Knowledge Representation and Inference

IT426: Artificial Intelligence

Information Technology Department

Intelligent Agent

- We learned how an agent decide , e.g. through search, classification, clustering…etc. (i.e. the control policy)
- But before deciding on an action, an intelligent agent needs **knowledge** about the real world
- An intelligent agent also needs **reasoning** to act efficiently.

In Other Words…

- An intelligent agent must able to do the following:
  - An agent should be able to represent states, actions, etc.
  - An agent should be able to incorporate new percepts
  - An agent can update the internal representation of the world
  - An agent can deduce the internal representation of the world
  - An agent can deduce appropriate actions.

Knowledge Representation And Reasoning

- Concerned with representing information about the real world so that a computer can understand and can utilize this knowledge to solve the complex real world problems
- Knowledge representation and reasoning enable agents to reach good decisions.
- How does it represent the knowledge to help it decide?

Knowledge Representation And Reasoning

1. **Knowledge-base**
    - It is a collection of sentences that are expressed in a language which is called a knowledge representation language.
    - The Knowledge-base stores fact about the world.
    - Knowledge-base is required for updating knowledge for an agent to learn with experiences and take action as per the knowledge.
2. **Inference system**.
    - Inference means deriving new sentences from old.
    - Inference system generates new facts so that an agent can update the KB.
    - An inference system works mainly in two rules which are given as:
        - **Forward chaining**
        - **Backward chaining**

Objectives Of Knowledge Representation And Reasoning

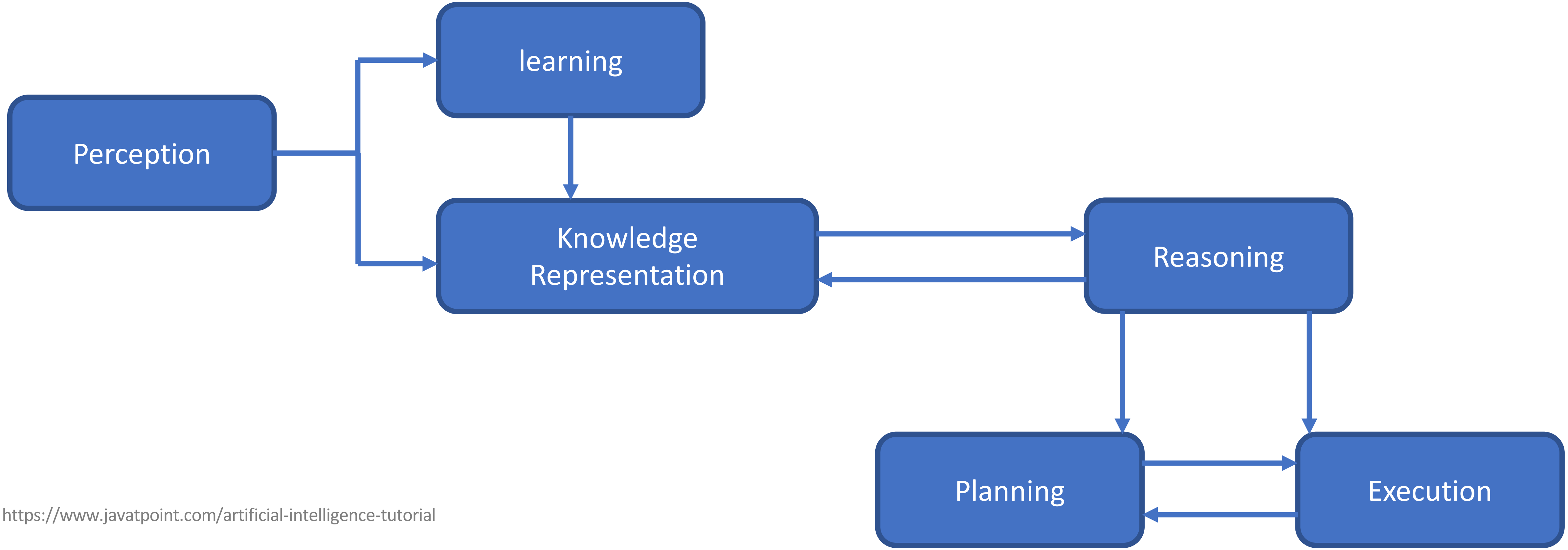
1. Form **representations** of the world.

2. Use a process of **inference** to derive **new representations** about the world.

3. Use these new representations to **deduce** what to do.

AI knowledge cycle

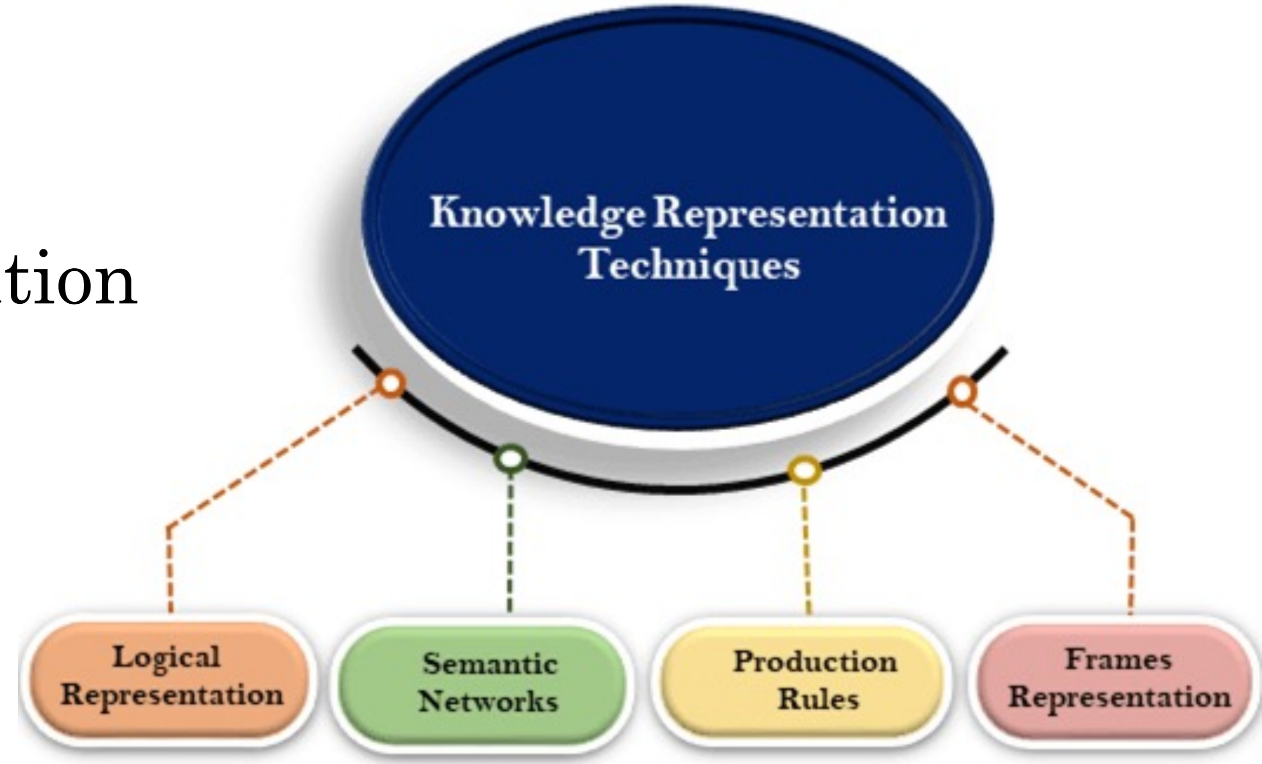
An Artificial intelligence system has the following components for displaying intelligent behavior:



https://www.javatpoint.com/artificial-intelligence-tutorial

Techniques of knowledge representation

- Logical Representation
- Semantic Network Representation
- Frame Representation
- Production Rules



https://www.javatpoint.com/artificial-intelligence-tutorial

Logical Representation

- Logical representation is based on rules
- It has syntax rules and semantic rules
- Uses these rules to translate sentences into logics

- Logical representation can be categorized into mainly two logics:
  - Propositional logic
  - First order logic
  - We will learn more about this topic later

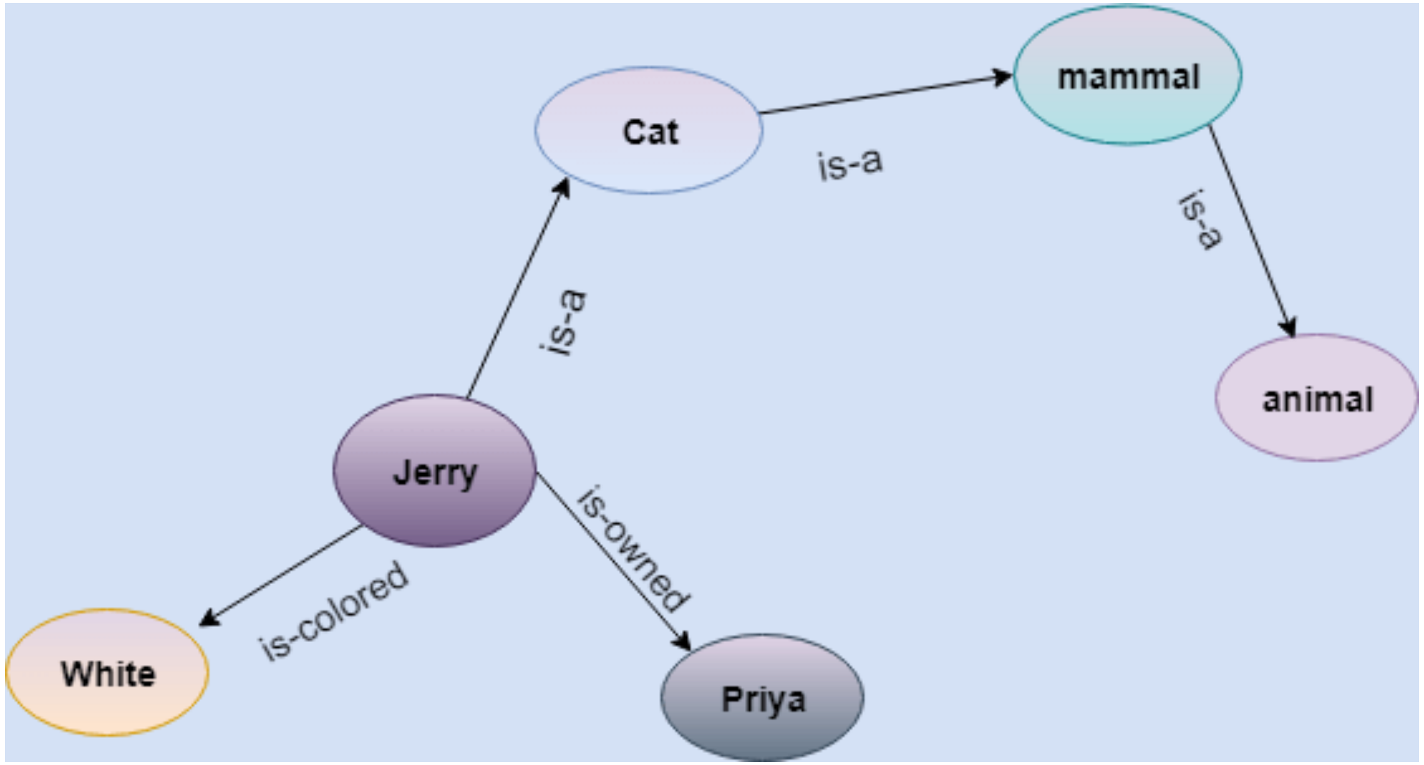
Semantic Network Representation

- Knowledge is represented using graphical networks
  - nodes represent objects
  - arcs represent relationship between those objects.
    - IS-A relation (Inheritance)
    - Kind-of-relation

Semantic Network Representation: An Example

KB has the following Statements:

- Jerry is a cat.
- Jerry is a mammal
- Jerry is owned by Priya.
- Jerry is brown colored.
- All Mammals are animal.



https://www.javatpoint.com/artificial-intelligence-tutorial

Frame Representation

- A frame is a data structure whose components are called slots.
- Slots have names and accommodate information of various kinds.
- E.g: frame for a book

Slots	Values
Title	Artificial Intelligence
Genre	Computer Science
Author	Peter Norvig
Edition	Third Edition
Year	1996
Page	1152

https://www.javatpoint.com/artificial-intelligence-tutorial

Production Rules

- Consist of (**condition, action**) pairs which mean, "If condition then action"

- E.g.
  - IF (at bus stop AND bus arrives) THEN action (get into the bus)
  - IF (on the bus AND paid AND empty seat) THEN action (sit down).
  - IF (on bus AND unpaid) THEN action (pay charges).
  - IF (bus arrives at destination) THEN action (get down from the bus).

