Artificial Intelligence

CSE 440/EEE 333/ETE333

Chapter 7
Summer 2016

Mirza Mohammad Lutfe Elahi
Department of Electrical and Computer Engineering
North South University

Knowledge Based Agents - Concept

- Knowledge Base or KB
  - a set of sentences describing the world
- Knowledge representation language
  - Expresses each sentence
- Inference
  - the process of deriving new sentences from the knowledge base
  - Tell(P) – function that adds knowledge P to the KB
  - Ask(P) – function that queries the agent about the truth of
- Have background knowledge about the world

Basic Actions – Ask/Tell

- A knowledge base keeps track of things
- We can TELL it facts or ASK for inference
- For example:
  - TELL: Father of John is Bob
  - TELL: Jane is John’s sister
  - TELL: John’s Father is the same as John’s sister’s father
  - ASK: Who’s Jane’s Father?

Knowledge Based Agents

- At every step:
  - Construct a sentence with assertion about percepts
  - Construct a sentence asking what action is next
  - Constructs a sentence asserting that action

Knowledge-Based Agents

function KB-AGENT(*percept*) **returns** an *action*
  persistent: *KB*, a knowledge base
          t, a counter, initially 0, indicating time

  TELL(*KB*, MAKE-PERCEPT-SENTENCE(*percept*, *t*))
  action $\leftarrow$ ASK(*KB*, MAKE-ACTION-QUERY(*t*))
  TELL(*KB*, MAKE-ACTION-SENTENCE(*action*, *t*))
  $t \leftarrow t + 1$
  return *action*

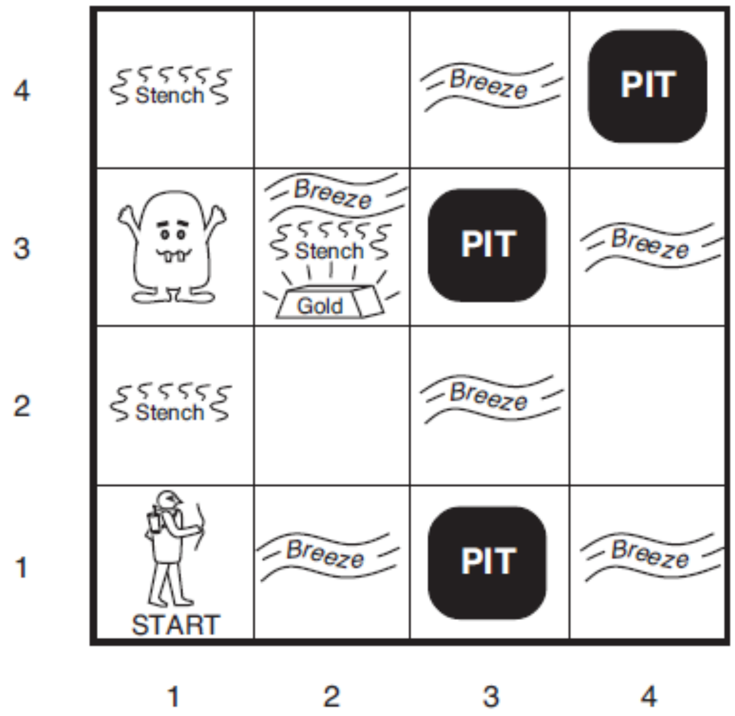
A generic knowledge-based agent.
Given a percept –

- the agent adds the percept to its knowledge base,
- asks the knowledge base for the best action, and
- tells the knowledge base that it has in fact taken that action.

The Wumpus World - A Dangerous Grid

- Adjacent rooms are connected (horizontally or vertically)
- Lurking in the cave is the Wumpus
- Player can smell the Wumpus (stench)
- Player feels a breeze if pit nearby
- Player can shoot ONE arrow at (and kill) the Wumpus
- Some rooms contain pits that will trap player
- One room contains a pot of gold (Yay!)

The Wumpus World



The Wumpus World

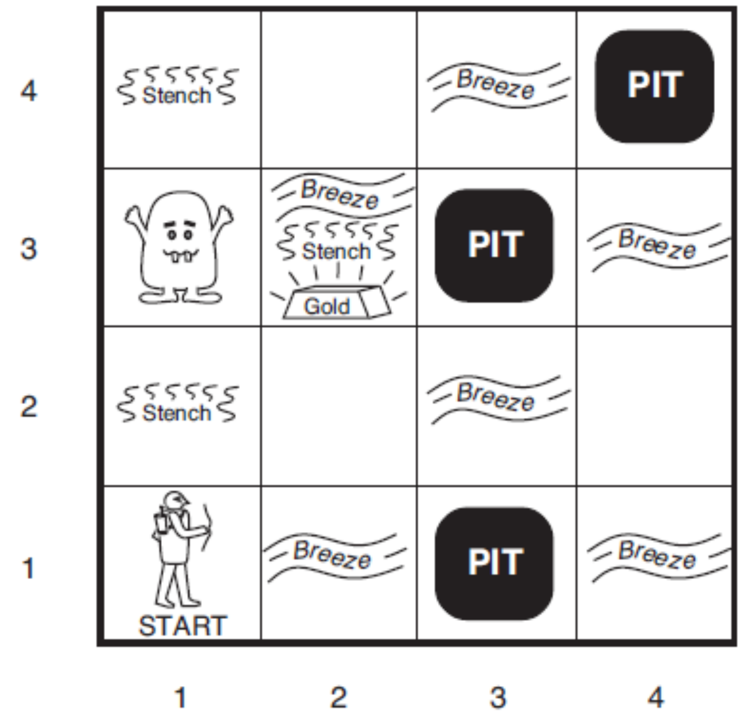
- PEAS:
  - **Performance measure**: +1000 for walk out w/gold; -1000 for dying; -1 for each action, -10 for arrow
  - **Environment** a 4 4 grid. Agent starts at [1,1]; gold and pits randomly distributed, etc..
  - **Actuators** Agent can move forward, left or right
  - **Sensors**: [*Smell*, *Breeze*, *Glitter*, *Bump*, *Scream*]

