  - Inference, validity, equivalence and satisfiability
  - Reasoning patterns
    - Resolution
    - Forward/backward chaining

“Thinking Rationally”

- Computational models of human “thought” processes
- Computational models of human behavior
- Computational systems that “think” rationally
- Computational systems that behave rationally

Introduction

- Humans, it seems, know things and do reasoning
- Problem solving agents perform well in complex environments
- A reflex agent could only find its way from Arad to Bucharest by dumb luck
- The knowledge of problem-solving agents is, however, very specific and inflexible
- E.g. Chess program can calculate the legal moves of its king, but does not know in any useful sense that no piece can be on two different squares at the same time

Introduction

- Knowledge and reasoning also play a crucial role in dealing with *partially observable* environments
- A knowledge-based agent can combine general knowledge with current percepts to infer hidden aspects of the current state prior to selecting actions
- For example, a physician diagnoses a patient-that is, infers a disease state that is not directly observable prior to choosing a treatment
- Some of the knowledge that the physician uses is in the form of rules learned from textbooks and teachers, and some is in the form of patterns of association that the physician may not be able to consciously describe.
- If it's inside the physician's head, it counts as knowledge

Representing knowledge using rules

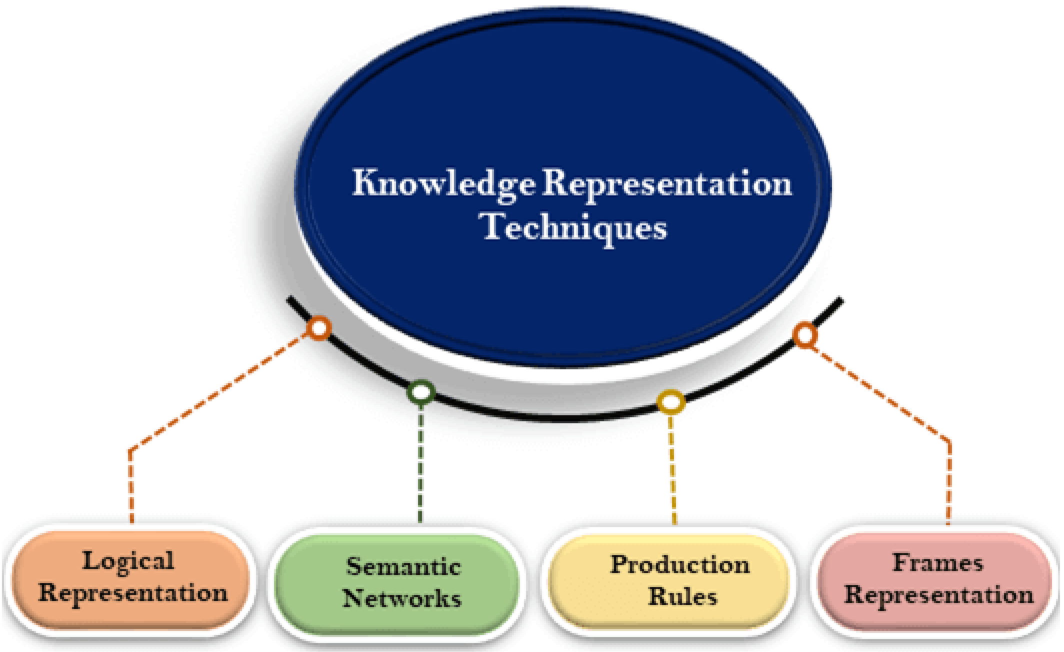
We can represent the knowledge in terms of

1. Propositional logic
2. predicate logic
3. using rules
4. frame system
5. Semantic nets

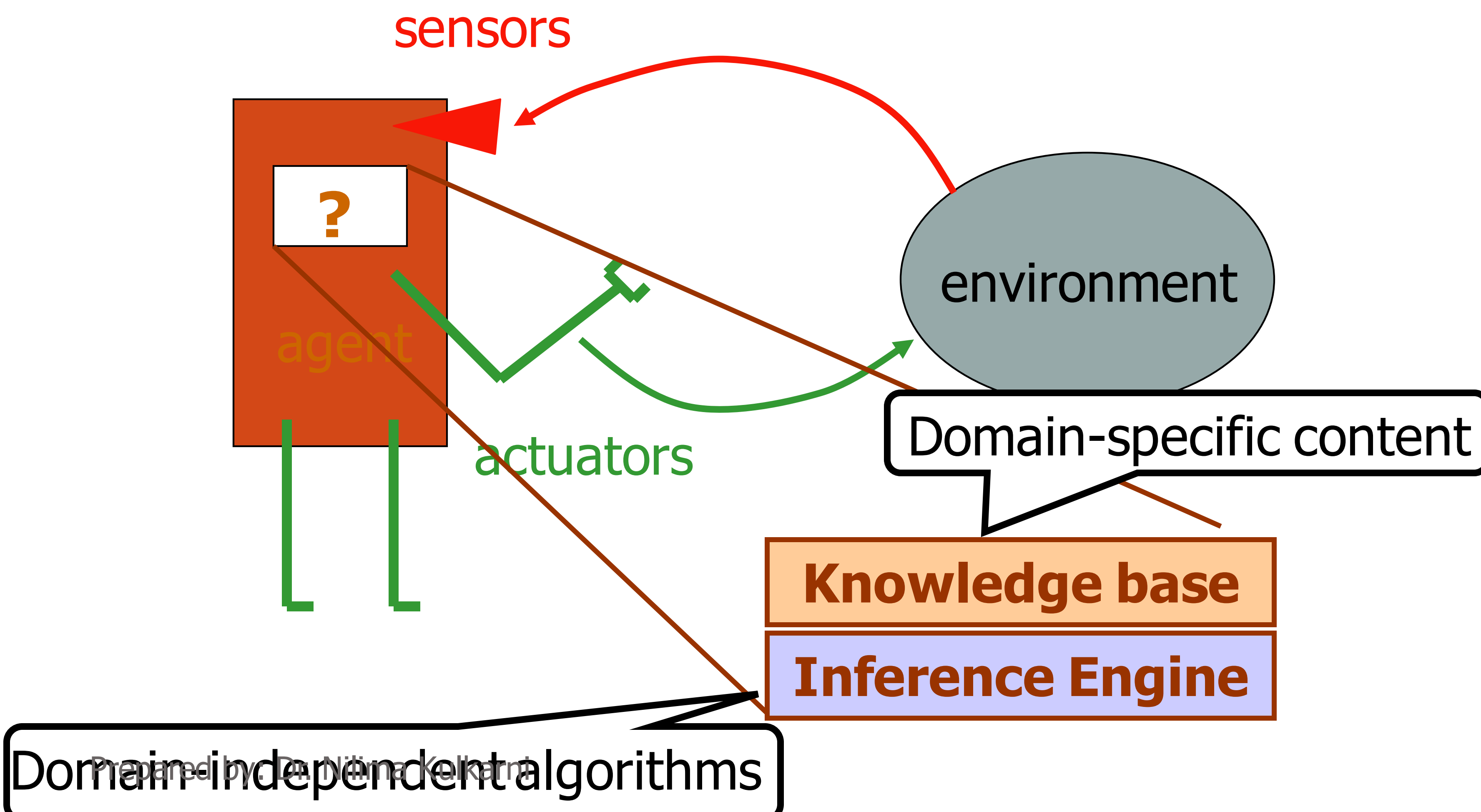
Representing knowledge using rules

We can represent the knowledge in terms of

1. Propositional logic
2. predicate logic
3. using rules
4. frame system
5. Semantic nets

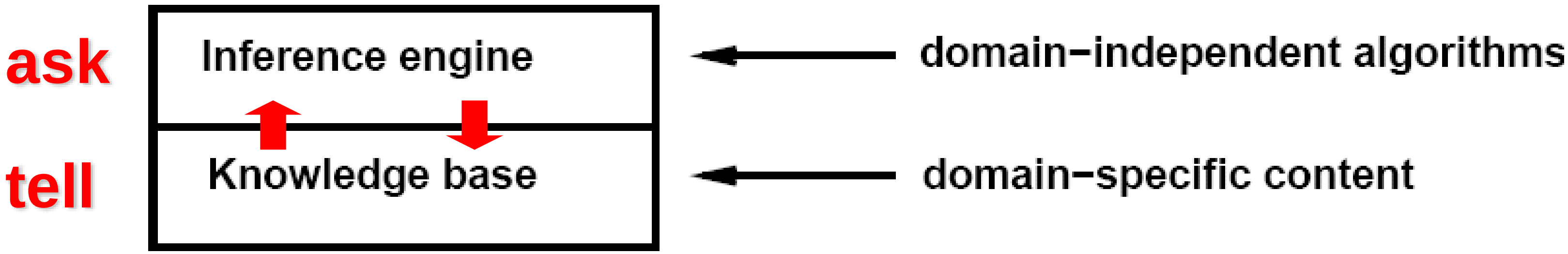


Knowledge-Based Agent



Knowledge Base

Knowledge Base: set of sentences represented in a knowledge representation language and represents assertions about the world.



Inference rule: when one ASKs questions of the KB, the answer should *follow* from what has been TELLed to the KB previously.