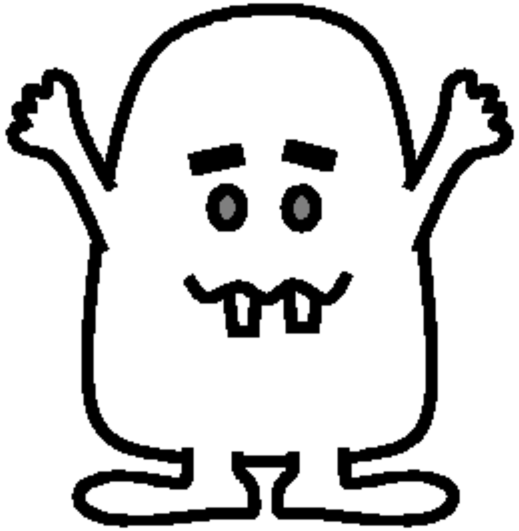
Logical Reasoning

- Reasoning is a general process of thinking rationally, to find valid conclusions
- In each case where the agent draws a conclusion from the available information, that conclusion is guaranteed to be *correct if the available information is correct*

To further understand the concept, first we are going to introduce an example…

Logical Reasoning

- Example: The Wumpus world

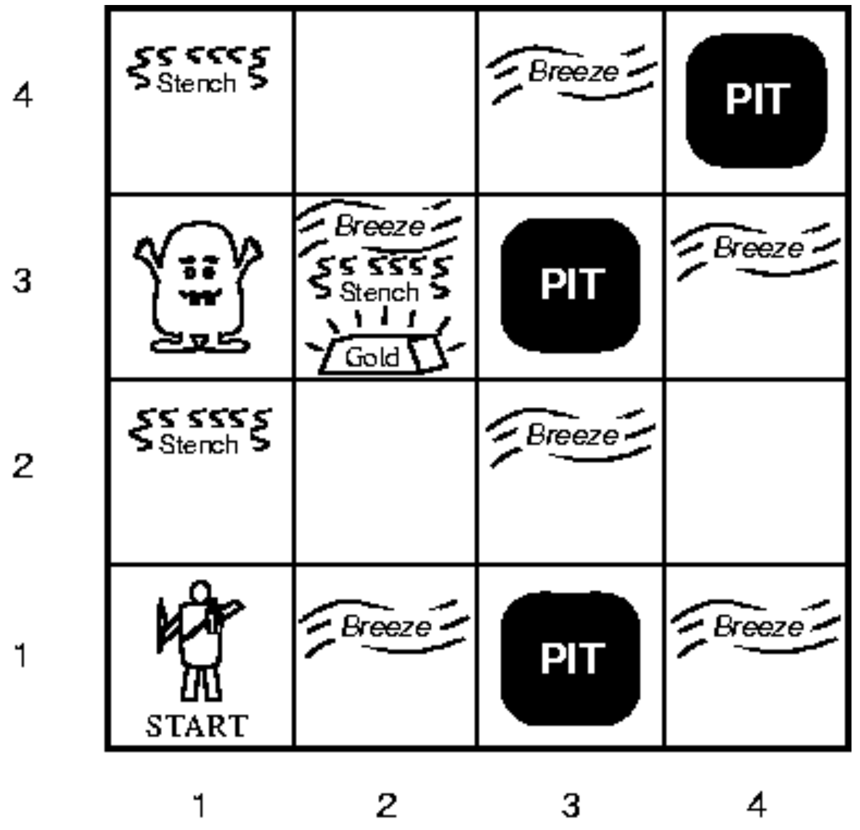


About The Wumpus

- The Wumpus world was first written as a computer game in the 70ies
- The wumpus world is a cave consisting of rooms connected by passageways
- The terrible wumpus is a beast that eats anyone who enters its room
- The wumpus can be shot by an agent, but the agent has only one arrow
- Some rooms contain bottomless pits that will trap anyone who enters it
- The agent is in this cave to look for gold

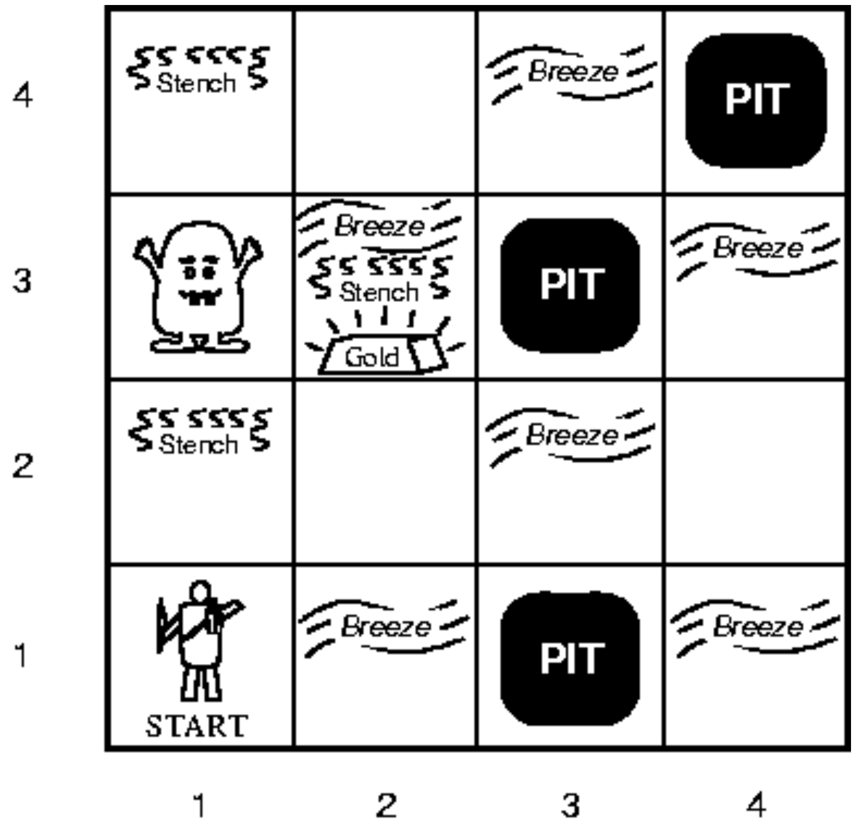
The Wumpus World Environment

- Squares adjacent to wumpus are smelly.
- Squares adjacent to pit are breezy.
- Glitter if and only if gold is in the same square.
- Shooting kills the wumpus if you are facing it.
- Shooting uses up the only arrow.
- Grabbing picks up the gold if in the same square.
- Releasing drops the gold in the same square.



The Wumpus World

- **Goals:** Get gold back to the start without entering wumpus square.
- **Percepts:** Breeze, Glitter, Smell.
- **Actions:** Left turn, Right turn, Forward, Grab, Release, Shoot.



The Wumpus World

- **Is the world deterministic?**

  Yes: outcomes are exactly specified.

- **Is the world fully accessible?**

  No: only local perception of square you are in.

- **Is the world static?**

  Yes: Wumpus and Pits do not move.

- **Is the world discrete?**

  Yes.

Exploring Wumpus World

A			

Exploring Wumpus World

Ok because:

Haven’t fallen into a pit.

Haven’t been eaten by a Wumpus.

OK A			

Exploring Wumpus World

OK since

no Stench,

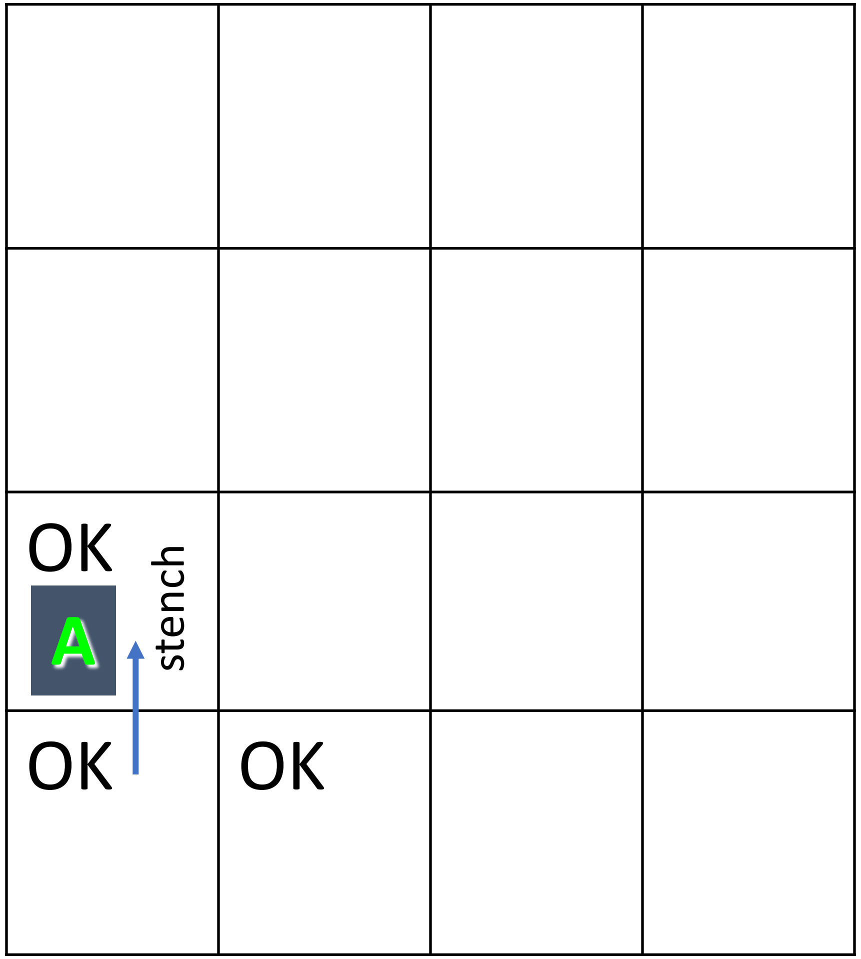
no Breeze,

neighbors are safe (OK).

OK			
OK A	OK		

Exploring Wumpus World

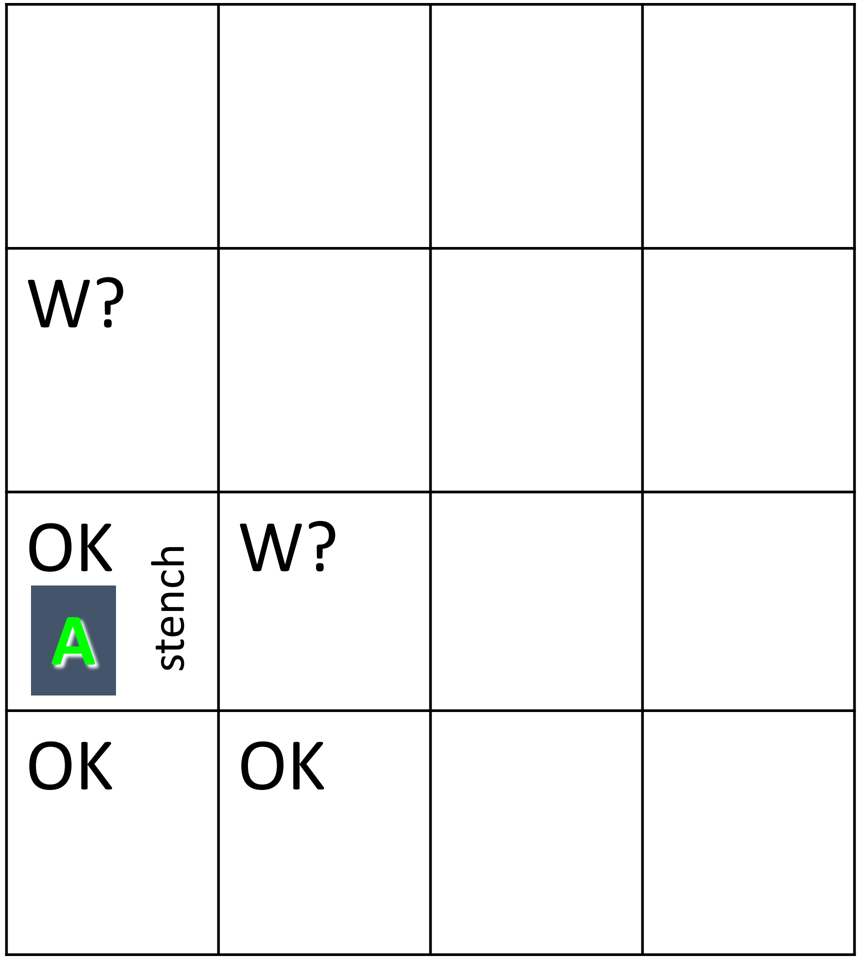
We move and smell a stench.