The Wumpus World

- Observable? No – only local perception
- Deterministic? Yes – outcomes exactly specified
- Episodic? No – sequential at the level of actions
- Static? Yes – Wumpus and Pits do not move
- Discrete? Yes – Discrete set of percepts and actions
- Single-agent? Yes – Wumpus is essentially a natural feature

The 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 World

First percept at [1,1]

[*None, None, None, None, None*]

Stench, Breeze, Glitter, Bump, Scream

Percept at [2,1]

[*None, Breeze, None, None, None*]

(a) (b)

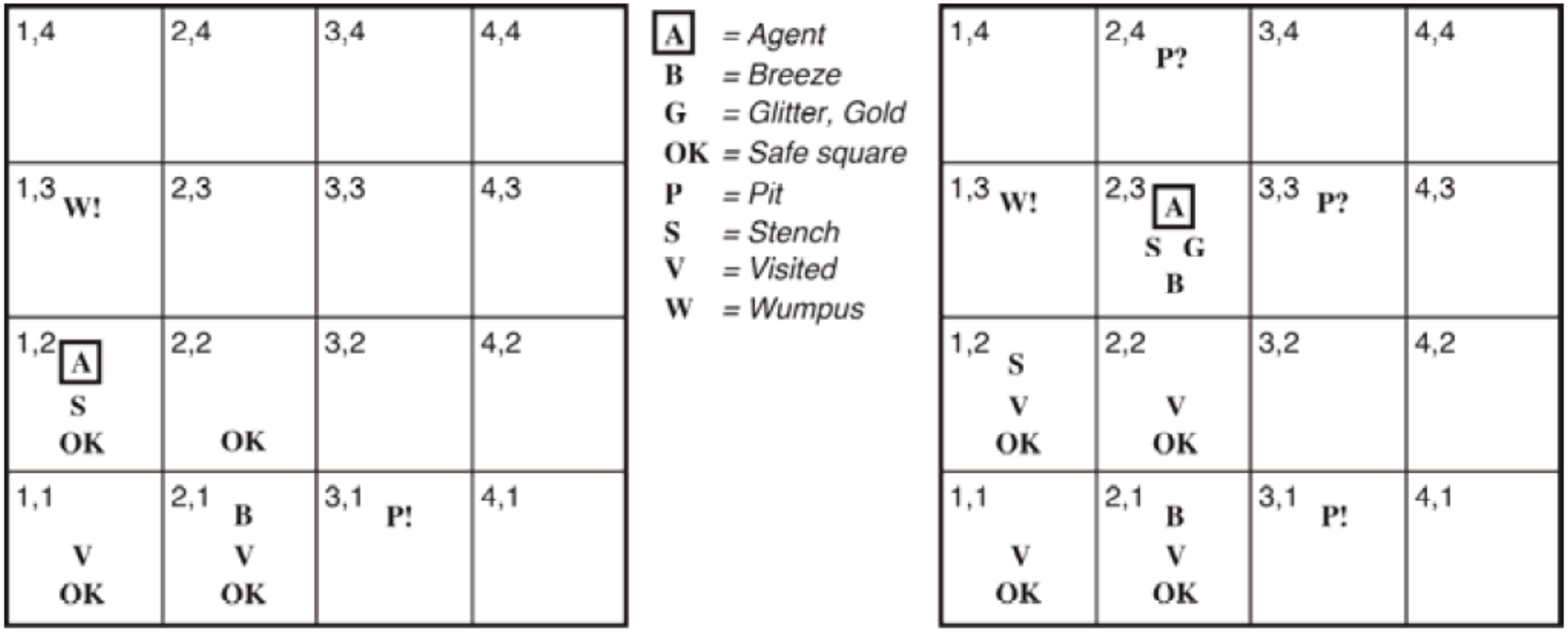
The Wumpus World

Percept at [1,2]

[*Stench, None, None, None, None*]

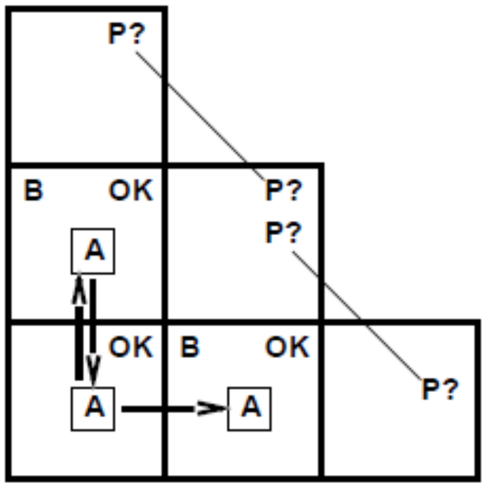
Percept at [2,3]

[*Stench, Breeze, Glitter, None, None*]



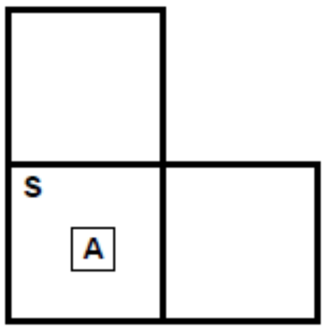
(a) (b)

The Wumpus World - Other tight spots



Breeze in (1,2) and (2,1)

→no safe actions



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

Logic - Basic

- A Logic has syntax
  - : e.g. “x + y = 4” is well formed; “x4y+” = is not
- Semantics define the **truth** of a sentence with respect to each possible world
  - For x + y = 4,
    - true in a world where x is 2 and y is 2
    - false in a world where x is 1 and y is 1
- Models describe possible worlds.
  - Mathematical abstraction
  - Possible models are just all possible assignments to variables
- If a sentence α is true in a model m, m satisfies α or m is a model of α
- $M(\alpha)$ is the set of all models of α