Abilities KB agent

- Agent must be able to:
  - Represent states and actions,
  - Incorporate new percepts
  - Update internal representation of the world
  - Deduce hidden properties of the world
  - Deduce appropriate actions

Types of Knowledge

- Procedural, e.g.: functions
  Such knowledge can only be used in one way -- by executing it

- Declarative, e.g.: constraints and rules
  It can be used to perform many different sorts of inferences

The Wumpus World

The Wumpus World

- 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

Wumpus PEAS description

- **Performance measure**:
  gold +1000, death -1000,
  -1 per step, -10 use arrow
- **Environment:**
  Squares adjacent to wumpus are smelly
  Squares adjacent to pit are breezy
  Glitter iff gold is in the same square
  Bump iff move into a wall
  Woeful scream iff the wumpus is killed
  Shooting kills wumpus if you are facing it
  Shooting uses up the only arrow
  Grabbing picks up gold if in same square
  Releasing drops the gold in same square
- **Sensors:** Stench, Breeze, Glitter, Bump, Scream
- **Actuators:** Let turn, Right turn, Forward, Grab, Release, Shoot

Wumpus PEAS description

Agent in a Wumpus world: Performance Measure

- The agents goal is to find the gold and bring it back to the start as quickly as possible without getting killed
- +1000 points reward for climbing out of the cave with the gold
- -1 point deducted for every action taken
- -1000 points penalty for falling into a pit or being eaten by the wumpus

Agent in a Wumpus world: Environment

- A 4×4 grid of rooms
- The agent always starts in the square [1,1], facing to the right
- The locations of the gold and the wumpus are chosen randomly
- In addition, each square other than the start can be a pit, with probability 0.2

Agent in a Wumpus world: Actuators

- **go forward**
- **turn right** 90 degrees
- **turn left** 90 degrees
- **grab** means pick up an object that is in the same square as the agent
- **shoot** means fire an arrow in a straight line in the direction the agent is looking.
  - The arrow continues until it either hits and kills the wumpus or hits the wall.
  - The agent has only one arrow.
  - Only the first shot has any effect.
- **climb** is used to leave the cave.
  - Only effective in start field.
- **die**, if the agent enters a square with a pit or a live wumpus.
  - (No take-backs!)

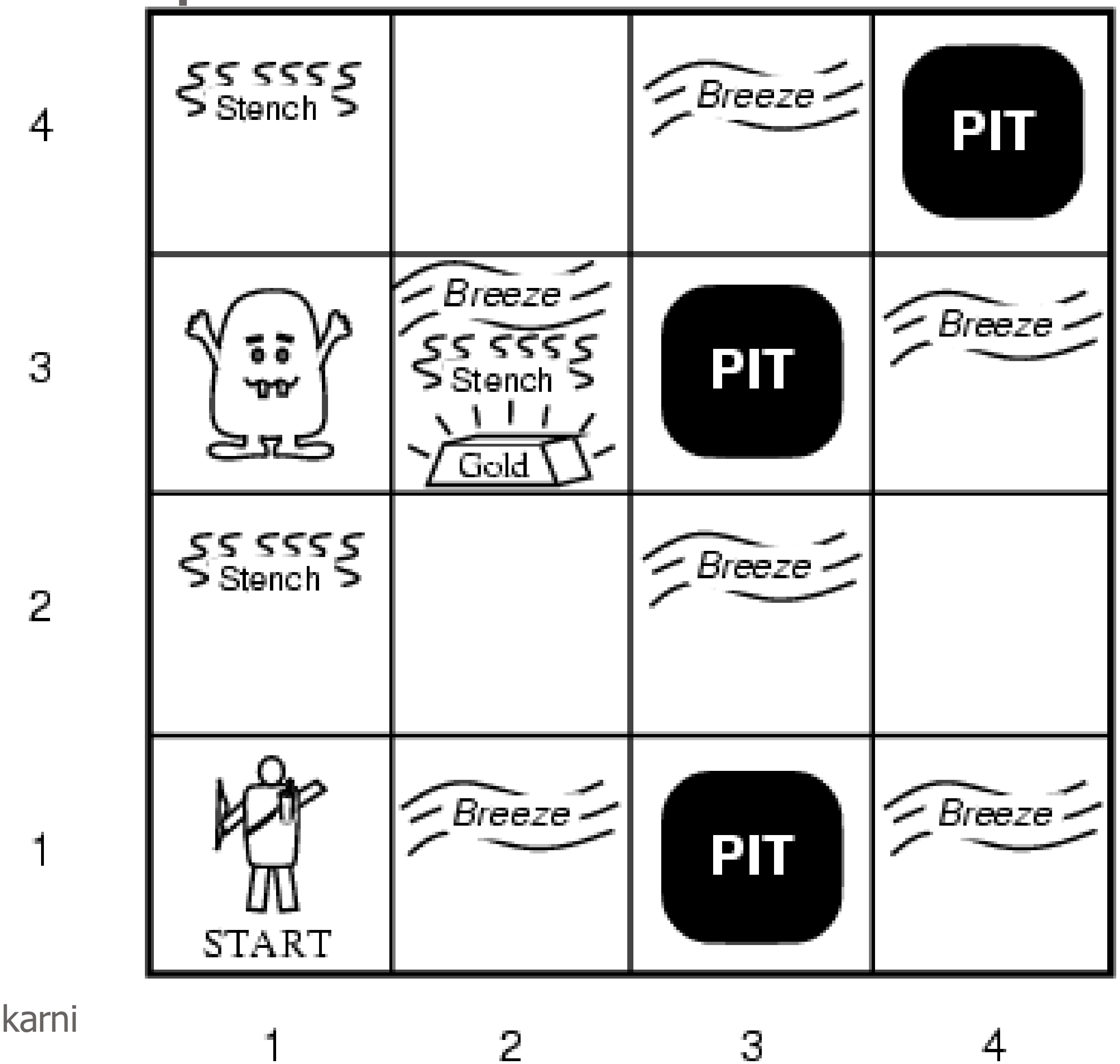
Agent in a Wumpus world: Sensors

- 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 scream everywhere in the cave, if the wumpus is killed
- The percepts will be 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]
- The agent can not perceive its own location.

A typical Wumpus world

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

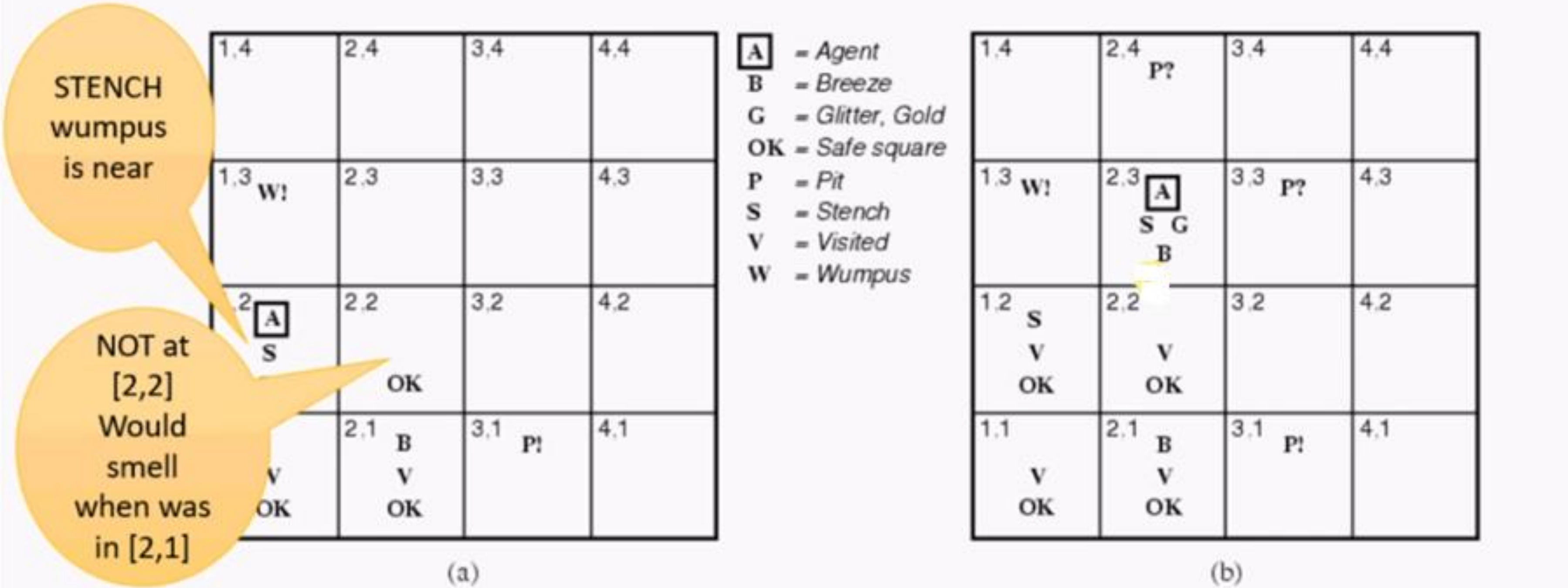


The Wumpus agent's first step

The first step taken by the agent in the wumpus world. (a) The initial situation, after percept [None, None, None, None, None]. (b) After one move, with percept [None, Breeze, None, None, None]

Percept: Stench, breeze, glitter, bump, scream

After Third and Fifth move



Two later stages in the progress of the agent. (a) After the third move, with percept [Stench, None, None, None, None]. (b) After the fifth move, with percept [Stench, Breeze, Glitter, None, None].

Percept: Stench, breeze, glitter, bump, scream

After Third and Fifth move

Two later stages in the progress of the agent. (a) After the third move, with percept [Stench, None, None, None, None]. (b) After the fifth move, with percept [Stench, Breeze, Glitter, None, None].