Exploring Wumpus World

We can infer the following.

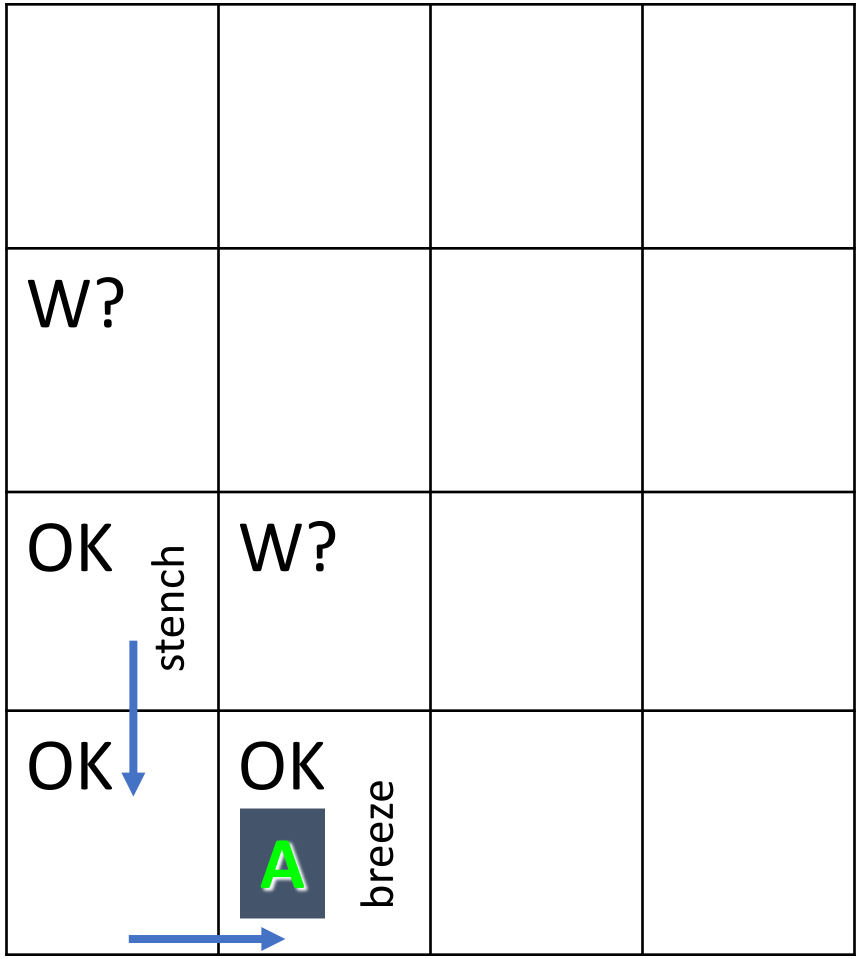
Note: square (1,1) remains OK.



Exploring Wumpus World

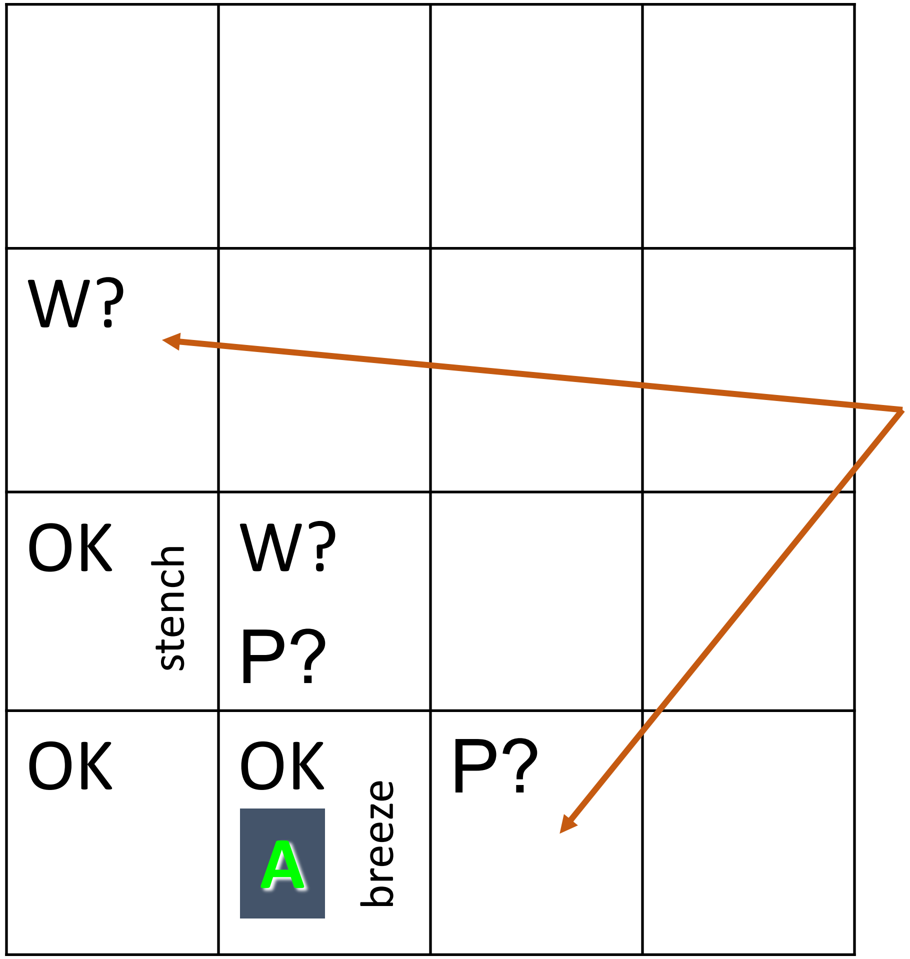
Move and feel a breeze

What can we conclude?



Exploring Wumpus World

But, can the 2,2 square really have either a Wumpus or a pit? **NO!**

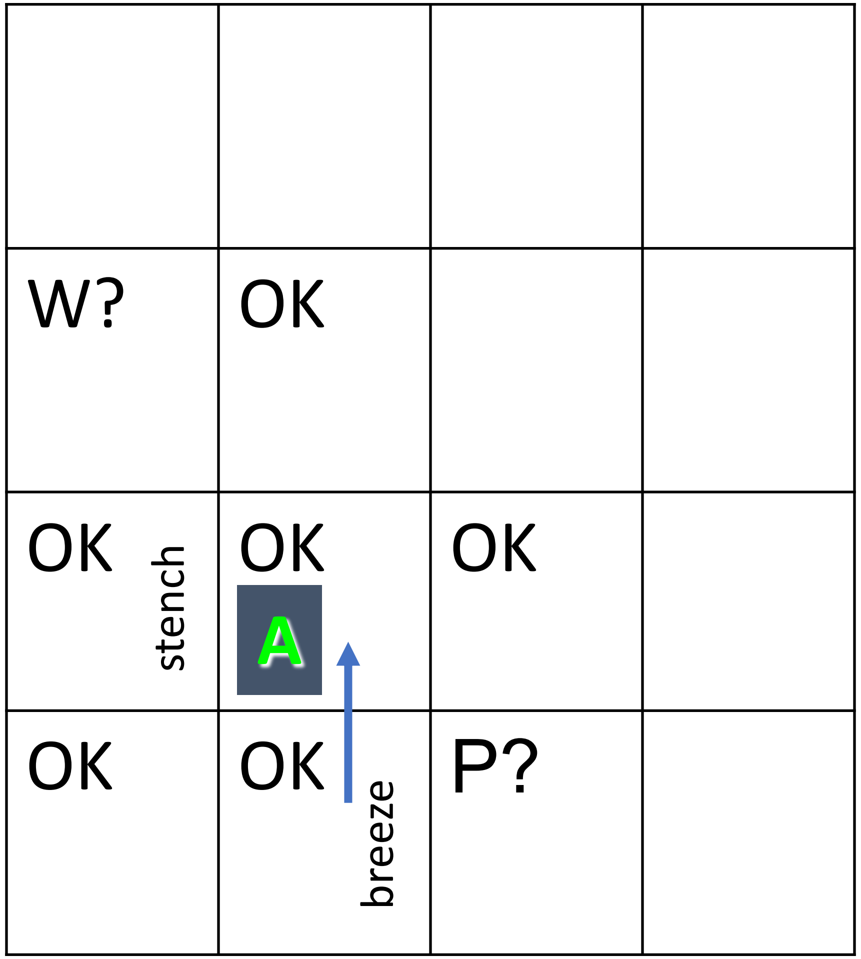


what about the other P? and W? squares

Exploring Wumpus World

W?			
OK stench	~~W? P?~~		
OK	OK A breeze	P?	