Logic - Entailment

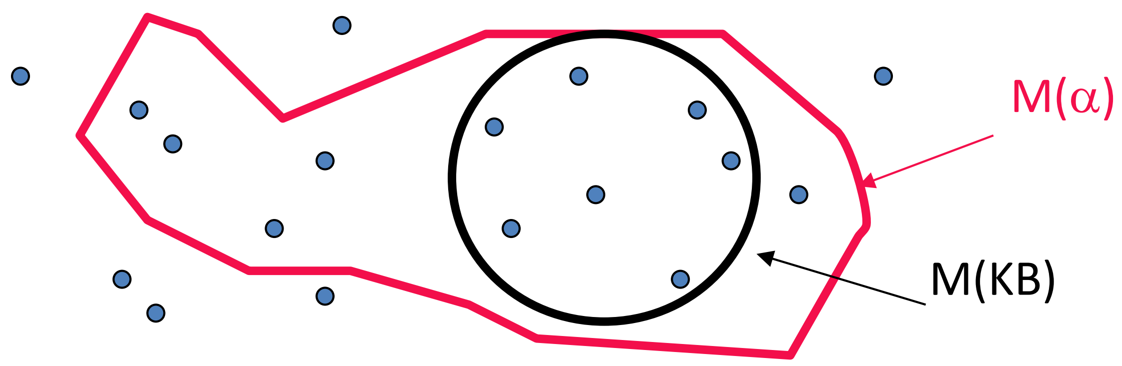
- Entailment is when a sentence logically follows from another $\alpha \models \beta$ - α entails β
- $\alpha \models \beta$ iff in every model where α is true, β is also true.
- $\alpha \models \beta$ iff $M(\alpha) \subseteq M(\beta)$.
- examples:
  - $(x = 0) \models (xy = 0)$
  - $(p = \text{true}) \models (p \vee q)$
  - $(p \wedge q) \models (p \vee q)$

Logic

- Model – the world being described by a KB
- Satisfaction – model *m* satisfies a sentence α, if α is true in *m*
- Entailment – the concept that a sentence follows from another sentence:
  - $\alpha \models \beta$ if α is true then β must also be true.
- Logical inference – the process of using entailment to derive conclusions
- Model checking – enumeration of all possible models to ensure that a sentence α is true in all models in which KB is true

Model

- Models are formal definitions of possible states of the world
- We say m is a model of a sentence α if α is true in m
- $M(\alpha)$ is the set of all models of α
- Then $KB \models \alpha$ if and only if $M(KB) \subseteq M(\alpha)$



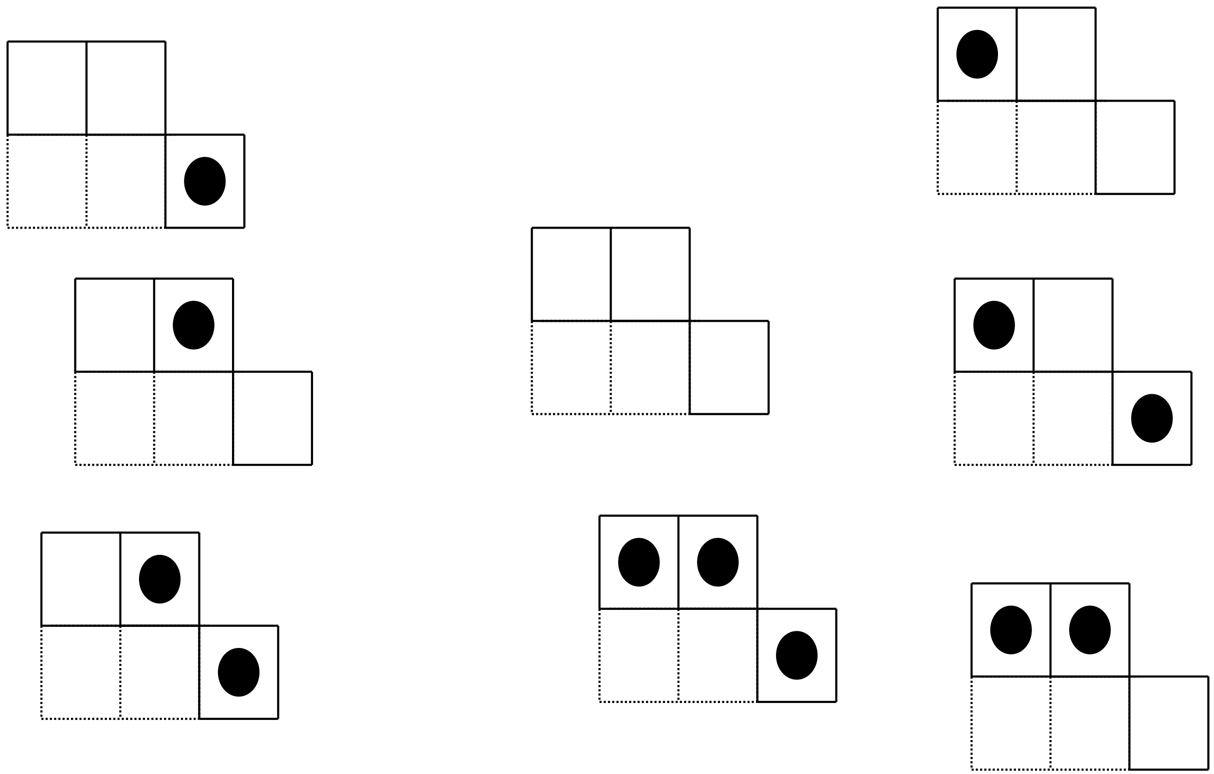
Entailment in the Wumpus World

- Situation after detecting nothing in [1,1], moving right, breeze in [2,1]
- What are possible models for ? – assume only possibility pit or no pit.

?	?		
V	B V	?	