Percept: Stench, breeze, glitter, bump, scream

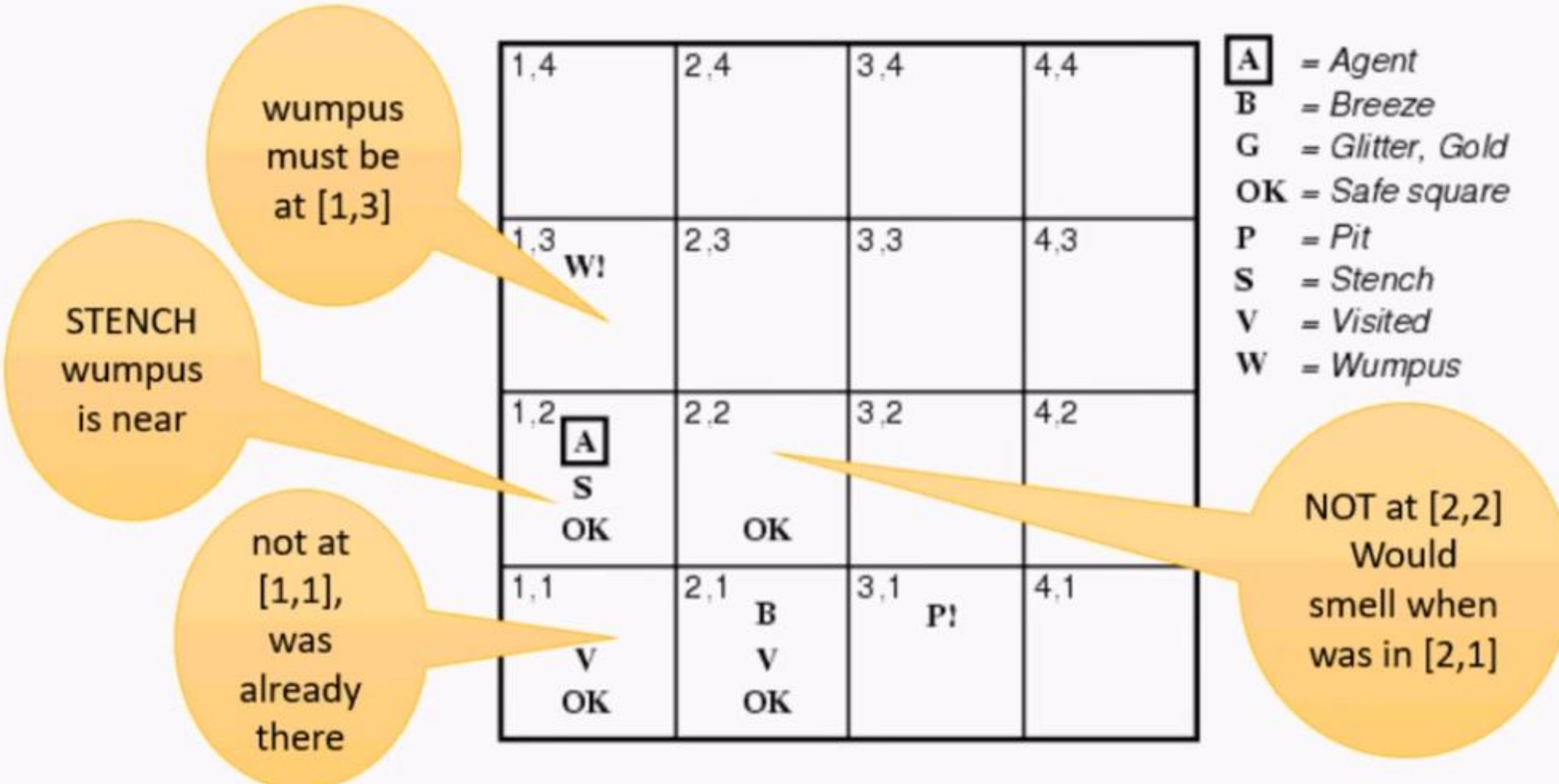
The Wumpus world: Environment

- **Fully observable ?**
  NO, the agent perceives just its direct neighbour (partially observable)
- **Deterministic ?**
  YES, the result of action is given
- **Episodic ?**
  NO, the order of actions is important (sequential)
- **Static ?**
  YES, the wumpus and pits do not move
- **Discrete ?**
  YES
- **One agent ?**
  YES, the wumpus does not act as an agent, it is merely a property of environment

Some Inference rules

RULE	PREMISE	CONCLUSION
Modus Ponens	$A, A \rightarrow B$	B
And Introduction	A, B	$A \land B$
And Elimination	$A \land B$	A
Modus Tollens	$\neg B, A \rightarrow B$	$\neg A$
Unit Resolution	$A \lor B, \neg B$	A
Resolution	$A \lor B, \neg B \lor C$	$A \lor C$

After the third move



Rules and Atomic Propositions

- Some Atomic Propositions

  S12 = There is a stench in cell (1,2)

  B34 = There is a breeze in cell (3,4)

  W22 = Wumpus is in cell (2,2)

  V11 = We've visited cell (1,1)

  OK11 = Cell (1,1) is safe etc.

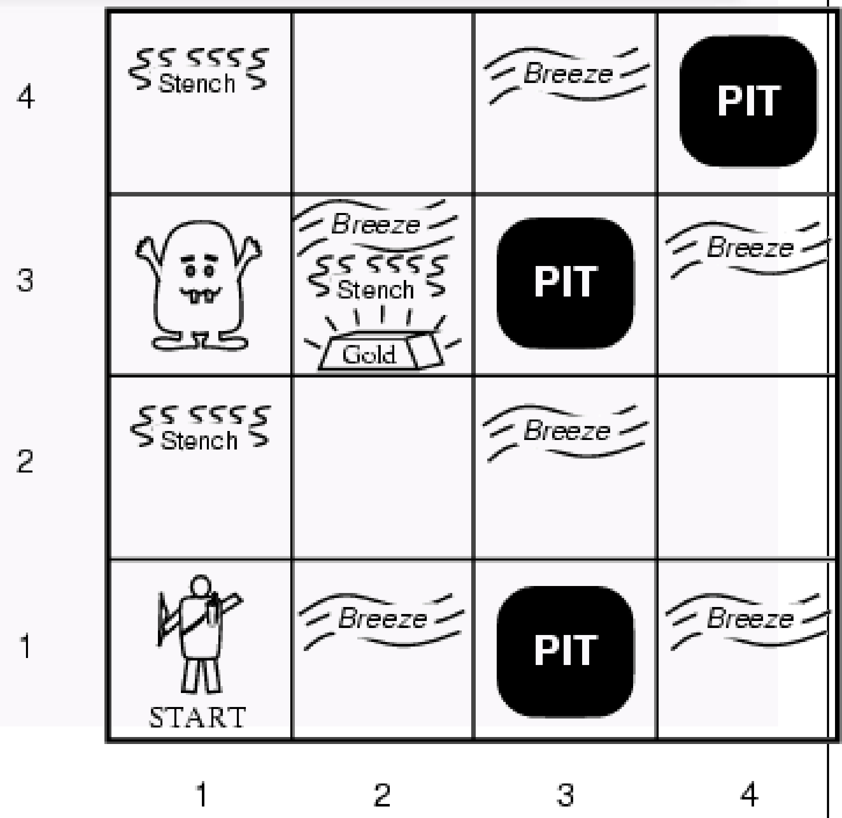
- Some rules

  (R1) $\neg S11 \rightarrow \neg W11 \wedge \neg W12 \wedge \neg W21$

  (R2) $\neg S21 \rightarrow \neg W11 \wedge \neg W21 \wedge \neg W22 \wedge \neg W31$

  (R3) $\neg S12 \rightarrow \neg W11 \wedge \neg W12 \wedge \neg W22 \wedge \neg W13$

  (R4) $S12 \rightarrow W13 \vee W12 \vee W22 \vee W11$ etc.



Prove that the Wumpus is in (1,3)

Apply MP with $\neg S11$ and R1:

$\neg W11 \wedge \neg W12 \wedge \neg W21$

(R1) $\neg S11 \rightarrow \neg W11 \wedge \neg W12 \wedge \neg W21$

$\neg S11$

$\neg W11 \wedge \neg W12 \wedge \neg W21$

Apply And-Elimination to this we get 3 sentences:

$\neg W11, \neg W12, \neg W21$

Prove that the Wumpus is in (1,3)

Apply MP to ¬ S21 and R2, then apply And-elimination:

¬W22, ¬W21, ¬W31

(R2) ¬S21 → ¬W11 ∧ ¬W21 ∧ ¬W22 ∧ ¬W31

¬S21

¬W22 , ¬W21 , ¬W31

Prove that the Wumpus is in (1,3)

Apply MP to S12 and R4 to obtain:

$W13 \vee W12 \vee W22 \vee W11$

(R4) $S12 \rightarrow W13 \vee W12 \vee W22 \vee W11$

$S12$

$(W13 \vee W12 \vee W22 \vee W11)$

Prove that the Wumpus is in (1,3)

Apply Unit resolution on (W13 ∨ W12 ∨ W22 ∨ W11) and ¬W11:

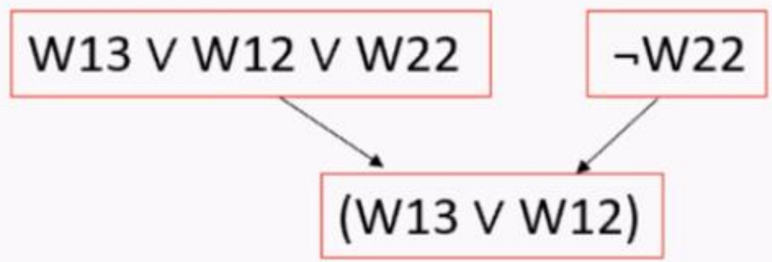
W13 ∨ W12 ∨ W22

$$
\begin{array}{ccc}
\boxed{W13 \lor W12 \lor W22 \lor W11} & & \boxed{\neg W11} \\
& \searrow \quad \swarrow & \\
& \boxed{(W13 \lor W12 \lor W22)} &
\end{array}
$$

Prove that the Wumpus is in (1,3)

Apply Unit Resolution with (W13 ∨ W12 ∨ W22) and ¬W22:

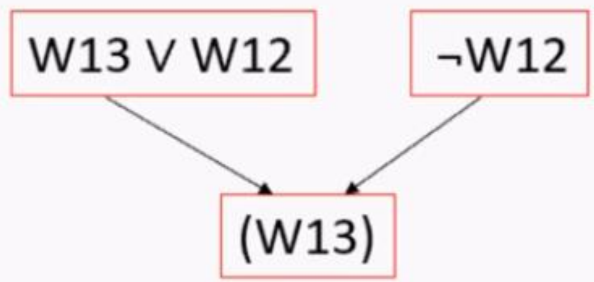
W13 ∨ W12



Prove that the Wumpus is in (1,3)

Apply Unit Resolution with (W13 ∨ W12) and ¬W12:

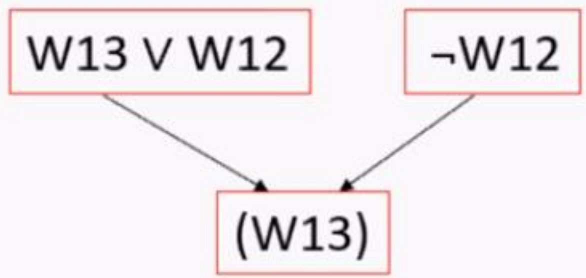
W13



Prove that the Wumpus is in (1,3)

Apply Unit Resolution with (W13 ∨ W12) and ¬W12:

W13



Proved

Propositional Logic

- Propositional logic (PL) is the simplest form of logic where all the statements are made by propositions.
- A proposition is a declarative statement which is either true or false.
- It is a technique of knowledge representation in logical and mathematical form.
- **Example:**

  a) It is Sunday.

  b) The Sun rises from West (False proposition)

  c) 3+3= 7(False proposition)

  d) 5 is a prime number.

Syntax of propositional logic

- The syntax of propositional logic defines the allowable sentences for the knowledge representation. There are two types of Propositions:

  Atomic Propositions

  Compound propositions

- **Atomic Proposition:** Atomic propositions are the simple propositions. It consists of a single proposition symbol. These are the sentences which must be either true or false.
  - Example:

  a) 2+2 is 4, it is an atomic proposition as it is a true fact.

  b) "The Sun is cold" is also a proposition as it is a false fact.

Syntax of propositional logic

- **Compound proposition:** Compound propositions are constructed by combining simpler or atomic propositions, using parenthesis and logical connectives.
- Example:

  a) "It is raining today, and street is wet."

  b) "Ankit is a doctor, and his clinic is in Mumbai."

Logical Connectives

- Logical connectives are used to connect two simpler propositions or representing a sentence logically. We can create compound propositions with the help of logical connectives. There are mainly five connectives, which are given as follows:
- **Negation:** A sentence such as ¬ P is called negation of P. A literal can be either Positive literal or negative literal.
- **Conjunction:** A sentence which has **∧** connective such as, **P ∧ Q** is called a conjunction.
  Example: Rohan is intelligent and hardworking. It can be written as,
  P= Rohan is intelligent,
  Q= Rohan is hardworking. → **P∧ Q**.

Logical Connectives

- **Disjunction:** A sentence which has V connective, such as **P V Q**. is called disjunction, where P and Q are the propositions.
  Example: "Ritika is a doctor or Engineer",
  Here P= Ritika is Doctor. Q= Ritika is Doctor, so we can write it as **P V Q**.
- **Implication:** A sentence such as $P \rightarrow Q$, is called an implication. Implications are also known as if-then rules. It can be represented as

  If it is raining, then the street is wet.

  Let P= It is raining, and Q= Street is wet, so it is represented as $P \rightarrow Q$

Logical Connectives

- **Biconditional:** A sentence such as **P⇔ Q is a Biconditional sentence, example If I am breathing, then I am alive**

  P= I am breathing, Q= I am alive, it can be represented as P ⇔ Q.

Propositional logic: Syntax

- Propositional logic is the simplest logic – illustrates basic ideas

- The proposition symbols P_1, P_2 etc are sentences

  - If S is a sentence, $\neg S$ is a sentence (negation)
  - If S_1 and S_2 are sentences, $S_1 \wedge S_2$ is a sentence (conjunction)
  - If S_1 and S_2 are sentences, $S_1 \vee S_2$ is a sentence (disjunction)
  - If S_1 and S_2 are sentences, $S_1 \Rightarrow S_2$ is a sentence (implication)
  - If S_1 and S_2 are sentences, $S_1 \Leftrightarrow S_2$ is a sentence (biconditional)

Syntax of Propositional Logic

- sentence → atomic sentence | complex sentence
- atomic sentence → Propositional symbol, *True*, *False*
- Complex sentence → ¬sentence
  | (sentence ∧ sentence)
  | (sentence ∨ sentence)
  | (sentence ⇒ sentence)
- Examples:
  - $((P \wedge Q) \Rightarrow R)$
  - $(A \Rightarrow B) \vee (\neg C)$

Order of Precedence

Precedence	Operators
First Precedence	Parenthesis
Second Precedence	Negation
Third Precedence	Conjunction(AND)
Fourth Precedence	Disjunction(OR)
Fifth Precedence	Implication
Six Precedence	Biconditional

Semantics of Propositional Logic

- It specifies how to determine the truth value of any sentence in a model m
- The truth value of *True* is *True*
- The truth value of *False* is *False*
- The truth value of each atomic sentence is given by m
- The truth value of every other sentence is obtained recursively by using truth tables

Truth Tables

A	B	$\neg A$	$A \wedge B$	$A \vee B$	$A \Rightarrow B$
True	*True*	*False*	*True*	*True*	*True*
True	*False*	*False*	*False*	*True*	*False*
False	*False*	*True*	*False*	*False*	*True*
False	*True*	*True*	*False*	*True*	*True*

Wumpus world sentences

Let $P_{i,j}$ be true if there is a pit in [i, j].

Let $B_{i,j}$ be true if there is a breeze in [i, j].

start: $\neg P_{1,1}$

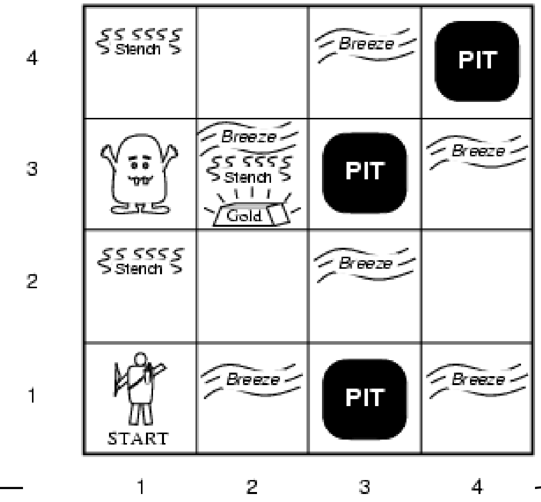
$\neg B_{1,1}$

$B_{2,1}$

- "Pits cause breezes in adjacent squares"

$B_{1,1} \Leftrightarrow (P_{1,2} \lor P_{2,1})$

$B_{2,1} \Leftrightarrow (P_{1,1} \lor P_{2,2} \lor P_{3,1})$



Logical equivalence

- To manipulate logical sentences we need some rewrite rules.
- Two sentences are logically equivalent iff they are true in same models: $\alpha \equiv \beta$ iff $\alpha \models \beta$ and $\beta \models \alpha$

You need to know these !

$$
\begin{array}{rcll}
(\alpha \wedge \beta) & \equiv & (\beta \wedge \alpha) & \text{commutativity of } \wedge \\
(\alpha \vee \beta) & \equiv & (\beta \vee \alpha) & \text{commutativity of } \vee \\
((\alpha \wedge \beta) \wedge \gamma) & \equiv & (\alpha \wedge (\beta \wedge \gamma)) & \text{associativity of } \wedge \\
((\alpha \vee \beta) \vee \gamma) & \equiv & (\alpha \vee (\beta \vee \gamma)) & \text{associativity of } \vee \\
\neg(\neg\alpha) & \equiv & \alpha & \text{double-negation elimination} \\
(\alpha \Rightarrow \beta) & \equiv & (\neg\beta \Rightarrow \neg\alpha) & \text{contraposition} \\
(\alpha \Rightarrow \beta) & \equiv & (\neg\alpha \vee \beta) & \text{implication elimination} \\
(\alpha \Leftrightarrow \beta) & \equiv & ((\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)) & \text{biconditional elimination} \\
\neg(\alpha \wedge \beta) & \equiv & (\neg\alpha \vee \neg\beta) & \text{de Morgan} \\
\neg(\alpha \vee \beta) & \equiv & (\neg\alpha \wedge \neg\beta) & \text{de Morgan} \\
(\alpha \wedge (\beta \vee \gamma)) & \equiv & ((\alpha \wedge \beta) \vee (\alpha \wedge \gamma)) & \text{distributivity of } \wedge \text{ over } \vee \\
(\alpha \vee (\beta \wedge \gamma)) & \equiv & ((\alpha \vee \beta) \wedge (\alpha \vee \gamma)) & \text{distributivity of } \vee \text{ over } \wedge
\end{array}
$$

Rules of Inference in Artificial intelligence

- **Inference:**
- In artificial intelligence, we need intelligent computers which can create new logic from old logic or by evidence, **so generating the conclusions from evidence and facts is termed as Inference**.
- **Inference rules:**
- Inference rules are the templates for generating valid arguments. Inference rules are applied to derive proofs in artificial intelligence, and the proof is a sequence of the conclusion that leads to the desired goal.
- In inference rules, the implication among all the connectives plays an important role. Following are some terminologies related to inference rules:

Rules of Inference in Artificial intelligence

- **Implication:** It is one of the logical connectives which can be represented as $P \rightarrow Q$. It is a Boolean expression.
- **Converse:** The converse of implication, which means the right-hand side proposition goes to the left-hand side and vice-versa. It can be written as $Q \rightarrow P$.
- **Contrapositive:** The negation of converse is termed as contrapositive, and it can be represented as $\neg Q \rightarrow \neg P$.
- **Inverse:** The negation of implication is called inverse. It can be represented as $\neg P \rightarrow \neg Q$.