Exploring Wumpus World

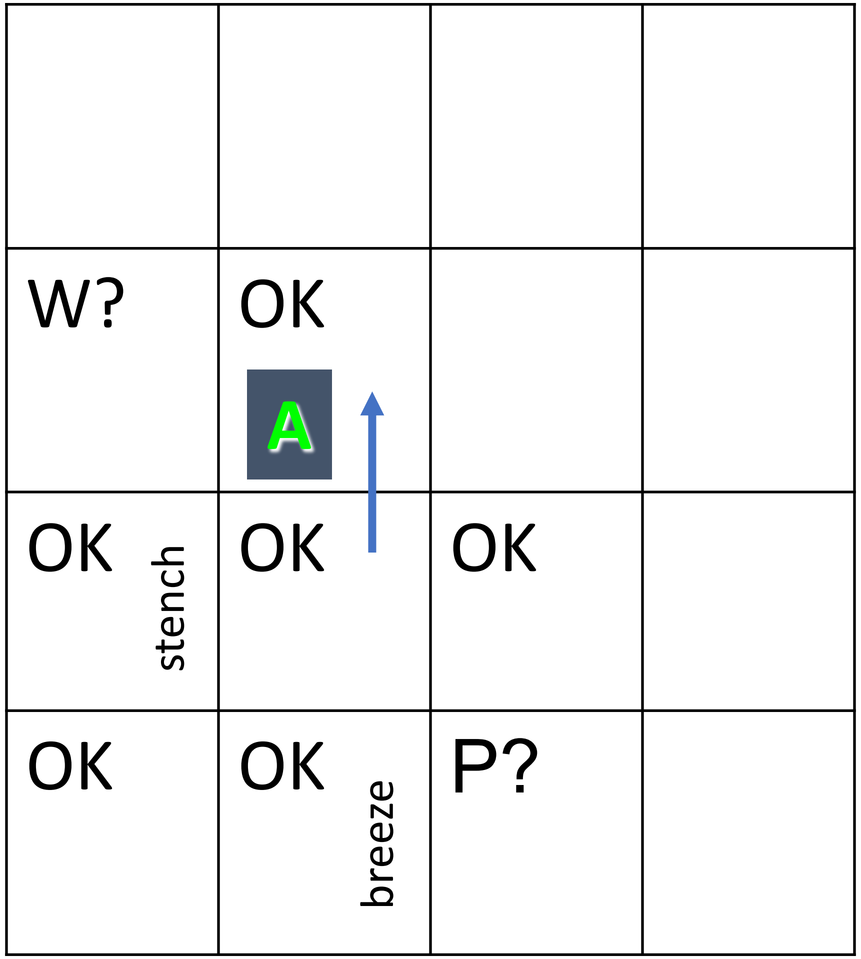


Exploring Wumpus World

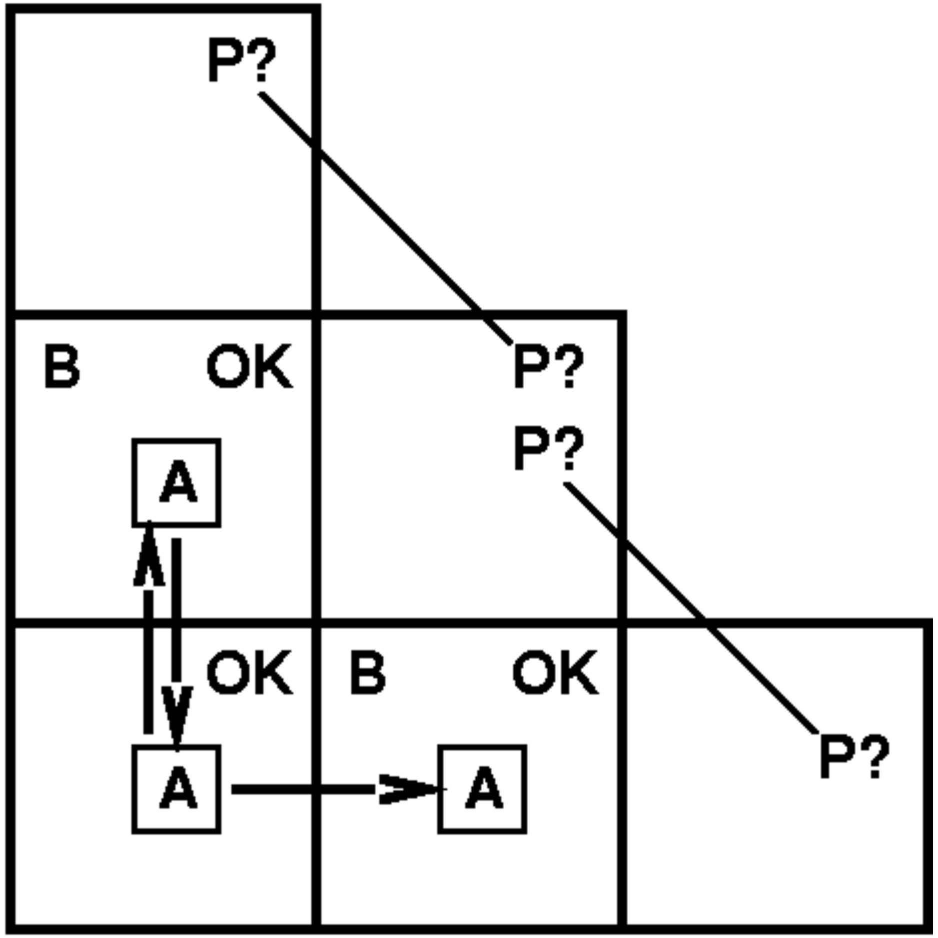
…

And the exploration continues onward until the gold is found.

…



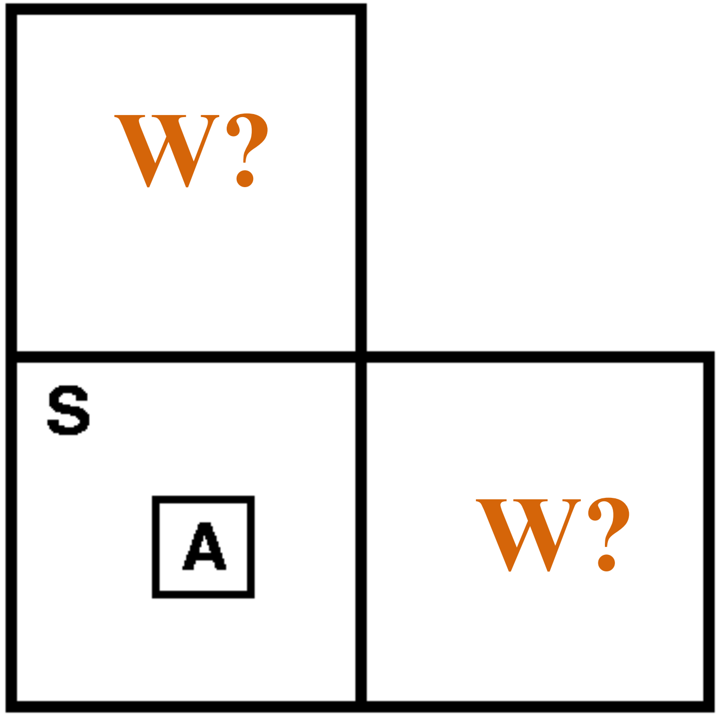
A Tight Spot



Breeze in (1,2) and (2,1)
→ no safe actions.

Assuming pits uniformly distributed, (2,2) is most likely to have a pit.

Another Tight Spot



Smell in (1,1)

→ cannot move.

Can use a strategy of coercion:

– shoot straight ahead;

– wumpus was there

→ dead → safe.

– wumpus wasn't there → safe.

Logical Representation

- **Logics** are formal languages for representing information such that conclusions can be drawn.
- Each sentence is defined by a **syntax** and a **semantic**.
- **Syntax** defines the sentences in the language. It specifies well formed sentences.
- **Semantics** define the ``meaning'' of sentences;
- i.e., in logic it defines the **truth of a sentence** in a possible world.

<u>For example, in the language of arithmetic:</u>

$x + 2 \geq y$ is a sentence.

$x + y >$ is not a sentence.

$x + 2 \geq y$ is true if the number x+2 is no less than the number y.

$x + 2 \geq y$ is true in a world where x = 7, y =1.

$x + 2 \geq y$ is false in a world where x = 0, y= 6.

Logical Representation

- **Model**: This word is used instead of “possible world” for the sake of precision.

  m is a model of a sentence a, means a is true in model m

  M(a) *means the set of all models of a*

  all the models where a is true

- **Definition:** A model is a mathematical abstraction that simply fixes the truth or falsehood of **every relevant sentence.**
- **Example:** x is the number of men and y is the number of women sitting at a table playing bridge.

  constraint: no negatives

  $x + y = 4$, is a sentence which is true when the total number is four.

In this case the **Model** is a **possible assignment** of numbers to the variables x and y. Each assignment fixes the truth of any sentence whose variables are x and y.

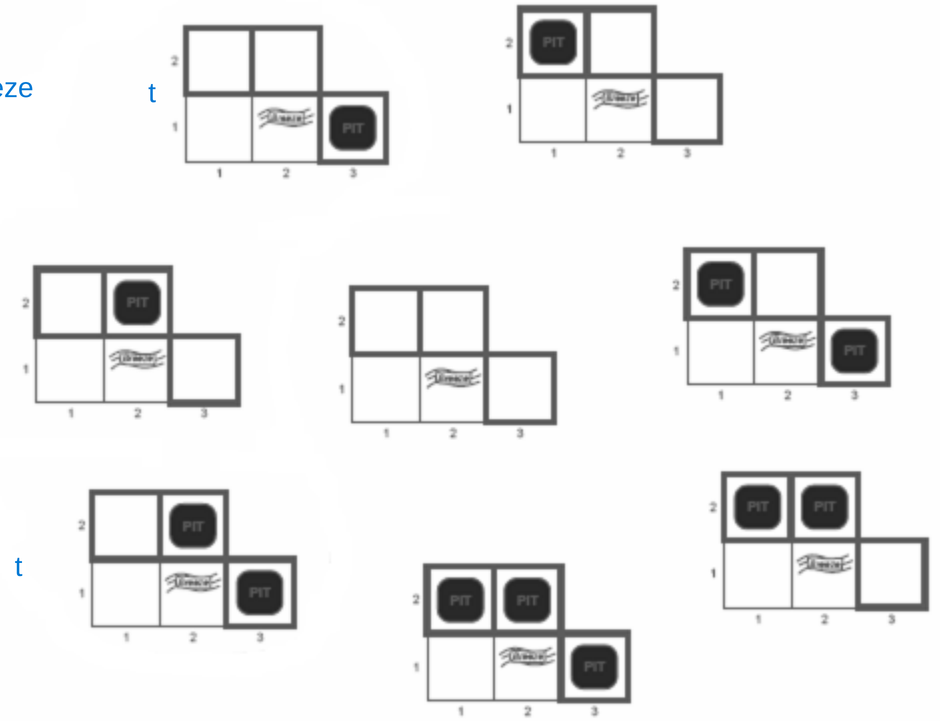
Potential Models Of The Wumpus World

at least one breeze

t

t

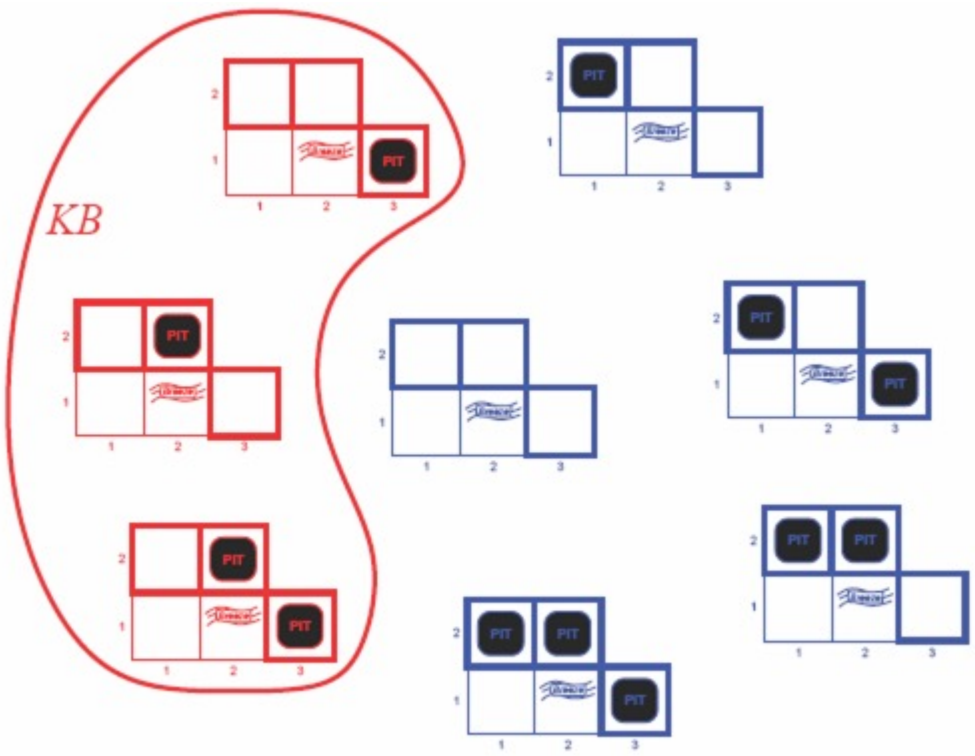
t



Following the wumpus-world rules, in which models the KB corresponding to the observations of nothing in [1,1] and breeze in [2,1] is true?

col , row

Potential Models Of The Wumpus World



KB = wumpus-world rules + observations

Logical Representation

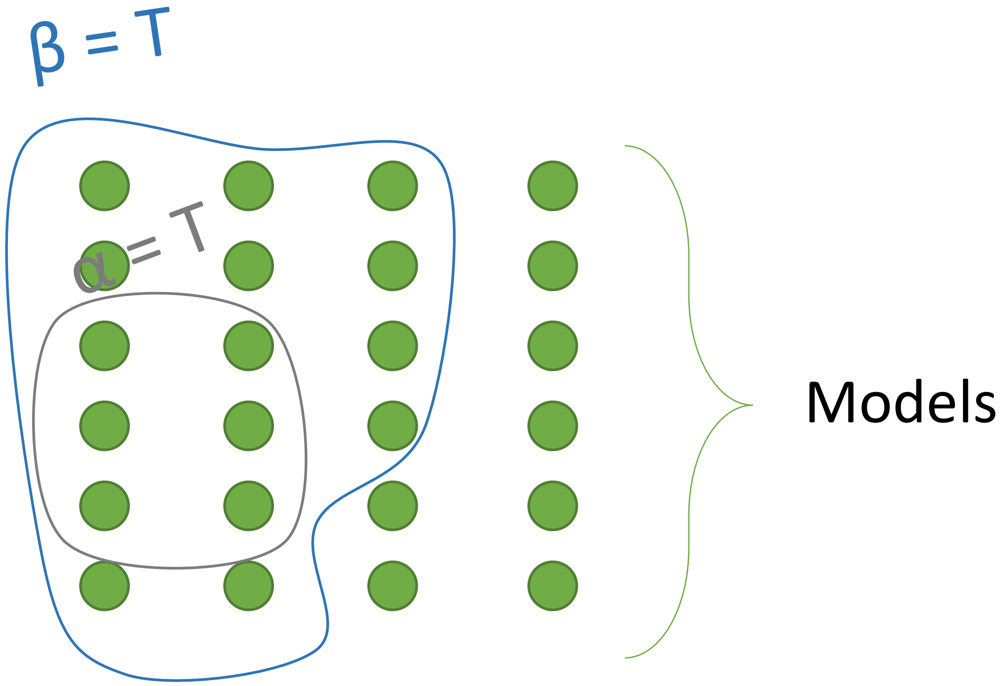
- **Entailment**: Logical reasoning requires the relation of logical entailment between sentences. ⇒ « a sentence follows logically from another sentence ».