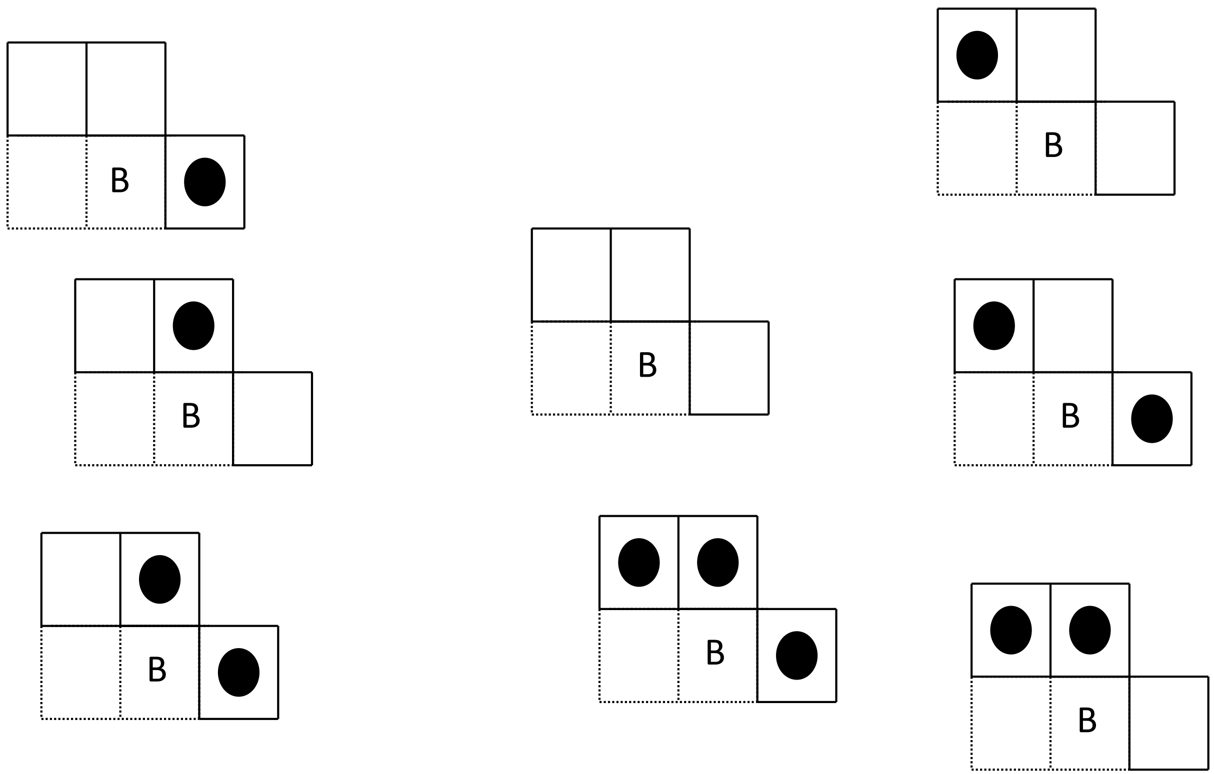
Wumpus Models

Model the presence of pits in squares [1,2][2,2] and [3,1].