Rules of Inference in Artificial intelligence

P	Q	$P \to Q$	$Q \to P$	$\neg Q \to \neg P$	$\neg P \to \neg Q.$
T	T	T	T	T	T
T	F	F	T	F	T
F	T	T	F	T	F
F	F	T	T	T	T

Types of Inference rules:

- **There are many inference rules**
- **1. Modus Ponens:**
- The Modus Ponens rule is one of the most important rules of inference, and it states that if P and P → Q is true, then we can infer that Q will be true. It can be represented as:
- **Example:**
- Statement-1: "If I am sleepy then I go to bed" ==> P→ Q
  Statement-2: "I am sleepy" ==> P
  Conclusion: "I go to bed." ==> Q.
  Hence, we can say that, if P→ Q is true and P is true then Q will be true.

Types of Inference rules:

- **Proof by Truth table:** **Proof by Truth table:**

Modus Ponens

$$\big(p \wedge (p \rightarrow q)\big) \rightarrow q$$

$$\begin{array}{ll} & p \\ & \underline{p \rightarrow q} \\ \therefore & q \end{array}$$

p	q	$p \rightarrow q$	$p \wedge (p \rightarrow q)$	$\big(p \wedge (p \rightarrow q)\big) \rightarrow q$
0	0	1	0	1
0	1	1	0	1
1	0	0	0	1
1	1	1	1	1

Reasoning Patterns

- Modus Ponens

$$\frac{\alpha \Rightarrow \beta, \qquad \alpha}{\beta}$$

- And-Elimination

$$\frac{\alpha \wedge \beta}{\alpha}$$

Rule 1: Modus Ponens

" If Tweety is a bird then Tweety flies. "

" Tweety is a bird. "

Therefore, Tweety flies .

Rule 1: Modus Ponens

" If Tweety is a bird then Tweety flies. "
" Tweety is a bird. "

Therefore, Tweety flies .

Notations for M.P.

$$\frac{P \rightarrow Q,\ P}{Q}$$

And elimination rule:

If the conjunction *A and B* is true, then *A* is true, and *B* is true.

The rule makes it possible to shorten longer proofs by deriving one of the conjuncts of a conjunction on a line by itself.

An example in English:

It's raining and it's pouring.
Therefore it's raining.

The rule consists of two separate sub-rules, which can be expressed in formal language as:

$$\frac{P \wedge Q}{\therefore P}$$

and

$$\frac{P \wedge Q}{\therefore Q}$$

First-Order logic / Predicate Logic

In the topic of Propositional logic, we have seen that how to represent statements using propositional logic.

But unfortunately, in propositional logic, we can only represent the facts, which are either true or false.

PL is not sufficient to represent the complex sentences or natural language statements.

The propositional logic has very limited expressive power. Consider the following sentence, which we cannot represent using PL logic.

"Some humans are intelligent", or
"Sachin likes cricket."

First-Order logic / Predicate Logic

- First-order logic is another way of knowledge representation in artificial intelligence. It is an extension to propositional logic.

- FOL is sufficiently expressive to represent the natural language statements in a concise way.

- First-order logic is also known as **Predicate logic or First-order predicate logic**.

- First-order logic is a powerful language that develops information about the objects in a more easy way and can also express the relationship between those objects.

- **First-order logic (like natural language) does not only assume that the world contains facts like propositional logic but also assumes the following things in the world**

First-Order logic:

1. **Objects:** A, B, people, numbers, colors, wars, theories, squares, pits, wumpus, ...
2. **Relations: It can be unary relation such as:** red, round, is adjacent, **or n-any relation such as:** the sister of, brother of, has color, comes between
3. **Function:** Father of, best friend, third inning of, end of,…

•As a natural language, first-order logic also has two main parts:

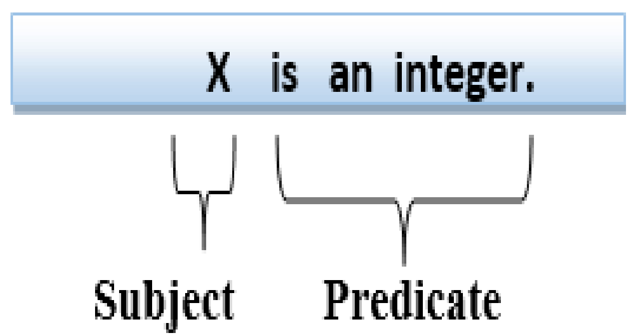
- **Syntax**
- **Semantics**

Basic Elements of First-order logic:

Following are the basic elements of FOL syntax:

Constant	1, 2, A, John, Mumbai, cat,....
Variables	x, y, z, a, b,....
Predicates	Brother, Father, >,....
Function	sqrt, LeftLegOf, ...
Connectives	$\wedge$, $\vee$, $\neg$, $\Rightarrow$, $\Leftrightarrow$
Equality	==
Quantifier	$\forall$, $\exists$

FIRST ORDER Logics



Atomic sentences:

- Atomic sentences are the most basic sentences of first-order logic. These sentences are formed from a predicate symbol followed by a parenthesis with a sequence of terms.
- We can represent atomic sentences as **Predicate (term1, term2, ..., term n)**.

Example: Ravi and Ajay are brothers: => Brothers(Ravi, Ajay).

Chinky is a cat: => cat (Chinky).

Complex Sentences:

- Complex sentences are made by combining atomic sentences using connectives.

First-order logic statements can be divided into two parts:

- **Subject:** Subject is the main part of the statement.
- **Predicate:** A predicate can be defined as a relation, which binds two atoms together in a statement.

Quantifiers in First-order logic:

A quantifier is a language element which generates quantification, and quantification specifies the quantity of specimen in the universe of discourse.

These are the symbols that permit to determine or identify the range and scope of the variable in the logical expression.

There are two types of quantifier

1. **Universal Quantifier, (for all, everyone, everything)**
2. **Existential quantifier, (for some, at least one).**

Quantifiers in First-order logic:

Universal Quantifier:

Universal quantifier is a symbol of logical representation, which specifies that the statement within its range is true for everything or every instance of a particular thing.

The Universal quantifier is represented by a symbol ∀, which resembles an inverted A.

Note: In universal quantifier we use implication "→".

If x is a variable, then ∀x is read as:

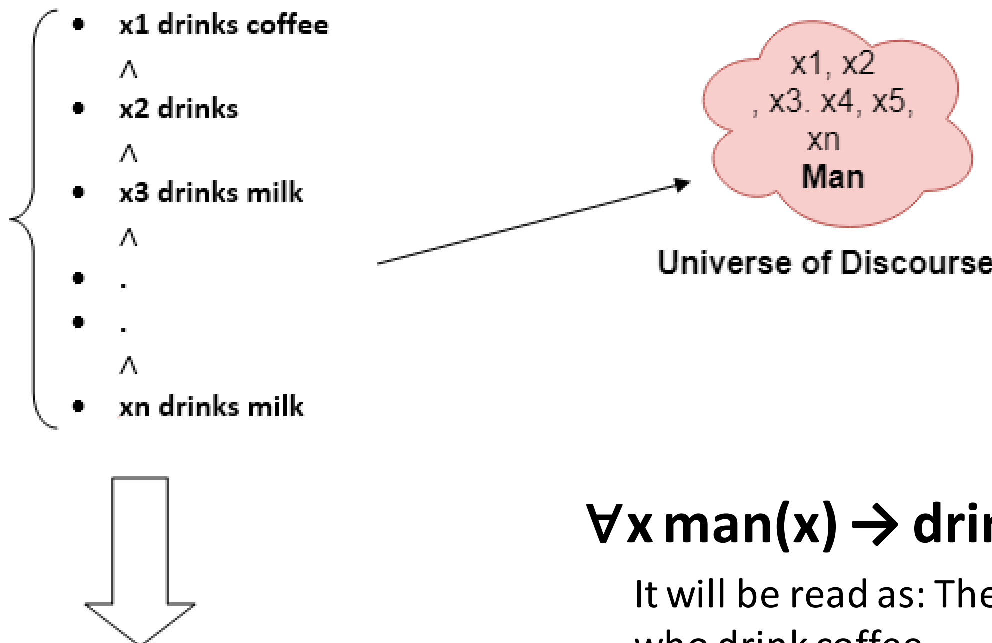
For all x
For each x
For every x.