  Mathematical notation: $\alpha \models \beta$ (α entails the sentence β)

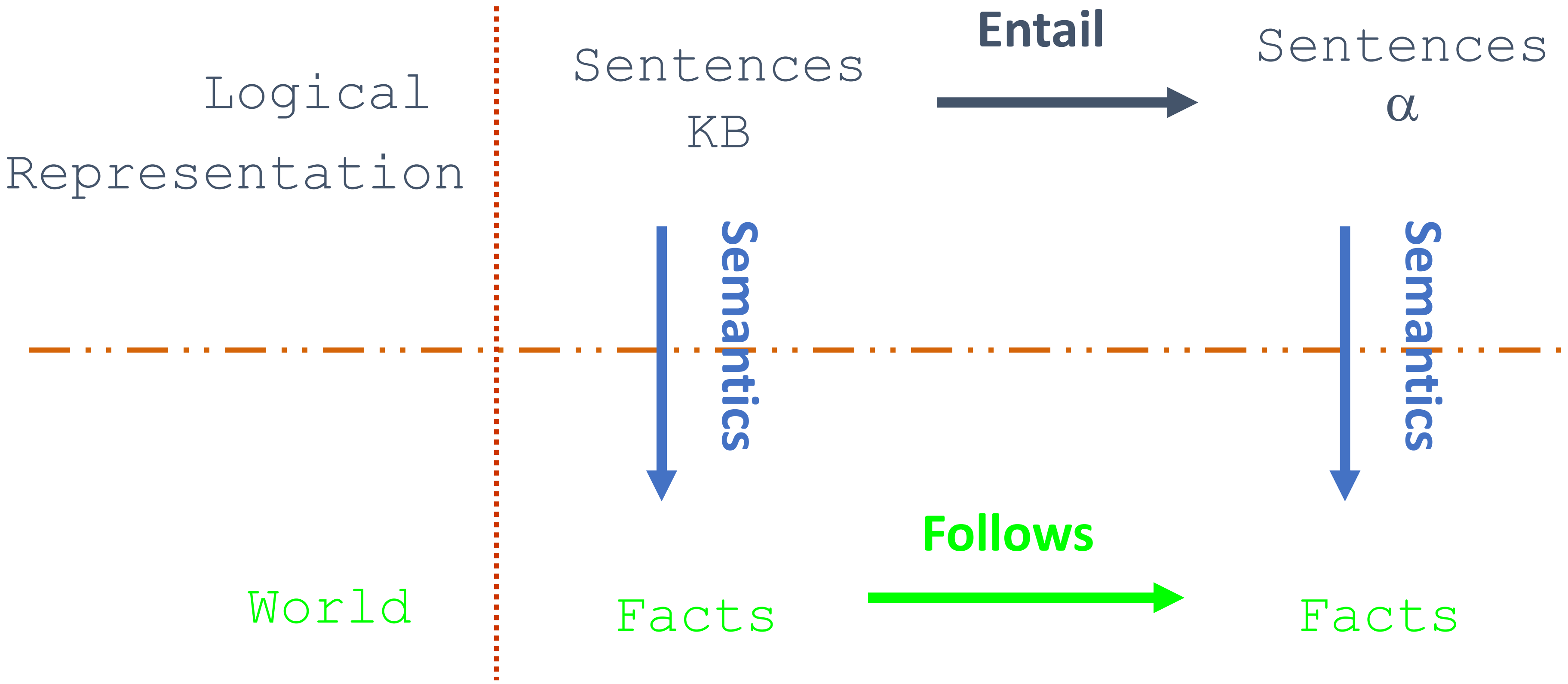
- Formal definition: $\alpha \models \beta$ if and only if in every model in which α is true, β is also true. (The truth of α is contained in the truth of β).

Entailment

- $\alpha \models \beta$ if and only if **in every model in which α is true, β is also true**. (The truth of α is contained in the truth of β).



Entailment



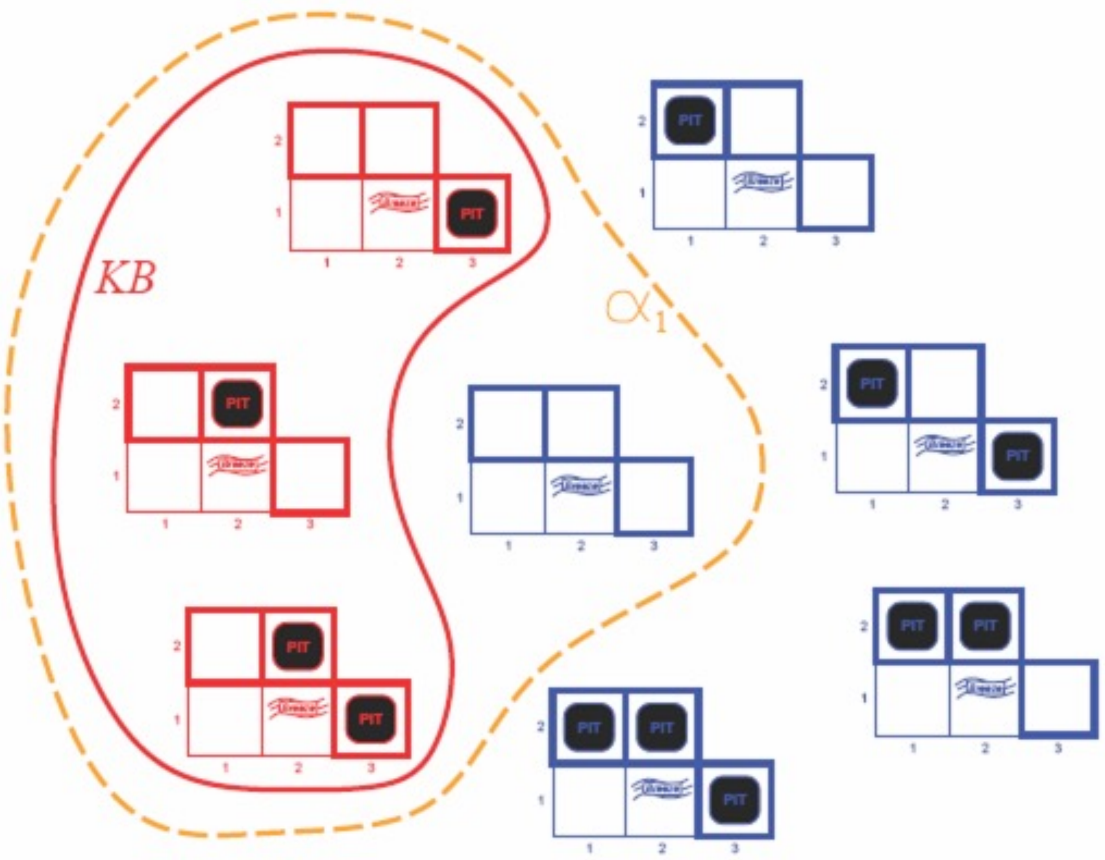
Logical reasoning should ensure that the **new configurations** represent aspects of the world that actually **follow from** the aspects that the **old configurations** represent.

Entailment

- **KB:** breeze in [2,1]
- $\boldsymbol{\alpha_1}$ **:** [1,2] is safe

- In every model in which KB is True, α1 is also True, therefore:

$$KB \models \alpha_1$$

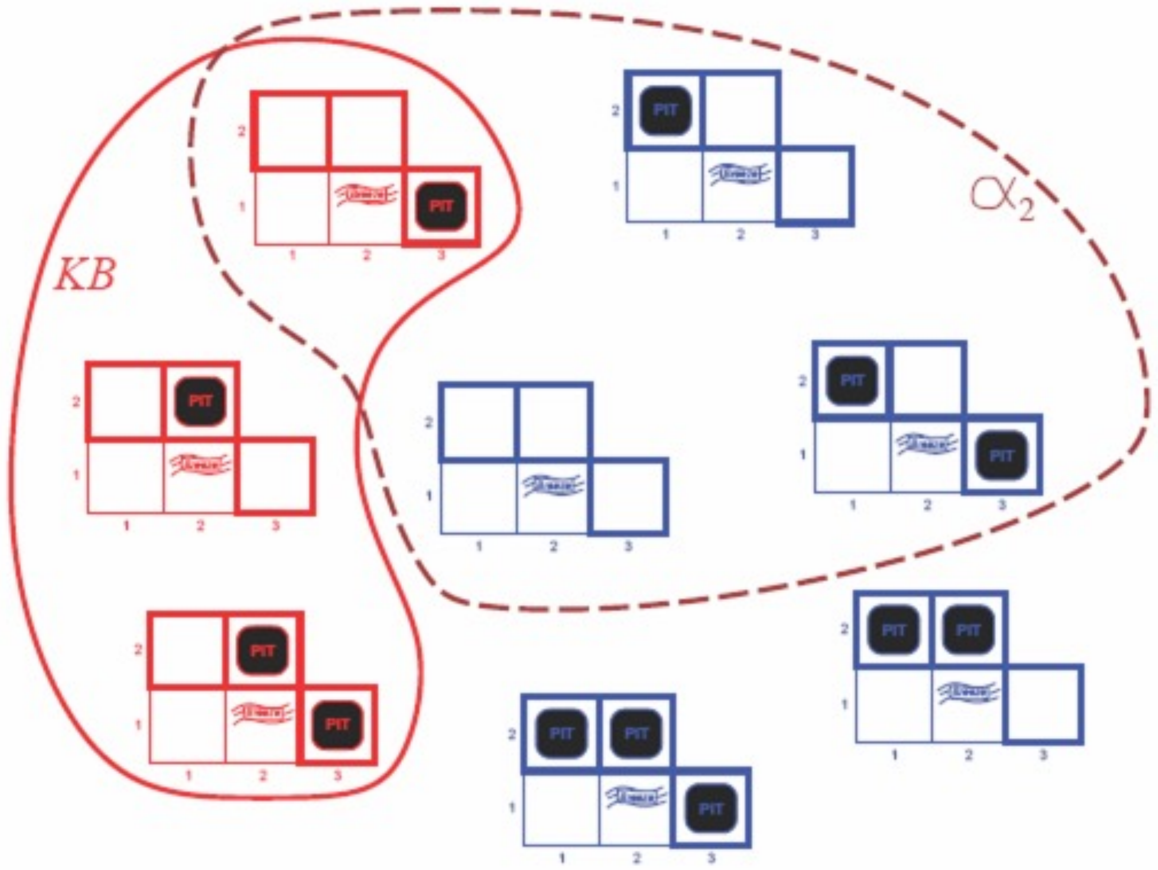


Entailment

- **KB:** breeze in [2,1]
- $\boldsymbol{\alpha_2}$ **:** [2,2] is safe

- There are models in which KB is True, and α_2 is *not* True, therefore:

$$KB \not\models \alpha_2$$



Model Checking

- The previous examples **not** only illustrate **entailment**, but also shows how **entailment** can be applied to ***derive conclusions*** or to carry out ***logical inference***
- The inference algorithm we used, is called **Model Checking**
- <u>**Model checking:**</u> Enumerates **all possible models** to check that α is true in **all models** in which KB is true.