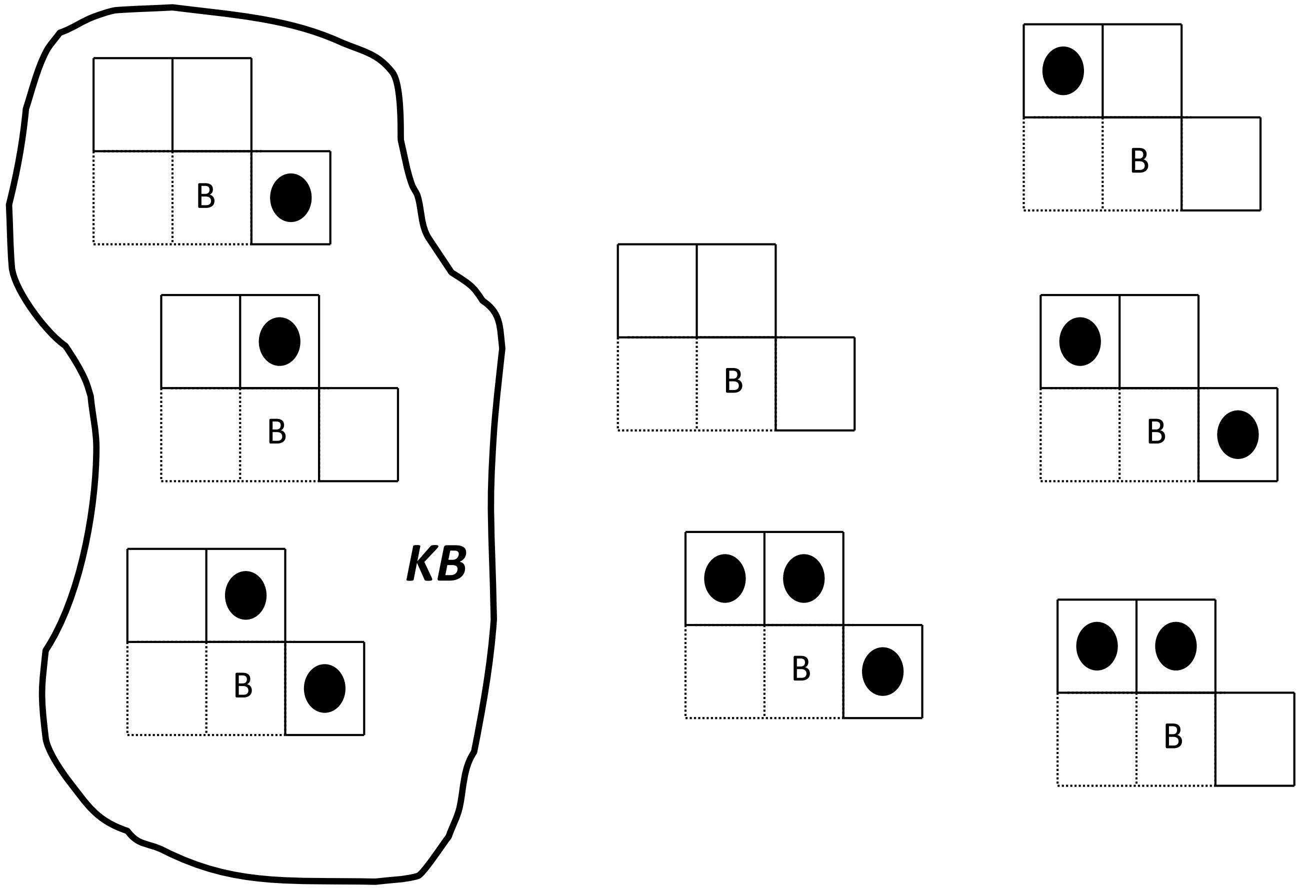


Wumpus Models

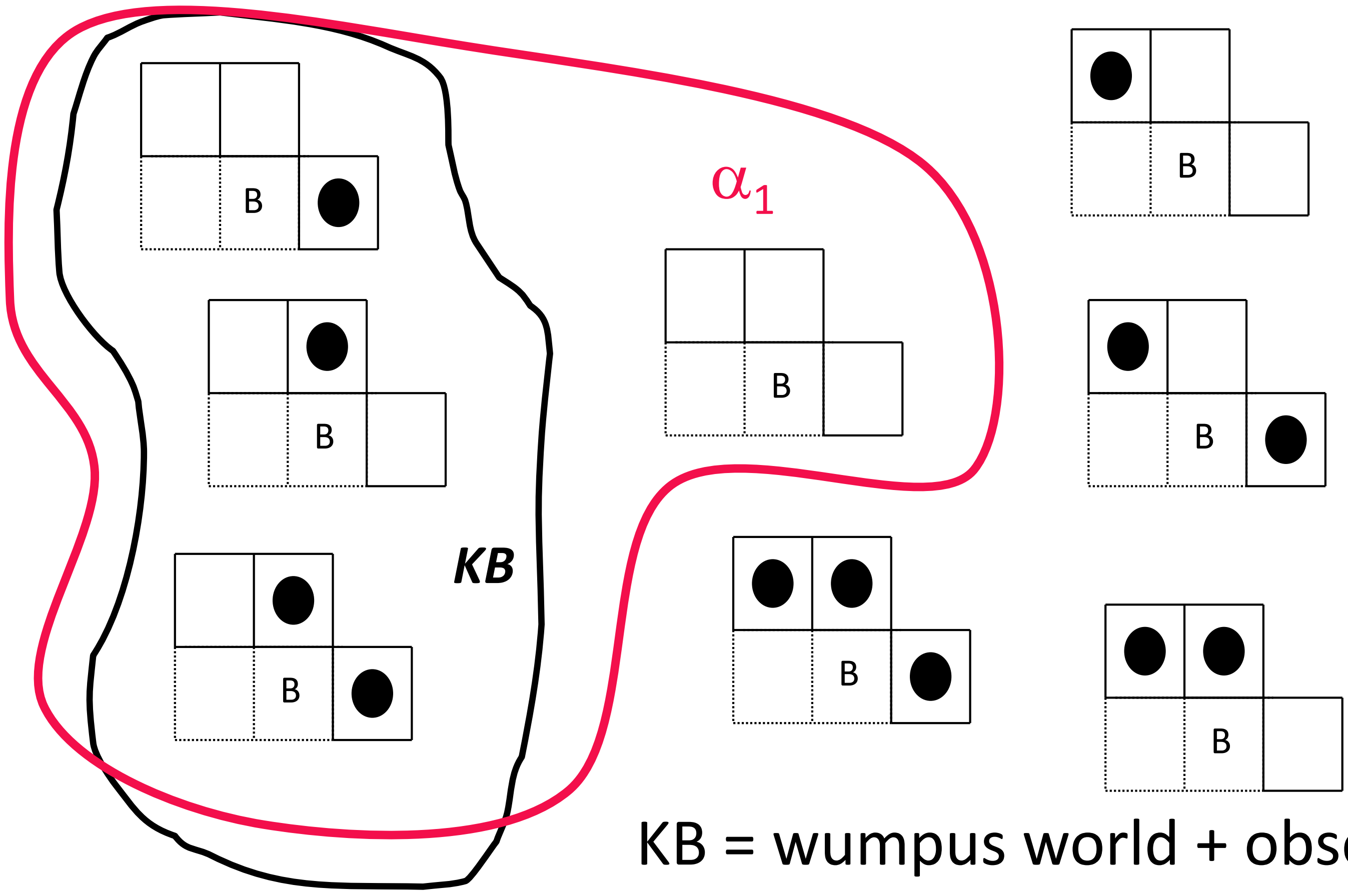
Model the presence of pits in squares [1,2][2,2] and [3,1].