Quantifiers in First-order logic:

Example:

All man drink coffee.

Let a variable x which refers to a cat so all x can be represented in UOD as below:



So in shorthand notation, we can write it as :

$\forall x\ man(x) \rightarrow drink(x, coffee)$.

It will be read as: There are all x where x is a man who drink coffee.

Quantifiers in First-order logic:

Existential Quantifier:

Existential quantifiers are the type of quantifiers, which express that the statement within its scope is true for at least one instance of something.

It is denoted by the logical operator ∃, which resembles as inverted E. When it is used with a predicate variable then it is called as an existential quantifier.

Note: In Existential quantifier we always use AND or Conjunction symbol (∧).

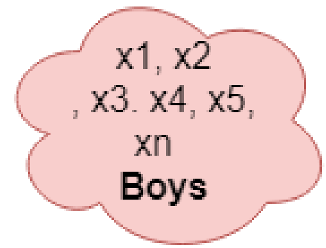
If x is a variable, then existential quantifier will be ∃x or ∃(x). And it will be read as:

There exists a 'x.'
For some 'x.'
For at least one 'x.'

Quantifiers in First-order logic:

Example:
Some boys are intelligent.

- **x1 is intelligent**
  V
- **x2 is intelligentV**
- **x3 is intelligent**
  V
- .
- .
  V
- **xn is intelligent**



Universe of Discourse

So in short-hand notation, we can write it as:

∃x: boys(x) ∧ intelligent(x)

It will be read as: There are some x where x is a boy who is intelligent.

Unification Algorithm

Algorithm: Unify(L1, L2)

1. If L1 or L2 are both variables or constants then:

    a. If L1 and L2 are identical then return NIL.

    b. Else if L1 is a variable then if L1 occurs in L2 then return {FAIL}, else return (L2/L1)

    c. Else if L2 is a variable then if L2 occurs in L1 then return {FAIL}, else return (L1/L2)

    d. Else return {FAIL}

2. If the initial predicate symbols in L1 and L2 are not identical then return {FAIL}

3. If L1 and L2 have a different number of arguments then return {FAIL}

Unification Algorithm

4. Set SUBST to NIL. At the end of this procedure SUBST will contain all the substitutions used to unify L1 and L2

5. For i <- 1 to number of arguments in L1:

   a. Call unify with the ith argument of L1 and ith argument of L2 putting result in S

   b. If S contains FAIL then return {FAIL}

   c. If S is not equal to NIL then:

      i. Apply S to the remainder of both L1 and L2

      ii. SUBST:=APPEND(S,SUBST).

6. Return SUBST

Example

Representing knowledge using rules

We can represent the knowledge in terms of

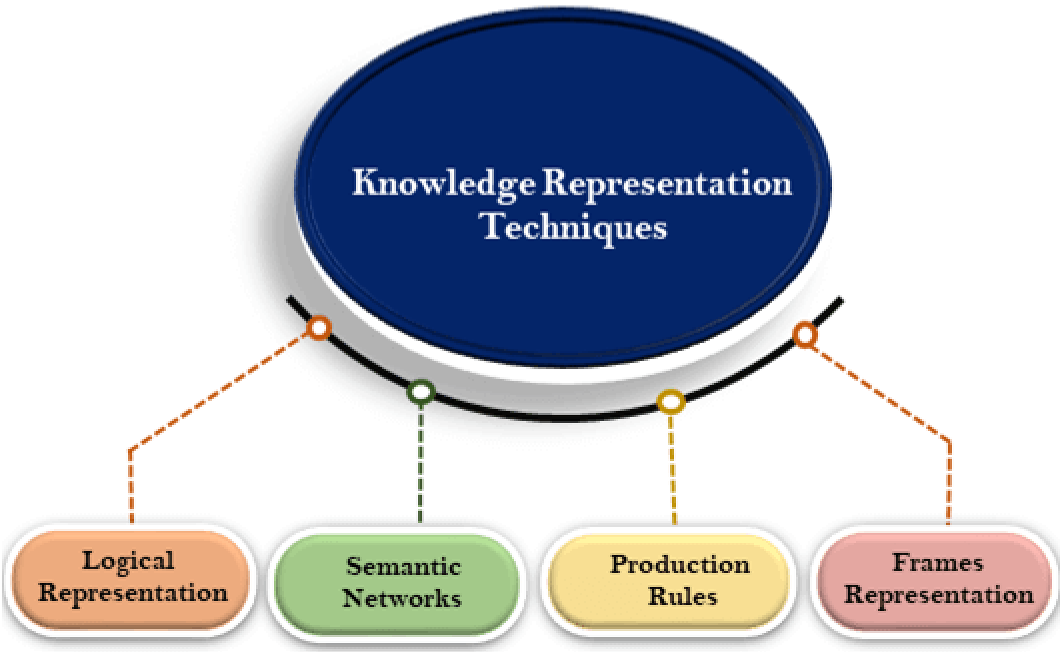
1. Propositional logic
2. predicate logic
3. using rules
4. frame system
5. Semantic nets

Example

Representing knowledge using rules

We can represent the knowledge in terms of

1. Propositional logic
2. predicate logic
3. using rules
4. frame system
5. Semantic nets



Semantic nets

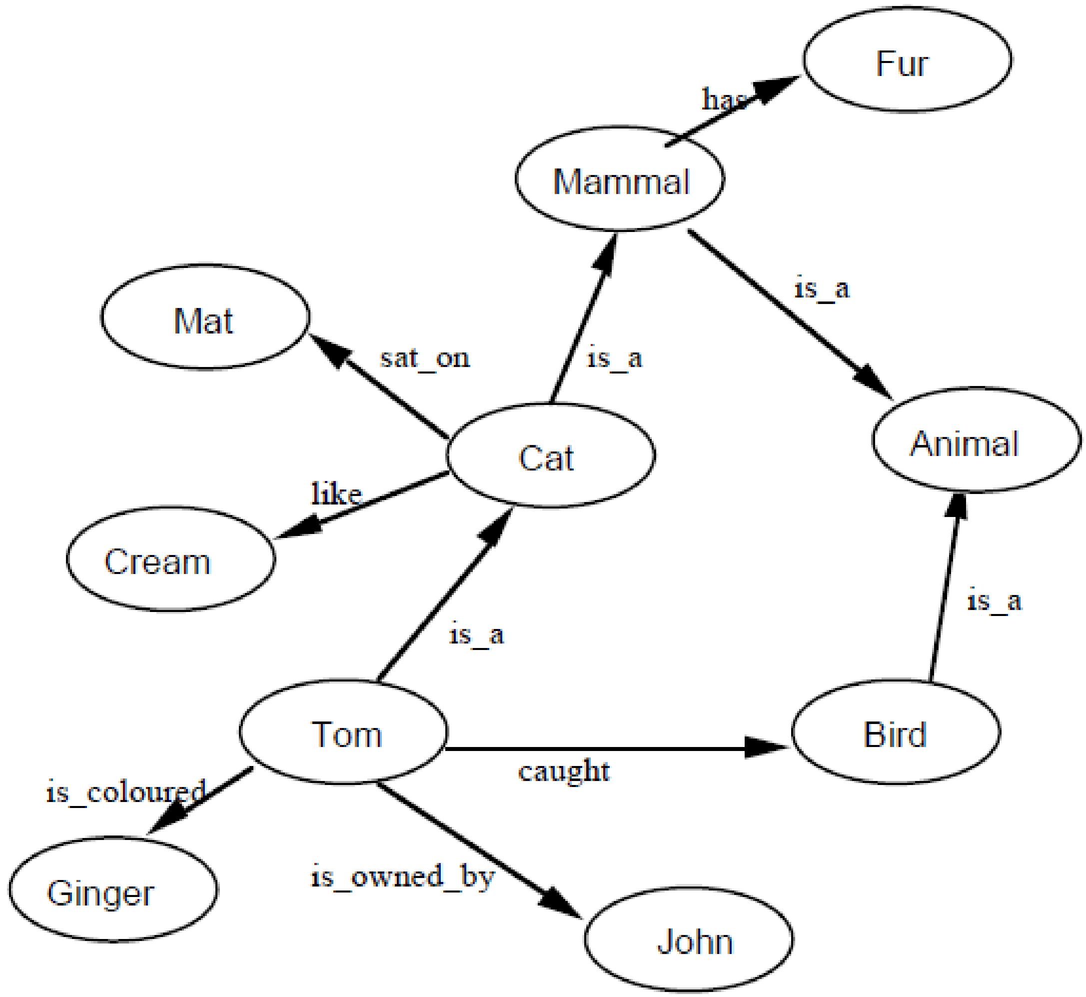
- Semantic networks are an alternative to predicate logic as a form of knowledge representation.
- The idea is that we can store our knowledge in the form of a graph, **with nodes representing objects in the world, and arcs representing relationships between those objects.**
- This network consists of nodes representing objects and arcs which describe the relationship between those objects.
- Semantic networks can categorize the object in different forms and can also link those objects. Semantic networks are easy to understand and can be easily extended.
- This representation consist of mainly two types of relations:
  - **IS-A relation (Inheritance)**
  - **Kind-of-relation**
- **Example:** Following are some statements which we need to represent in the form of nodes and arcs.