Mathematical notation: $KB \vdash_i \alpha$

Where: ***i*** is the **inference algorithm** used.

The notation says:

- **α is derived from KB by *i***
- **or *i* derives α from KB.**

Entailment And Inference

- All consequence of KB is a **haystack**
- **α is a needle**

- Entailment:

  The needle in the haystack

- Inference:

  Finding the needle

Knowledge Representation & Reasoning

- An **inference procedure** can do two things:
  - Given KB, **generate new sentence α** purported to be **entailed by KB.**
  - Given KB and α, report **whether or not** α is entailed by KB.
- **Sound or truth preserving:** inference algorithm that derives **only entailed sentences.**
  - *i* is sound if: Whenever $KB \vdash_i \alpha$, it is also true that $KB \models \alpha$
- **Completeness**: an inference algorithm is complete, if it can derive **any sentence that is entailed**.
  - *i* is complete if: Whenever $KB \models \alpha$, it is also true that $KB \vdash_i \alpha$

Knowledge Representation & Reasoning

- **Soundness**: if the system proves that something is true, then it really is true. The system **doesn't derive contradictions**

- **Completeness:** if something is really true, **it can be proven using the system**. The system can be used to derive all the true mathematical statements one by one

Propositional Logic

- Propositional logic is the ***simplest logic***.
- Statements are made by propositions; declarative statements which are either true or false

- For example:
  - It is Sunday.
  - The Sun rises from West (False proposition)
  - 3+3= 7(False proposition)
  - 5 is a prime number.

Propositional Logic

Syntax: It defines the allowable sentences.

1. Atomic sentence:
   - **Single proposition symbol.**
     - uppercase names for symbols must have some mnemonic value: example $W_{1,3}$ to say the Wumpus is in [1,3].
   - **True and False:** proposition symbols with fixed meaning.
2. Complex sentences:
   - they are **constructed** from simpler sentences using logical connectives.
     - E.g. "It is raining today, and street is wet."

Propositional Logic

Logical connectives:

- $\neg$ (NOT) negation.
- $\wedge$ (AND) conjunction, operands are conjuncts.
- $\vee$ (OR), operands are disjuncts.
- $\Rightarrow$ (Implication or conditional) It is also known as rule or if-then statement.
- $\Leftrightarrow$ if and only if (biconditional).

Propositional Logic

- Logical **constants** TRUE and FALSE are sentences.
- **Proposition symbols** P1, P2 etc. are sentences.
- Symbols **P1** and negated symbols $\neg$ **P1** are called **literals**.
- If S is a sentence, $\neg$ S is a sentence (NOT).
- If S1 and S2 is a sentence, S1 $\wedge$ S2 is a sentence (AND).
- If S1 and S2 is a sentence, S1 $\vee$ S2 is a sentence (OR).
- If S1 and S2 is a sentence, S1 $\Rightarrow$ S2 is a sentence (Implies).
- If S1 and S2 is a sentence, S1 $\Leftrightarrow$ S2 is a sentence (Equivalent).

Propositional Logic

A BNF(Backus-Naur Form) grammar of sentences in propositional Logic is defined by the following rules:

$$\begin{aligned}
Sentence &\rightarrow AtomicSentence \mid ComplexSentence \\
AtomicSentence &\rightarrow \mathrm{True} \mid \mathrm{False} \mid Symbol \\
Symbol &\rightarrow \mathrm{P} \mid \mathrm{Q} \mid \mathrm{R} \ldots \\
ComplexSentence &\rightarrow \neg\, Sentence \\
&\mid (Sentence \Rightarrow Sentence) \\
&\mid (Sentence \Leftrightarrow Sentence) \\
&\mid (Sentence \land Sentence) \\
&\mid (Sentence \lor Sentence)
\end{aligned}$$

Propositional Logic

Order of precedence

From highest to lowest:

parenthesis (Sentence)

- NOT $\neg$
- AND $\wedge$
- OR $\vee$
- Implies $\Rightarrow$
- Biconditional $\Leftrightarrow$

- Special cases: $A \vee B \vee C$ no parentheses are needed
- What about $A \Rightarrow B \Rightarrow C$???

Propositional Logic

- Semantic: it defines the **rules** for determining the truth of a sentence with respect to a particular model.

How to compute the **truth value** of any sentence given a **model**?

Propositional Logic

Sentences

4 Models: P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \Rightarrow Q$	$P \Leftrightarrow Q$
False	False	True	False	False	True	True
False	True	True	False	True	True	False
True	False	False	False	True	False	False
True	True	False	True	True	True	True

$P \vee Q$ is True in 3 models:
P = F & Q =T
P = T & Q = F
P = T & Q = T

✓Most sentences are **sometimes true.**

$$P \wedge Q$$

✓ Some sentences are **always true** (valid).

$$\neg P \vee P$$

✓ Some sentences are **never true** (unsatisfiable).

$$\neg P \wedge P$$

Propositional Logic

P	Q		$P \Rightarrow Q$
False	False		True
False	True		True
True	False		False
True	True		True

Implication: $P \Rightarrow Q$

"If P is True, then Q is true; otherwise I'm making no claims about the truth of Q."
(Or: $P \Rightarrow Q$ is equivalent to $\neg P \vee Q$)

Under this definition, the following statement is true

Pigs_fly $\Rightarrow$ Everyone_gets_an_A

Since "Pigs_Fly" is false, the statement is true irrespective of the truth of "Everyone_gets_an_A". [Or is it? Correct inference only when "Pigs_Fly" is known to be false.]

Propositional Logic

Propositional Inference:

Using Enumeration Method (Model checking)

- Let $\boldsymbol{\alpha = A \vee B}$ and

KB =(A ∨ C) ∧(B ∨ ¬C)

- Is it the case that KB $\models$ α ?
- Check all possible models -- α must be true whenever KB is true.

A	B	C	KB (A ∨ C) ∧ (B ∨ ¬C)	α A ∨ B
False	False	False	False	False
False	False	True	False	False
False	True	False	False	True
False	True	True	True	True
True	False	False	True	True
True	False	True	False	True
True	True	False	True	True
True	True	True	True	True

Propositional Logic

A	B	C	**KB** **$(A \lor C) \land (B \lor \neg C)$**	**α** **$A \lor B$**
False	False	False	False	False
False	False	True	False	False
False	True	False	False	True
False	True	True	True	True
True	False	False	True	True
True	False	True	False	True
True	True	False	True	True
True	True	True	True	True

Propositional Logic

A	B	C	**KB** (A ∨ C) ∧ (B ∨ ¬C)	**α** A ∨ B
False	False	False	False	False
False	False	True	Fal	False
False	True	False	False	True
False	True	True	True	True
True	False	False	True	True
True	False	True	False	True
True	True	False	True	True
True	True	True	True	True

KB $\models$ α

Propositional Logic: Proof methods

1. **Model checking**
   - Truth table enumeration (sound and complete for propositional logic).
   - For n symbols, the number of modules is 2^n , the time complexity is $O(2^n)$.
   - Need a smarter way to do inference

2. **Application of inference rules**
   - Legitimate (sound) generation of new sentences from old.
   - Proof = a sequence of inference rule applications.

   Can use inference rules as operators in a standard search algorithm. How?

Validity and Satisfiability

- A sentence is **valid** (a tautology) if it is true in all models

  e.g., *True*, $A \vee \neg A$, $A \Rightarrow A$,

- **Validity is connected to inference** via the Deduction Theorem:

  $KB \models \alpha$ if and only if $(KB \Rightarrow \alpha)$ is valid

- A sentence is **satisfiable** if it is true in **some** model

  e.g., $A \vee B$

- A sentence is **unsatisfiable** if it is false in **all** models

  e.g., $A \wedge \neg A$

- **Satisfiability is connected to inference** via the following:

  $KB \models \alpha$ if and only if $(KB \wedge \neg\alpha)$ is ***unsatisfiable***

  (there is no model for which KB=true and α is false)

Propositional Logic: Inference rules

An inference rule is **sound** if the conclusion is true in all cases where the premises are true.

$$\frac{\alpha}{\beta} \qquad \begin{array}{l}\text{Premise}\\ \text{Conclusion}\end{array}$$

Propositional Logic: Inference rules

An inference rule: <u>Modus Ponens</u>

- From an implication and the premise of the implication, you can infer the conclusion.

$$\frac{\alpha \Rightarrow \beta, \quad \alpha}{\beta} \qquad \begin{array}{l}\text{Premise} \\ \text{Conclusion}\end{array}$$

Example:

"raining implies soggy courts", "raining"

Infer: "soggy courts"

Propositional Logic: Inference rules

An inference rule: <u>Modus Tollens</u>

- if a conditional statement ("if α then β") is accepted, and the consequent does not hold (*not* β), then the negation of the antecedent (*not* α) can be inferred.

$$\frac{\alpha \Rightarrow \beta, \quad \neg \beta}{\neg \alpha}$$

premise

Conclusion inferred

Example:

"raining implies soggy courts", "courts not soggy"

Infer: "not raining"

Propositional Logic: Inference rules

An inference rule: <u>AND elimination</u>

- From a conjunction, you can infer any of the conjuncts.

$$\frac{\alpha 1 \wedge \alpha 2 \wedge \ldots \alpha n}{\alpha i} \qquad \begin{array}{l} \text{Premise} \\ \text{Conclusion} \end{array}$$

Propositional Logic: Inference rules

other inference rules…

- **And-Introduction**

$$\frac{\alpha_1, \alpha_2, \ldots, \alpha_n}{\alpha_1 \wedge \alpha_2 \wedge \ldots \alpha_n}$$

Premise

Conclusion

- **Double Negation**

$$\frac{\neg\neg\alpha}{\alpha}$$

Premise

Conclusion

Propositional Logic: Equivalence rules

- Two sentences are logically **equivalent** iff they are **true in the same models:** $\alpha \equiv \beta$ iff $\alpha \models \beta$ and $\beta \models \alpha$.