Wumpus Models



Wumpus Models

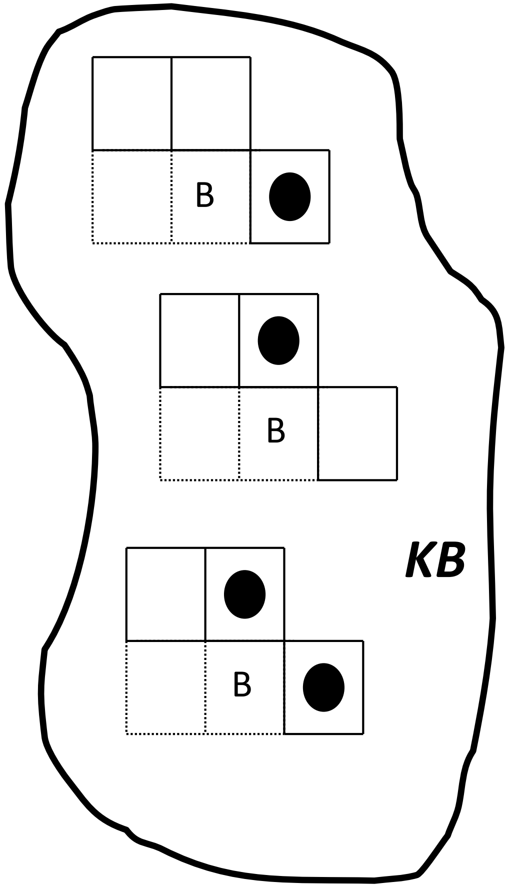


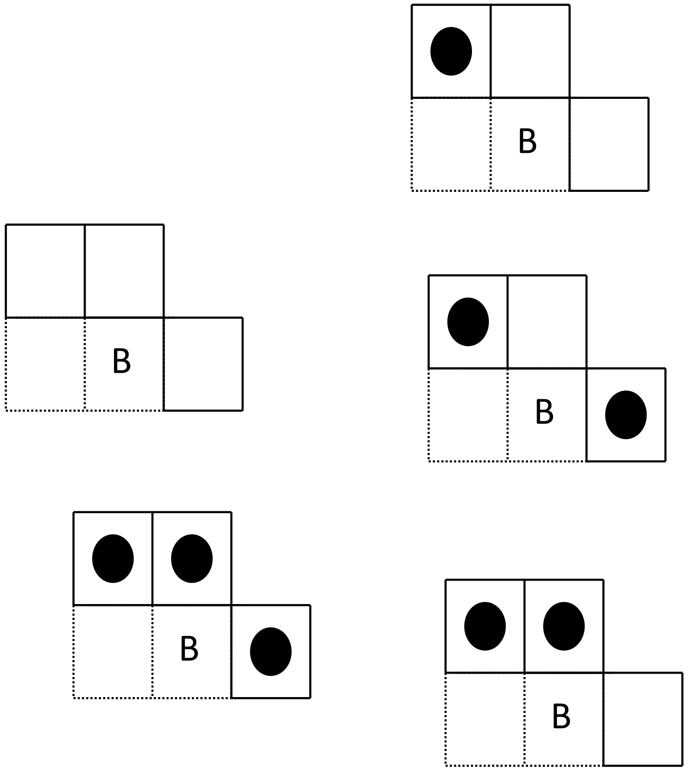
KB = wumpus world + observations

α_1="[1,2] is safe"

KB |= α_1, proved by model checking

Wumpus Models



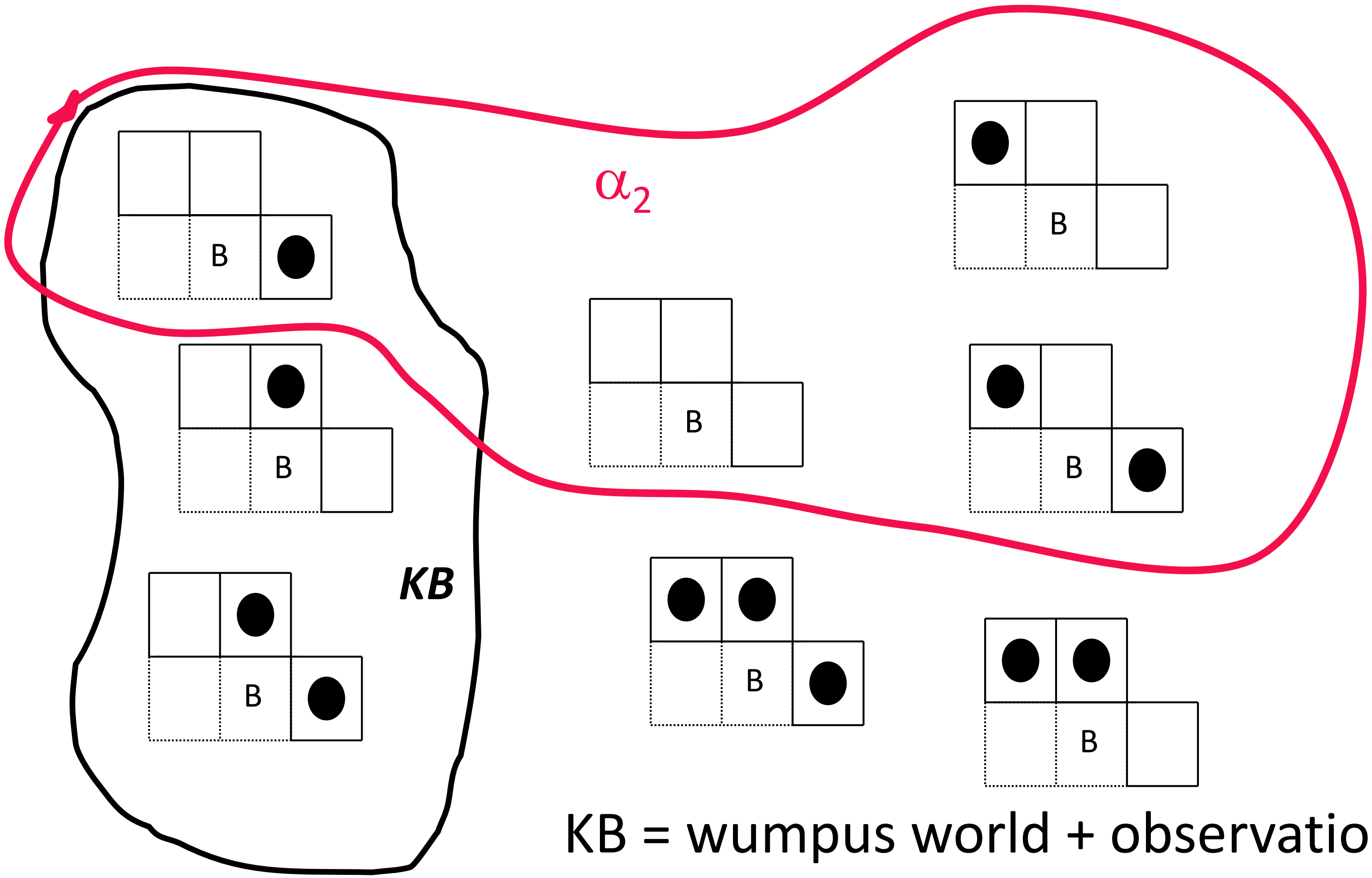


KB = wumpus world + observations

α_2="[2,2] is safe"

KB |= α_2 ??