Semantic nets Example

- *Tom is a cat.*
- *Tom caught a bird.*
- *Tom is owned by John.*
- *Tom is ginger in colour.*
- *Cats like cream.*
- *The cat sat on the mat.*
- *A cat is a mammal.*
- *A bird is an animal.*
- *All mammals are animals.*
- *Mammals have fur.*

Semantic nets Example

- *Tom is a cat.*
- *Tom caught a bird.*
- *Tom is owned by John.*
- *Tom is ginger in colour.*
- *Cats like cream.*
- *The cat sat on the mat.*
- *A cat is a mammal.*
- *A bird is an animal.*
- *All mammals are animals.*
- *Mammals have fur.*



Frame Representation

- A frame is a record like structure which consists of a collection of attributes and its values to describe an entity in the world.
- Frames are the AI data structure which divides **knowledge into substructures by representing stereotypes situations.**
- **It consists of a collection of slots and slot values**. These slots may be of any type and sizes.
- **Slots have names and values which are called facets.**

Frame Representation

- **Facets:** The various aspects of a slot is known as **Facets**. Facets are features of frames which enable us to put constraints on the frames
- A frame may consist of any number of slots, and a slot may include any number of facets and facets may have any number of values.
- A frame is also known as **slot-filter knowledge representation** in artificial intelligence.

Frame Representation

Let's take an example of a frame for a book

Slots	**Filters**
Title	Artificial Intelligence
Branch	Computer Science
Author	Peter Norvig
Edition	Third Edition
Year	1996
Page	1152

Frame Representation

Frame: (Person details)

Person name

Value is keyword

Slots (abc)
Profession (value Engineer)
Age (value 25)
City (value pune)
State (value Maharashtra)

Frames:

Example 1:

Employee Details

```
Raj Sharma(
        (Profession        (Value    Manager))
        (EmpID             (Value    100213))
        (Address           (Value    Delhi))
)
```

Uncertainty and methods

Prepared by: Dr. Nilima Kulkarni

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.

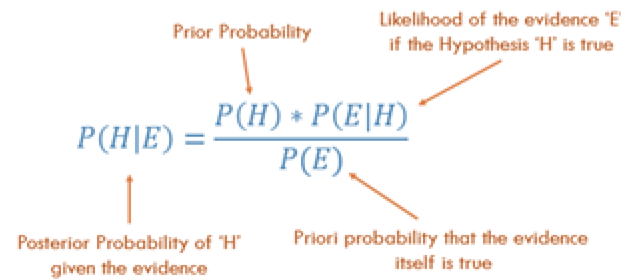
- Information occurred from unreliable sources.
- Experimental Errors
- Equipment fault
- Temperature variation
- Climate change.

Probabilistic reasoning:

- Probabilistic reasoning 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.

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:
  1. **Bayes' rule**
  2. **Bayesian Statistics**



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:
  1. **Bayes' rule**
  2. **Bayesian Statistics**

Bayesian Probability and belief network
Probabilistic Reasoning

How to build network models to reason under uncertainty according to the laws of the probability theory

Probability Basics

- Begin with a set S: the sample space
  - e.g., 6 possible rolls of a die.
- x ϵ S is a sample point/possible world/atomic event
- A probability space or probability model is a sample space with an assignment P(x) for every x s.t. $0 \leq P(x) \leq 1$ and $\sum P(x) = 1$
- An event A is any subset of S
  - e.g. A= 'die roll < 4'
- A random variable is a function from sample points to some range, e.g., the reals or Booleans

Types of Probability Spaces

Propositional or Boolean random variables

e.g., $Cavity$ (do I have a cavity?)

Discrete random variables (*finite* or *infinite*)

e.g., $Weather$ is one of $\langle sunny, rain, cloudy, snow\rangle$

$Weather = rain$ is a proposition

Values must be exhaustive and mutually exclusive

Continuous random variables (*bounded* or *unbounded*)

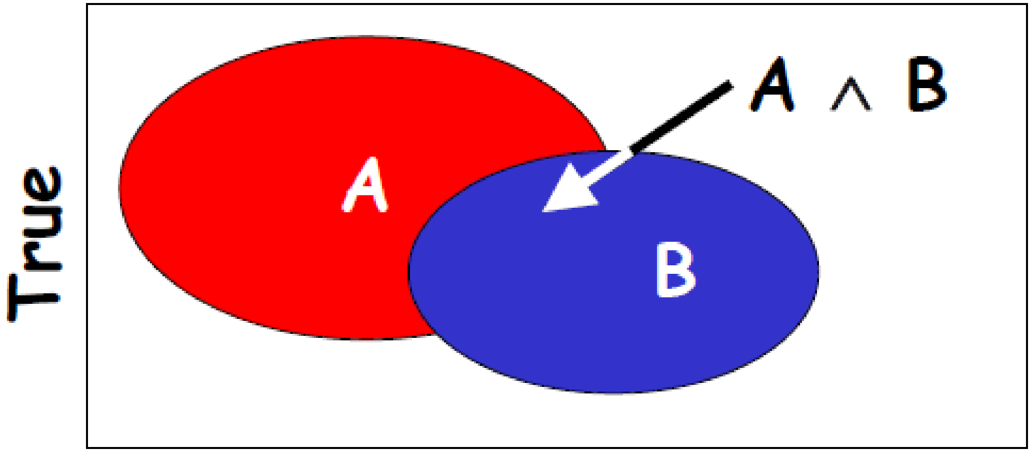
e.g., $Temp = 21.6$; also allow, e.g., $Temp < 22.0$.

Arbitrary Boolean combinations of basic propositions

Axioms of Probability Theory

- All probabilities between 0 and 1
  - $0 \le P(A) \le 1$
  - $P(true) = 1$
  - $P(false) = 0.$
- The probability of disjunction is:

$$P(A \vee B) = P(A) + P(B) - P(A \wedge B)$$



Prior Probability

Prior or unconditional probabilities of propositions

e.g., $P(Cavity = true) = 0.1$ and $P(Weather = sunny) = 0.72$

correspond to belief prior to arrival of any (new) evidence

Probability distribution gives values for all possible assignments:

$\mathbf{P}(Weather) = \langle 0.72, 0.1, 0.08, 0.1 \rangle$ (*normalized*, i.e., sums to 1)

Joint probability distribution for a set of r.v.s gives the probability of every atomic event on those r.v.s

$\mathbf{P}(Weather, Cavity) =$ a 4×2 matrix of values:

Joint distribution can answer any question

Prior Probability

Prior or unconditional probabilities of propositions

e.g., $P(Cavity = true) = 0.1$ and $P(Weather = sunny) = 0.72$

correspond to belief prior to arrival of any (new) evidence

Probability distribution gives values for all possible assignments:

$\mathbf{P}(Weather) = \langle 0.72, 0.1, 0.08, 0.1 \rangle$ (*normalized*, i.e., sums to 1)

Joint probability distribution for a set of r.v.s gives the probability of every atomic event on those r.v.s

$\mathbf{P}(Weather, Cavity) =$ a 4×2 matrix of values:

$Weather =$	$sunny$	$rain$	$cloudy$	$snow$
$Cavity = true$	0.144	0.02	0.016	0.02
$Cavity = false$	0.576	0.08	0.064	0.08

Conditional probability

- Conditional or posterior probabilities

  e.g., P(*cavity* | *toothache*) = 0.8

  i.e., given that *toothache* is all I know there is 80% chance of cavity

Motivations

- Full joint probability distribution can answer any question but can become intractably large as number of variable increases
- Specifying probabilities for atomic events can be difficult, e.g., large set of data, statistical estimates, etc.
- Independence and conditional independence reduce the probabilities needed for full joint probability distribution.

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 Rule

Bayes rules!

posterior

$$P(x,y) = P(x \mid y)P(y) = P(y \mid x)P(x)$$

$$\boxed{P(x \mid y) = \frac{P(y \mid x)\ P(x)}{P(y)} = \frac{\text{likelihood} \cdot \text{prior}}{\text{evidence}}}$$

Useful for assessing diagnostic probability from causal probability:

$$P(Cause|Effect) = \frac{P(Effect|Cause)P(Cause)}{P(Effect)}$$

Computing Diagnostic Prob. from Causal Prob.

$$P(Cause|Effect) = \frac{P(Effect|Cause)P(Cause)}{P(Effect)}$$

E.g. let M be meningitis, S be stiff neck

P(M) = 0.0001,
P(S) = 0.1,
P(S|M)= 0.8

P(M|S) $= \frac{P(s|m)P(m)}{P(s)} = \frac{0.8 \times 0.0001}{0.1} = 0.0008$

Note: posterior probability of meningitis still very small!

Application of Bayes' theorem in Artificial intelligence:
Following are some applications of Bayes' theorem:

1. It is used to calculate the next step of the robot when the already executed step is given.

2. Bayes' theorem is helpful in weather forecasting.

3. It can solve the Monty Hall problem.

"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 networks

- A directed, acyclic graph (DAG)
- A set of nodes, one per variable (discrete or continuous)
- A set of directed links (arrows) connects pairs of nodes. *X* is a parent of *Y* if there is an arrow (direct influence) from node *X* to node *Y*.
- Each node X_i has a conditional probability distribution that quantifies the effect of the parents on the node.