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

Propositional Logic: CNF & DNF

Other approaches to inference use syntactic operations on sentences, often expressed in standardized forms

Conjunctive Normal Form (CNF—universal)

$\mathit{conjunction}$ of $\underbrace{\mathit{disjunctions} \text{ of } \mathit{literals}}_{\mathit{clauses}}$

E.g., $(A \vee \neg B) \wedge (B \vee \neg C \vee \neg D)$

Disjunctive Normal Form (DNF—universal)

$\mathit{disjunction}$ of $\underbrace{\mathit{conjunctions} \text{ of } \mathit{literals}}_{\mathit{terms}}$

E.g., $(A \wedge B) \vee (A \wedge \neg C) \vee (A \wedge \neg D) \vee (\neg B \wedge \neg C) \vee (\neg B \wedge \neg D)$

Horn Form (restricted)

conjunction of *Horn clauses* (clauses with ≤ 1 positive literal)

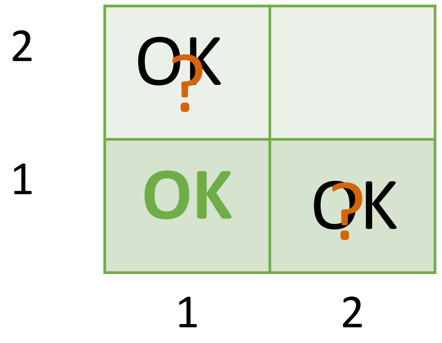
E.g., $(A \vee \neg B) \wedge (B \vee \neg C \vee \neg D)$

Often written as set of implications:

$B \Rightarrow A$ and $(C \wedge D) \Rightarrow B$

Inference In Wumpus World

- Let $S_{i,j}$ be true if there is a stench in cell i,j
- Let $B_{i,j}$ be true if there is a breeze in cell i,j
- Let $W_{i,j}$ be true if there is a Wumpus in cell i,j



Given:

1. ¬B1,1 → *observation*

2. B1,1 ⇔ (P1,2 ∨ P2,1) → *from Wumpus World rules*

Let's make some inferences:

1. **(B1,1 ⇒ (P1,2 ∨ P2,1)) ∧ ((P1,2 ∨ P2,1) ⇒ B1,1)** ***(By definition of the biconditional)***
2. **(P1,2 ∨ P2,1) ⇒ B1,1** ***(And-elimination)***
3. **¬B1,1 ⇒ ¬(P1,2 ∨ P2,1)** ***(equivalence with contrapositive)***
4. **¬(P1,2 ∨ P2,1)** ***(modus ponens)***
5. **¬P1,2 ∧ ¬P2,1** ***(DeMorgan's rule)***
6. **¬P1,2 , ¬ P2,1*****(And Elimination)***

Inference in Wumpus World

Initial KB

Percept Sentences

$\neg S_{1,1}$ $\neg B_{1,1}$

$S_{2,1}$ $\neg B_{1,2}$

$\neg S_{1,2}$ $B_{2,1}$

…

Environment Knowledge

R_1: $\neg S_{1,1} \Rightarrow \neg W_{1,1} \wedge \neg W_{2,1} \wedge \neg W_{1,2}$

R_2: $S_{2,1} \Rightarrow W_{1,1} \vee W_{2,1} \vee W_{2,2} \vee W_{3,1}$

R_3: $\neg B_{1,1} \Rightarrow \neg P_{1,1} \wedge \neg P_{2,1} \wedge \neg P_{1,2}$

R_5: $B_{1,2} \Rightarrow P_{1,1} \vee P_{1,2} \vee P_{2,2} \vee P_{1,3}$

...

Some inferences:

Apply **Modus Ponens** to R_1

Add to KB

$$\neg W_{1,1} \wedge \neg W_{2,1} \wedge \neg W_{1,2}$$

Apply to this **AND-Elimination**

Add to KB

$\neg W_{1,1}$

$\neg W_{2,1}$

$\neg W_{1,2}$

Inference In Wumpus World

$B_{1,2} \Leftrightarrow P_{1,1} \lor P_{2,2} \lor P_{1,3}$ *(From Wumpus world rules)*

$(B_{1,2} \Rightarrow P_{1,1} \lor P_{2,2} \lor P_{1,3}) \land (P_{1,1} \lor P_{2,2} \lor P_{1,3} \Rightarrow B_{1,2})$ *(Biconditional elimination)*

$(P_{1,1} \lor P_{2,2} \lor P_{1,3} \Rightarrow B_{1,2})$ *(And-Elimination)*

$\neg B_{1,2} \Rightarrow \neg (P_{1,1} \lor P_{2,2} \lor P_{1,3})$ *(Contraposition)*

$\neg B_{1,2} \Rightarrow \neg P_{1,1} \land \neg P_{2,2} \land \neg P_{1,3}$ *(De Morgan)*

Recall that when we were at (1,2) we could not decide on a safe move, so we backtracked, and explored (2,1), which yielded $\neg B_{1,2}$

Using $(\neg B_{1,2} \Rightarrow \neg P_{1,1} \land \neg P_{1,3} \land \neg P_{2,2})$ **&** $\neg B_{1,2}$

this yields to:

$\neg P_{1,1} \land \neg P_{1,3} \land \neg P_{2,2}$ *(Modes Ponens)*

$\neg P_{1,1}, \neg P_{1,3}, \neg P_{2,2}$ *(And-Elimination)*

3	W?		
2	S	P? W?	
1	OK	B	P?
	1	2	3

- **Now we can consider the implications of B2,1.**

Inference In Wumpus World

1. $B_{2,1} \Leftrightarrow (P_{1,1} \lor P_{2,2} \lor P_{3,1})$
2. $B_{2,1} \Rightarrow (P_{1,1} \lor P_{2,2} \lor P_{3,1}) \land (P_{1,1} \lor P_{2,2} \lor P_{3,1}) \Rightarrow B_{2,1}$ ***(biconditional Elimination)***
3. $B_{2,1} \Rightarrow (P_{1,1} \lor P_{2,2} \lor P_{3,1})$ ***(And-Elimination)*** **+** $B_{2,1}$
4. $P_{1,1} \lor P_{2,2} \lor P_{3,1}$ ***(modus ponens)***
5. $P_{1,1} \lor P_{3,1}$ (**resolution rule** because no pit in (2,2) $\neg P_{2,2}$)
6. $P_{3,1}$ (**resolution rule** because no pit in (1,1) $\neg P_{1,1}$)

- **<u>The resolution rule:</u>** if there is a pit in (1,1) or (3,1), and it's not in (1,1), then it's in (3,1).

$$\frac{P_{1,1} \lor P_{3,1}, \neg P_{1,1}}{P_{3,1}}$$

Resolution

Unit Resolution inference rule:

$$\frac{l_1 \vee \ldots \vee l_k, \; m}{l_1 \vee \ldots \vee l_{i-1} \vee l_{i+1} \vee \ldots \vee l_k}$$

where l_i and m are complementary literals.

Resolution

Full resolution inference rule:

$$\frac{l_1 \vee \ldots \vee l_k, \qquad m_1 \vee \ldots \vee m_n}{l_1 \vee \ldots \vee l_{i-1} \vee l_{i+1} \vee \ldots \vee l_k \vee m_1 \vee \ldots \vee m_{j-1} \vee m_{j+1} \vee \ldots \vee m_n}$$

where l_i and m_j are complementary literals.

Resolution

For simplicity let's consider clauses of length two:

$$\frac{l_1 \vee l_2, \neg l_2 \vee l_3}{l_1 \vee l_3}$$

To derive the **soundness of resolution** consider the values l_2 can take:

- If $\boldsymbol{l_2}$ **is *True***, then since we know that $\neg l_2 \vee l_3$ holds, it must be the case that $\boldsymbol{l_3}$ **is True**.
- If $\boldsymbol{l_2}$ **is *False***, then since we know that $l_1 \vee l_2$ holds, it must be the case that $\boldsymbol{l_1}$ **is True**.

Resolution

1. Properties of the resolution rule:
    - Sound
    - Complete (yields to a complete inference algorithm).

2. The resolution rule forms **the basis** for a family of complete inference algorithms.

3. Resolution rule is used to either confirm or refute a sentence but it **cannot** be used to enumerate true sentences.

Resolution

4. Resolution can be applied only to **disjunctions** of literals. How can it lead to a **complete inference procedure** for all propositional logic?

5. Turns out any knowledge base can be expressed as a **conjunction of disjunctions** (conjunctive normal form, CNF).

   E.g., $(A \vee \neg B) \wedge (B \vee \neg C \vee \neg D)$

Resolution: Inference Procedure

6. Inference procedures based on resolution work by using the principle of **proof by contradiction:**

To show that KB $\models \alpha$ we show that (KB $\land \neg\alpha$) is unsatisfiable

The process: 1. convert KB $\land \neg\alpha$ to CNF

2. resolution rule is applied to the resulting clauses.

Resolution: Inference Procedure

King Saud University

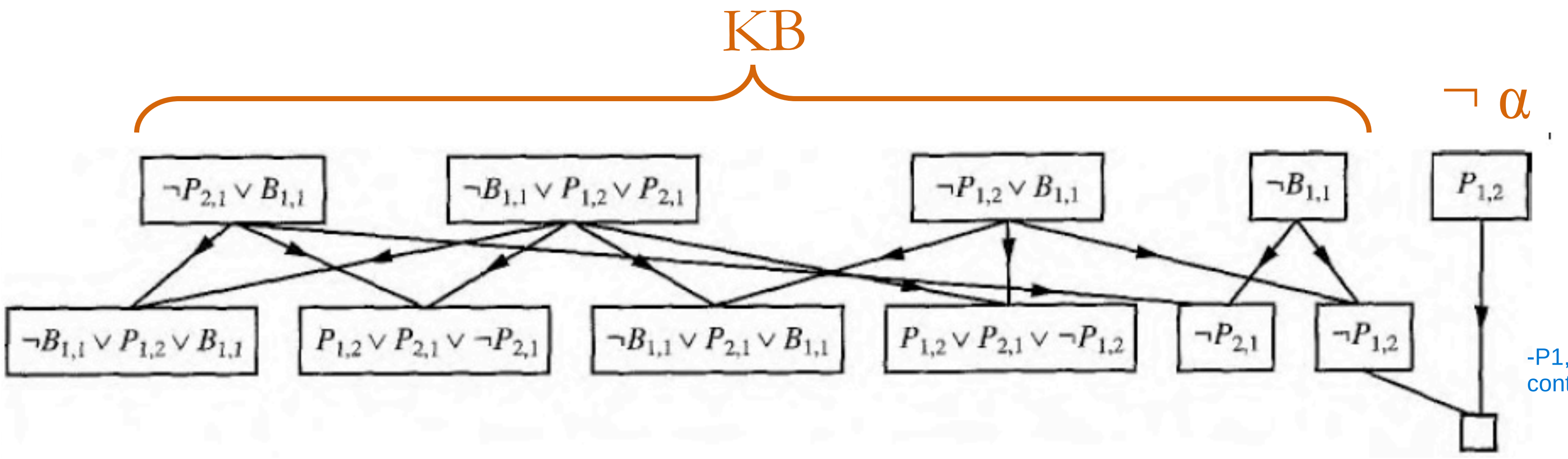
function PL-RESOLUTION(KB, α) **returns** *true* or *false*
  inputs: **KB,** the knowledge base, a sentence in propositional logic
    α, the query, a sentence in propositional logic

  clauses ← the set of clauses in the CNF representation of KB $\wedge \neg\alpha$
  new ← { }
  loop do
    for each C_i, C_j **in** *clauses* **do**
      resolvents ← PL-RESOLVE(C_i, C_j)
      if *resolvents* contains the empty clause **then return** *true*
      new ← *new* U *resolvents*
    if *new* $\subseteq$ *clauses* **then return** *false*
    clauses ← *clauses* U *new*

Resolution: Inference Procedure

Example of proof by contradiction:

- $KB = (B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})) \wedge \neg B_{1,1}$
- $\alpha = \neg P_{1,2}$



-P1,2 andn P1,2 are true contracditcion

Empty Clause

Question: convert $(KB \wedge \neg\alpha)$ to CNF

Resolution: Inference Procedure

Answer:

$(B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})) \wedge\neg B_{1,1} \wedge\neg\neg P_{1,2}$

$(\mathbf{B1,1} \Rightarrow (\mathbf{P1,2} \vee \mathbf{P2,1})) \wedge ((\mathbf{P1,2} \vee \mathbf{P2,1}) \Rightarrow \mathbf{B1,1}) \wedge\neg B_{1,1} \wedge\neg\neg P_{1,2}$ ***(biconditional)***

$(\neg\mathbf{B1,1} \vee (\mathbf{P1,2} \vee \mathbf{P2,1})) \wedge (\neg(\mathbf{P1,2} \vee \mathbf{P2,1}) \vee \mathbf{B1,1}) \wedge\neg B_{1,1} \wedge\neg\neg P_{1,2}$ ***(contrapositive)***

$(\neg\mathbf{B1,1} \vee \mathbf{P1,2} \vee \mathbf{P2,1}) \wedge ((\neg\mathbf{P1,2} \wedge \neg\mathbf{P2,1}) \vee \mathbf{B1,1}) \wedge\neg B_{1,1} \wedge\neg\neg P_{1,2}$ ***(de-morgan)***

$(\neg\mathbf{B1,1} \vee \mathbf{P1,2} \vee \mathbf{P2,1}) \wedge (\neg\boldsymbol{P1,2} \vee \boldsymbol{B1,1}) \wedge (\neg\boldsymbol{P2,1} \vee \boldsymbol{B1,1}) \wedge\neg B_{1,1} \wedge\neg\neg P_{1,2}$ ***(distributivity)***

$(\neg\mathbf{B1,1} \vee \mathbf{P1,2} \vee \mathbf{P2,1}) \wedge (\neg\boldsymbol{P1,2} \vee \boldsymbol{B1,1}) \wedge (\neg\boldsymbol{P2,1} \vee \boldsymbol{B1,1}) \wedge\neg B_{1,1} \wedge P_{1,2}$

Inference For Horn Clauses

. **Definite clause:** a disjunction of literals of which exactly one is positive.