Wumpus Models

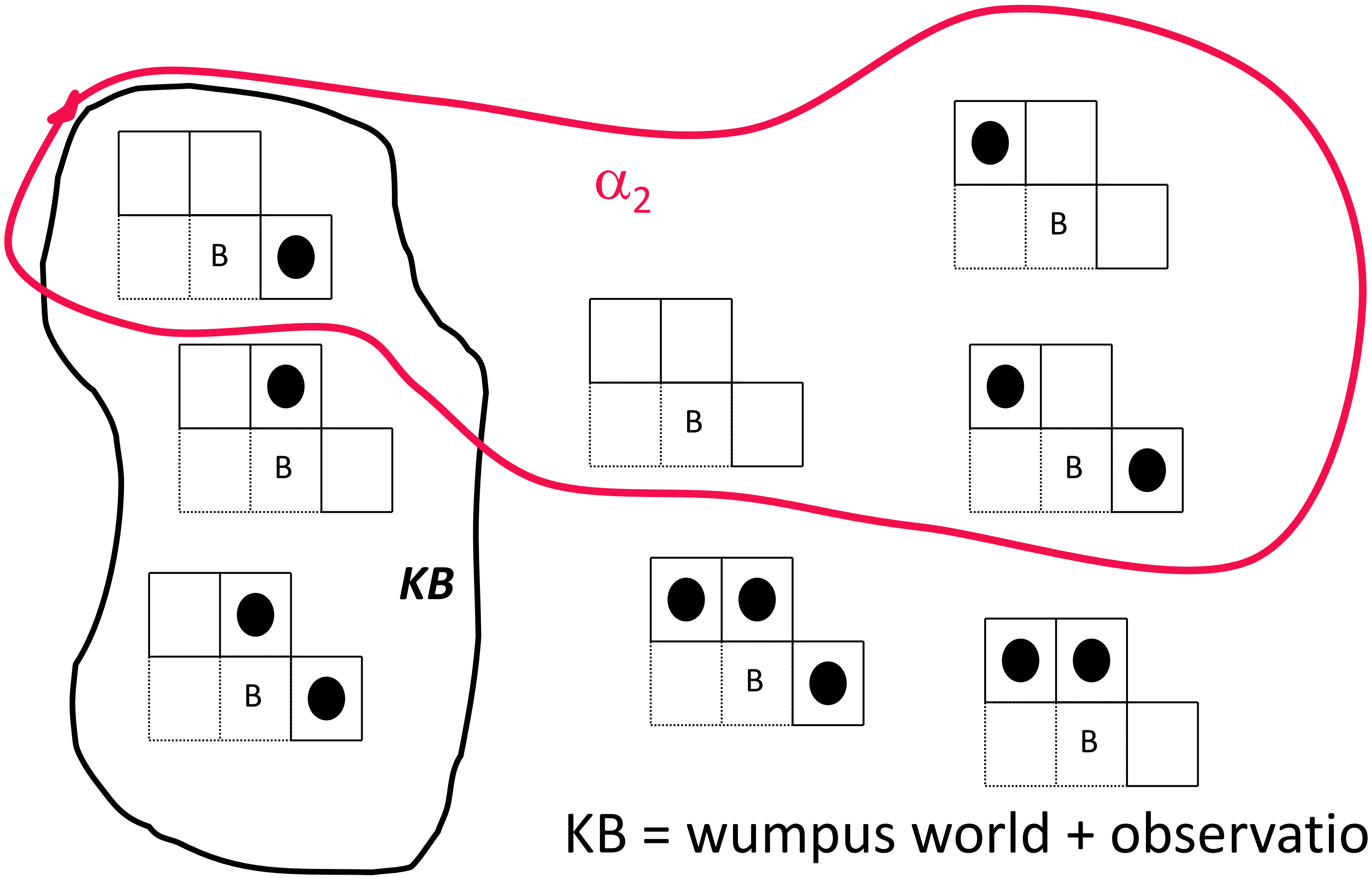


KB = wumpus world + observations

α_2="[2,2] is safe"

KB |= α_2 ??

Wumpus Models



KB = wumpus world + observations

α_2="[2,2] is safe"

KB $\not\models$ α_2 NOT!

Propositional Logic - Syntax

$$
\begin{array}{rcl}
Sentence & \rightarrow & AtomicSentence \mid ComplexSentence \\
AtomicSentence & \rightarrow & True \mid False \mid P \mid Q \mid R \mid \ldots \\
ComplexSentence & \rightarrow & (\ Sentence\ ) \mid [\ Sentence\ ] \\
& \mid & \neg\ Sentence \\
& \mid & Sentence \land Sentence \\
& \mid & Sentence \lor Sentence \\
& \mid & Sentence \ \Rightarrow \ Sentence \\
& \mid & Sentence \ \Leftrightarrow \ Sentence \\
\\
\textsc{Operator Precedence} & : & \neg, \land, \lor, \Rightarrow, \Leftrightarrow
\end{array}
$$

Propositional Logic - Semantics

- Defines the rules for determining the truth of a sentence respect to particular model
- Model simply fixes the truth value
- For example:
- m = {$P_{1,2}$ = *false*, $P_{2,2}$ = *false*, $P_{2,1}$ = *True*}

  Or: m = {$P_{1,2}$ = *false*, $P_{2,2}$ = *True*, $P_{2,1}$ = *false*}
- $P_{1,2}$ is just a symbol. It can mean anything.
- Truth value is computed recursively according to...