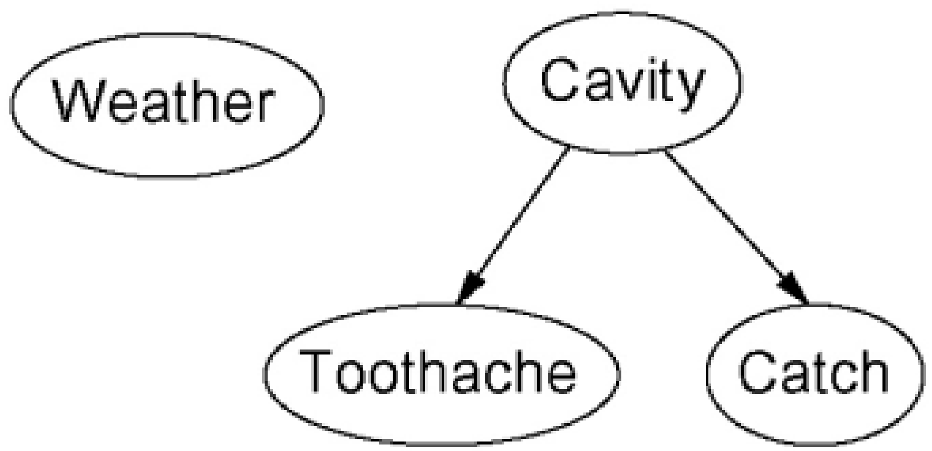
$$P\left(X_i \mid Parents(X_i)\right)$$

- Combinations of the topology and the conditional distributions specify (implicitly) the full joint distribution for all the variables.

Bayesian networks

Example 1： The Teeth Disease Bayesian

Topology of network encodes conditional independence assertions:



Weather is independent of the other variables

Toothache and *Catch* are conditionally independent given *Cavity*

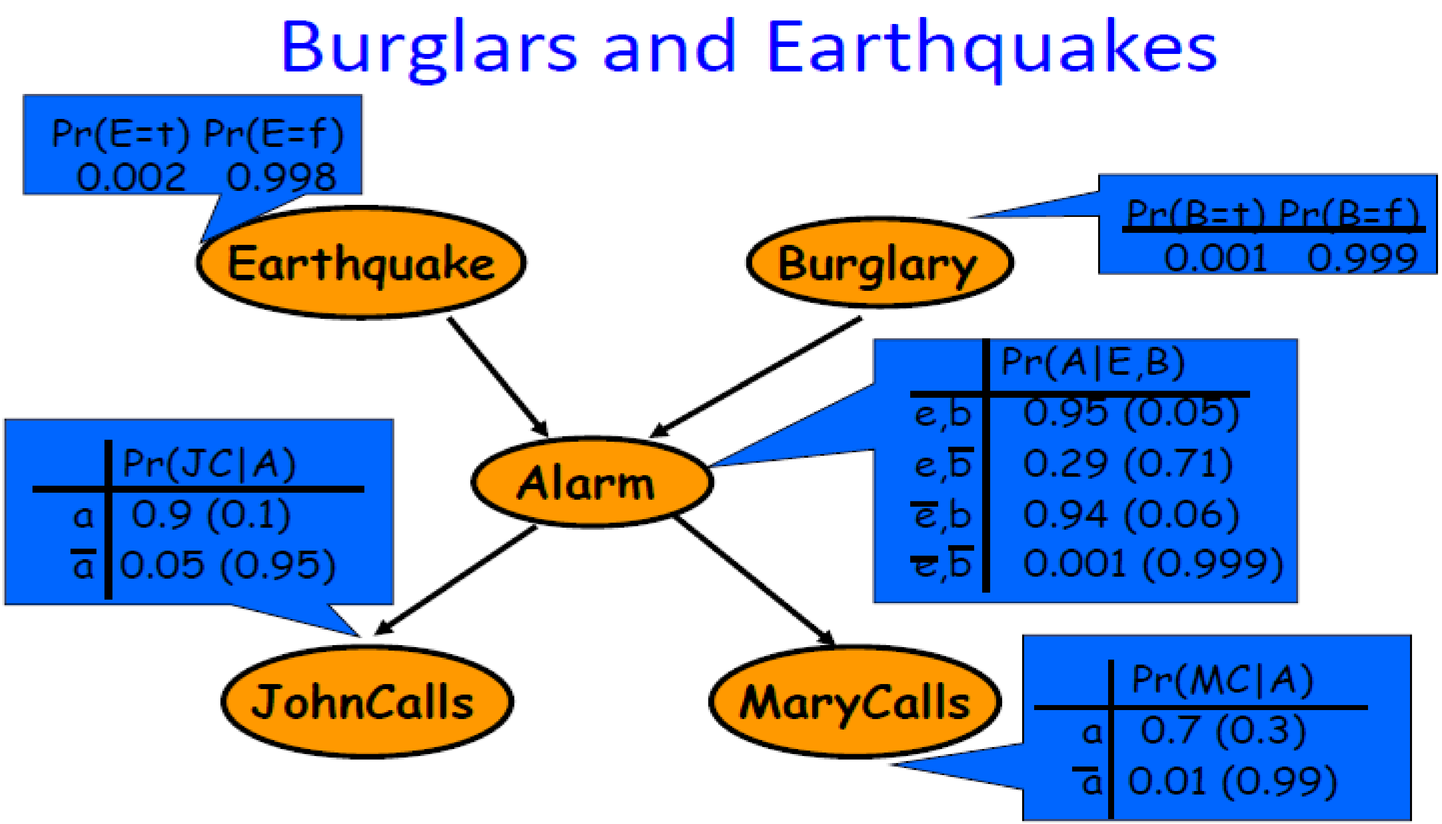
Example: Burglar alarm system

- I have a burglar alarm installed at home
  - It is fairly reliable at detecting a burglary, but also responds on occasion to minor earth quakes.
- I also have two neighbors, John and Mary
  - They have promised to call me at work when they hear the alarm
  - John always calls when he hears the alarm, but sometimes confuses the telephone ringing with the alarm and calls then, too.
  - Mary likes rather loud music and sometimes misses the alarm altogether.
- Bayesian networks variables:
  - *Burglar*, *Earthquake*, *Alarm*, *JohnCalls*, *MaryCalls*

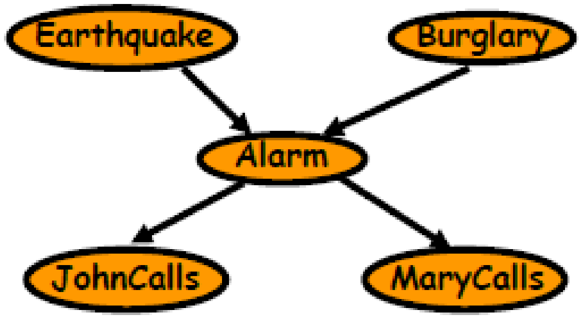
Example: Burglar alarm system

- Network topology reflects "causal" knowledge:
  - A burglar can set the alarm on
  - An earthquake can set the alarm on
  - The alarm can cause Mary to call
  - The alarm can cause John to call

conditional probability table (CPT): each row contains the conditional probability of each node value for a conditioning case (a possible combination values for the parent nodes).



Earthquake Example (cont'd)



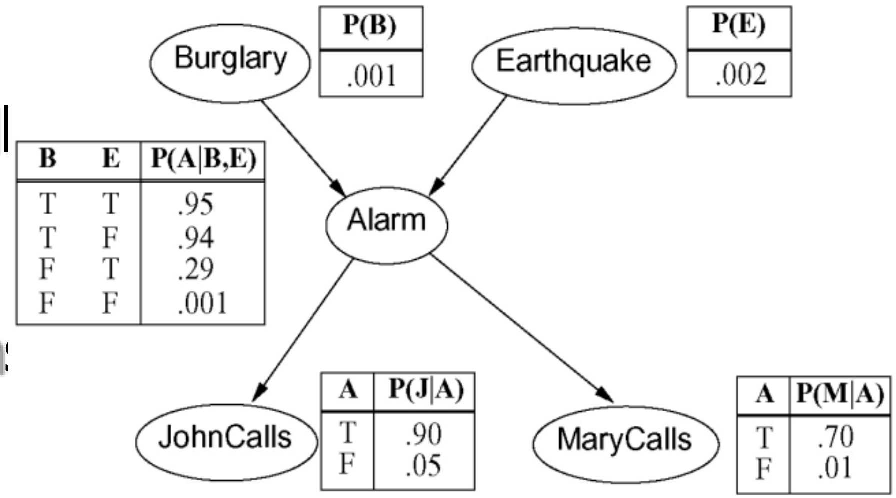
- **If we know *Alarm*, no other evidence influences our degree of belief in *JohnCalls***
  - $P(JC|MC,A,E,B) = P(JC|A)$
  - also: $P(MC|JC,A,E,B) = P(MC|A)$ and $P(E|B) = P(E)$
- **By the chain rule we have**

$$P(JC,MC,A,E,B) = P(JC|MC,A,E,B) \cdot P(MC|A,E,B) \cdot P(A|E,B) \cdot P(E|B) \cdot P(B)$$

$$= P(JC|A) \cdot P(MC|A) \cdot P(A|B,E) \cdot P(E) \cdot P(B)$$

Global semantics of Bayesian networks

Global semantics defines the full joint distribution as the product the local conditional distributions



$$p(X_1,\ldots,X_n)=\prod_{i=1}^{n}p(X_i\,|\,Parents(X_i))$$

$e.g.$

$$\begin{aligned}
&p(j\wedge m\wedge a\wedge\neg b\wedge\neg e)\\
&\quad=p(j\,|\,a)\,p(m\,|\,\mathrm{a})\,p(a\,|\,\neg b,\neg e)\,p(\neg b)\,p(\neg e)\\
&\quad=0.90\times0.70\times0.001\times0.999\times0.998\\
&\quad=0.00062
\end{aligned}$$

Thank you