Example: ($\neg L_{1,1}$ v $\neg$Breeze v $B_{1,1}$) , so all definite clauses are Horn clauses

. **Goal clauses:** clauses with no positive litterals.

. **Horn Form (special form of CNF)**

KB = conjunction of Horn clauses

Horn clause = propositional symbol; or

(conjunction of symbols) $\Rightarrow$ symbol

e.g., $C \wedge (B \Rightarrow A) \wedge (C \wedge D \Rightarrow B)$

Modus Ponens is a natural way to make **inference in Horn KBs**

Inference For Horn Clauses

$$\frac{\alpha_1, \ldots, \alpha_n, \quad \alpha_1 \wedge \ldots \wedge \alpha_n \Rightarrow \beta}{\beta}$$

Successive application of modus ponens leads to algorithms that are **sound and complete**, and run in linear time.

Inference For Horn Clauses: Forward & Backward Chaining

- Forward chaining is a form of reasoning which start with atomic sentences in the knowledge base and applies inference rules (Modus Ponens) in the forward direction to extract more data until a goal is reached.

- Backward-chaining is a form of reasoning, which starts with the goal and works backward, chaining through rules to find known facts that support the goal.

Forward Chaining Example

- Assume the KB with the following rules and facts:
- KB:

R1: $A \land B \Rightarrow C$	F1: A
R2: $C \land D \Rightarrow E$	F2: B
R3: $C \land F \Rightarrow G$	F3: D

Theorem: E ?

- Idea: fire any rule whose premises are satisfied in the KB and add its conclusion to the KB, until query is found.

Forward Chaining Example

- Theorem: E
- KB:
- R1: $A \land B \Rightarrow C$
- R2: $C \land D \Rightarrow E$
- R3: $C \land F \Rightarrow G$
- F1: A
- F2: B
- F3: D

Forward Chaining Example

- Theorem: E
- KB:
- R1: $A \wedge B \Rightarrow C$
- R2: $C \wedge D \Rightarrow E$
- R3: $C \wedge F \Rightarrow G$
- F1: A
- F2: B
- F3: D

Rule R1 is satisfied

- F4: C

Forward Chaining Example

- Theorem: E
- KB:
- R1: $A \wedge B \Rightarrow C$
- R2: $C \wedge D \Rightarrow E$
- R3: $C \wedge F \Rightarrow G$
- F1: A
- F2: B
- F3: D
- F4: C

Rule R2 is satisfied

- F5: E (proof)

Backward Chaining Example

- Assume the KB with the following rules and facts:
- KB:

R1: $A \wedge B \Rightarrow C$	F1: A
R2: $C \wedge D \Rightarrow E$	F2: B
R3: $C \wedge F \Rightarrow G$	F3: D

Theorem: E ?

- Idea: work backwards from the query q: check if q is known already, or prove by backward chaining all premises of some rule concluding q.

Backward Chaining Example

- Theorem: E
- KB:
- R1: $A \wedge B \Rightarrow C$
- R2: $C \wedge D \Rightarrow E$
- R3: $C \wedge F \Rightarrow G$
- F1: A
- F2: B
- F3: D

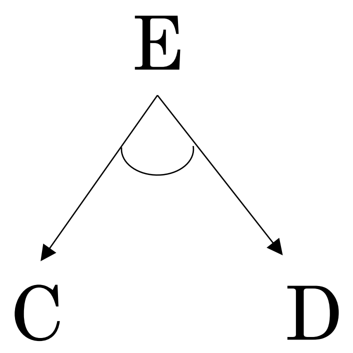
Backward Chaining Example

- Theorem: E
- KB:
- R1: $A \land B \Rightarrow C$
- R2: $C \land D \Rightarrow E$
- R3: $C \land F \Rightarrow G$
- F1: A
- F2: B
- F3: D

E

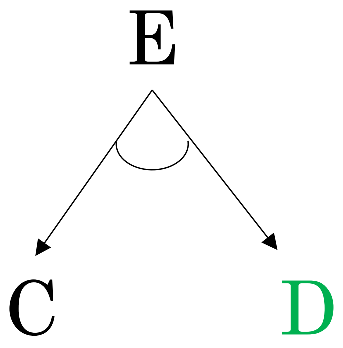
Backward Chaining Example

- Theorem: E
- KB:
- R1: $A \land B \Rightarrow C$
- R2: $C \land D \Rightarrow E$
- R3: $C \land F \Rightarrow G$
- F1: A
- F2: B
- F3: D



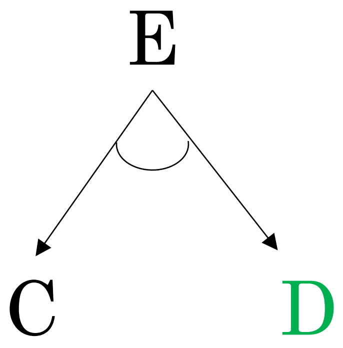
Backward Chaining Example

- Theorem: E
- KB:
- R1: $A \wedge B \Rightarrow C$
- R2: $C \wedge D \Rightarrow E$
- R3: $C \wedge F \Rightarrow G$
- F1: A
- F2: B
- F3: D



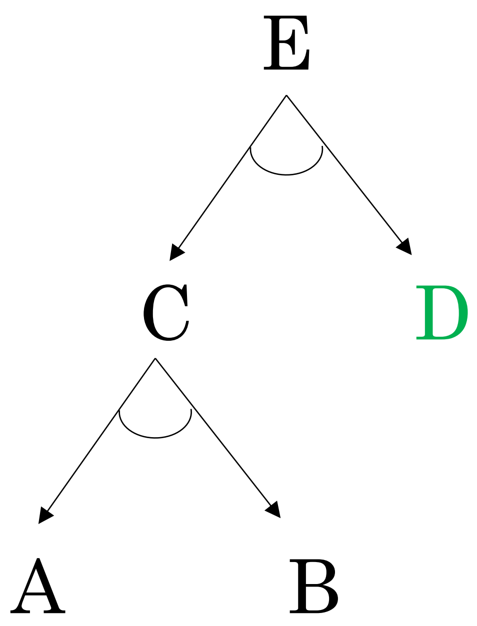
Backward Chaining Example

- Theorem: E
- KB:
- R1: A∧B⇒C
- R2: C∧D⇒E
- R3: C∧F⇒G
- F1: A
- F2: B
- F3: D



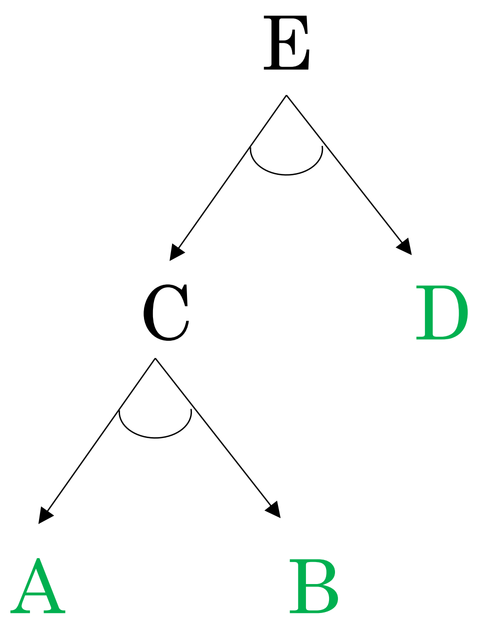
Backward Chaining Example

- Theorem: E
- KB:
- R1: $A \wedge B \Rightarrow C$
- R2: $C \wedge D \Rightarrow E$
- R3: $C \wedge F \Rightarrow G$
- F1: A
- F2: B
- F3: D



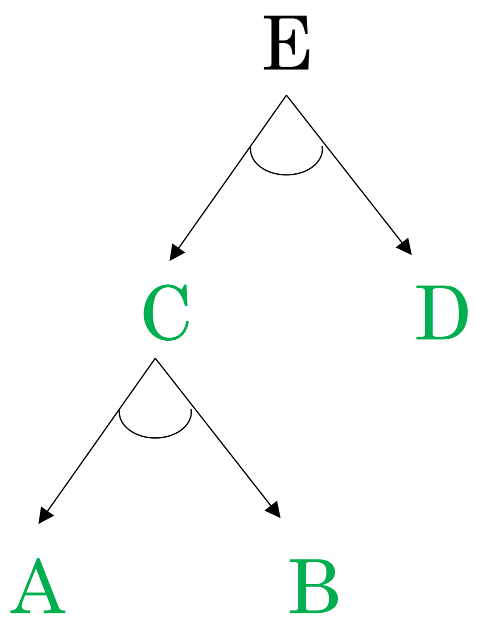
Backward Chaining Example

- Theorem: E
- KB:
- R1: $A \land B \Rightarrow C$
- R2: $C \land D \Rightarrow E$
- R3: $C \land F \Rightarrow G$
- F1: A
- F2: B
- F3: D



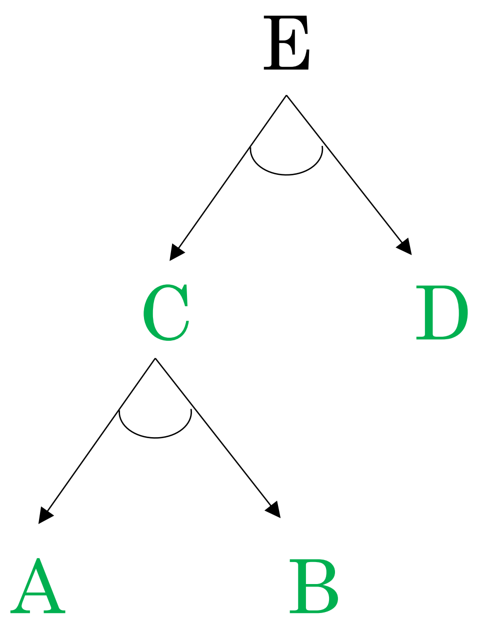
Backward Chaining Example

- Theorem: E
- KB:
- R1: A∧B⇒C
- R2: C∧D⇒E
- R3: C∧F⇒G
- F1: A
- F2: B
- F3: D
- F4: C



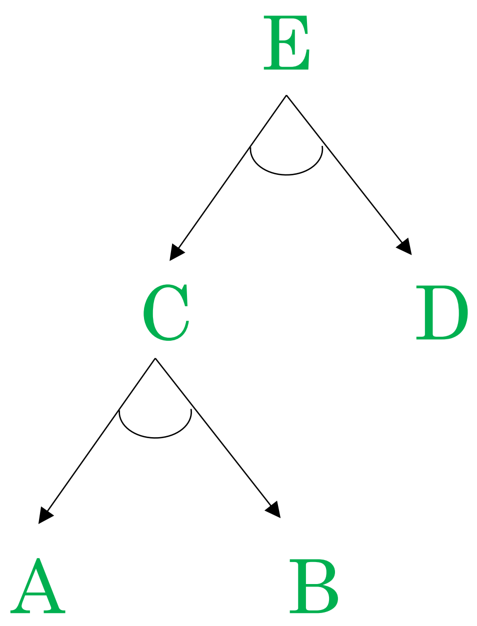
Backward Chaining Example

- Theorem: E
- KB:
- R1: $A \land B \Rightarrow C$
- R2: $C \land D \Rightarrow E$
- R3: $C \land F \Rightarrow G$
- F1: A
- F2: B
- F3: D
- F4: C



Backward Chaining Example

- Theorem: E
- KB:
- R1: $A \land B \Rightarrow C$
- R2: $C \land D \Rightarrow E$
- R3: $C \land F \Rightarrow G$
- F1: A
- F2: B
- F3: D
- F4: C
- F5: E (proof)



References

- https://www.javatpoint.com/artificial-intelligence-tutorial
- https://people.cs.pitt.edu/~milos/courses/cs2740/Lectures/class6.pdf