Propositional Logic - Semantics

- ¬P is true if P is false in m (negation)
- P ∧ Q is true iff both P and Q are true in m (conjunction)
- P ∨ Q is true iff either P or Q are true in m (disjunction)
- P →Q is true unless P is true and Q is false (implication)
- P ↔ Q is true iff P and Q are both true or both false (biconditional)

Propositional Logic - Semantics

in the model $m = \{P_{1,2} = \mathit{false}, P_{2,2} = \mathit{false}, P_{2,1} = \mathit{True}\}$

Evaluate $\neg P_{1,2} \wedge P_{2,2} \vee P_{3,1}$

Evaluate it for $m = \{P_{1,2} = \mathit{true}, P_{2,2} = \mathit{true}, P_{2,1} = \mathit{false}\}$

Propositional Logic

- Truth table – a (simple) representation of a complex sentence by enumerating its truth in terms of the possible values of each of its symbols.
- Truth table for connectives:

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*

A Simple KB - Definitions

For each location [*x, y*]

- $P_{x,y}$ is true if there's a pit in [*x, y*]
- $W_{x,y}$ is true if there is a Wumpus in [*x, y*], dead or alive
- $B_{x,y}$ is true if the agent perceives a breeze in [*x, y*]
- $S_{x,y}$ is true if the agent perceives a stench in [*x, y*]

A Simple KB - Rules

- For the Wumpus world in general.
  - $R_1 : \neg P_{1,1}$
  - $R_2 : B_{1,1} \leftrightarrow (P_{1,2} \vee P_{2,1})$
  - $R_3 : B_{2,1} \leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{3,1})$
- Now, after visiting [1,1], [1,2] and [2,1]
  - $R_4 : \neg B_{1,1}$
  - $R_5 : B_{2,1}$
- KB = $R_1 \wedge R_2 \wedge R_3 \wedge R_4 \wedge R_5$

Inference

- Main goal: decide whether KB $\models \alpha$
- Want to find whether KB says there's no pit in [1, 2]
- That is, does KB $\models \neg P_{1,2}$?
- We say that $\neg P_{1,2}$ is a sentence
- α can be a much more complex query

Inference – Simple Method

- enumerate the models
- for each model, check that:
- if it is true in α is has to be true in KB

In the Wumpus world: 7 relevant symbols:

$B_{1,1}, B_{2,1}, P_{1,1}, P_{1,2}, P_{2,1}, P_{2,2}, P_{3,1}$

2^7 = 128 models. Only 3 are true

Inference - All Possible Models

$B_{1,1}$	$B_{2,1}$	$P_{1,1}$	$P_{1,2}$	$P_{2,1}$	$P_{2,2}$	$P_{3,1}$	R_1	R_2	R_3	R_4	R_5	KB
false	*false*	*false*	*false*	*false*	*false*	*false*	*true*	*true*	*true*	*true*	*false*	*false*
false	*false*	*false*	*false*	*false*	*false*	*true*	*true*	*true*	*false*	*true*	*false*	*false*
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
false	*true*	*false*	*false*	*false*	*false*	*false*	*true*	*true*	*false*	*true*	*true*	*false*
false	*true*	*false*	*false*	*false*	*false*	*true*	*true*	*true*	*true*	*true*	*true*	*true*
false	*true*	*false*	*false*	*false*	*true*	*false*	*true*	*true*	*true*	*true*	*true*	*true*
false	*true*	*false*	*false*	*false*	*true*	*true*	*true*	*true*	*true*	*true*	*true*	*true*
false	*true*	*false*	*false*	*true*	*false*	*false*	*true*	*false*	*false*	*true*	*true*	*false*
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
true	*true*	*true*	*true*	*true*	*true*	*true*	*false*	*true*	*true*	*false*	*true*	*false*

Truth Table for Wumpus World KB, consisting of $2^7 = 128$ rows, one each for the different assignments of truth values to the 7 proposition symbols $B_{1,1}$, …, $P_{3,1}$. KB is true if R_1 through R_5 are true, which occurs just in 3 rows.

Inference - Model Checking Complexity

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

  $symbols \leftarrow$ a list of the proposition symbols in KB and α
  return TT-CHECK-ALL($KB, \alpha, symbols, \{\ \}$)

---

function TT-CHECK-ALL($KB, \alpha, symbols, model$) **returns** *true* or *false*
  if EMPTY?($symbols$) **then**
    if PL-TRUE?($KB, model$) **then return** PL-TRUE?($\alpha, model$)
    else return *true // when KB is false, always return true*
  else do
    $P \leftarrow$ FIRST($symbols$)
    $rest \leftarrow$ REST($symbols$)
    return (TT-CHECK-ALL($KB, \alpha, rest, model \cup \{P = true\}$)
      and
      TT-CHECK-ALL($KB, \alpha, rest, model \cup \{P = false\}$))

PL-TRUE? returns true if a sentence holds within a model.

Inference - Model Checking Complexity

If KB and α contain *n* symbols in all:

- Time complexity: $\boldsymbol{O(2^n)}$
- Space complexity: $\boldsymbol{O(n)}$ because it is depth first.

Inference - Logical Equivalence

$(\alpha \wedge \beta)$	$\equiv$ $(\beta \wedge \alpha)$	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 $\wedge$

Standard logical equivalences. The symbols α, β and γ stand for arbitrary sentences of propositional logic

Inference By Theorem Proving - Concepts

- Logical Equivalence: $\alpha \equiv \beta$ iff $\alpha \models \beta$ and $\beta \models \alpha$
- Validity: A sentence is valid if it is true in **all** models

  e.g. *P* v ¬*P* (also known as tautology)
- Deduction: $\alpha \models \beta$ iff $\alpha \rightarrow \beta$ is valid
- Satisfiability: a sentence is satisfiable if it is true in, or satisfied by, some model.

Inference Theorem Proving - Proofs

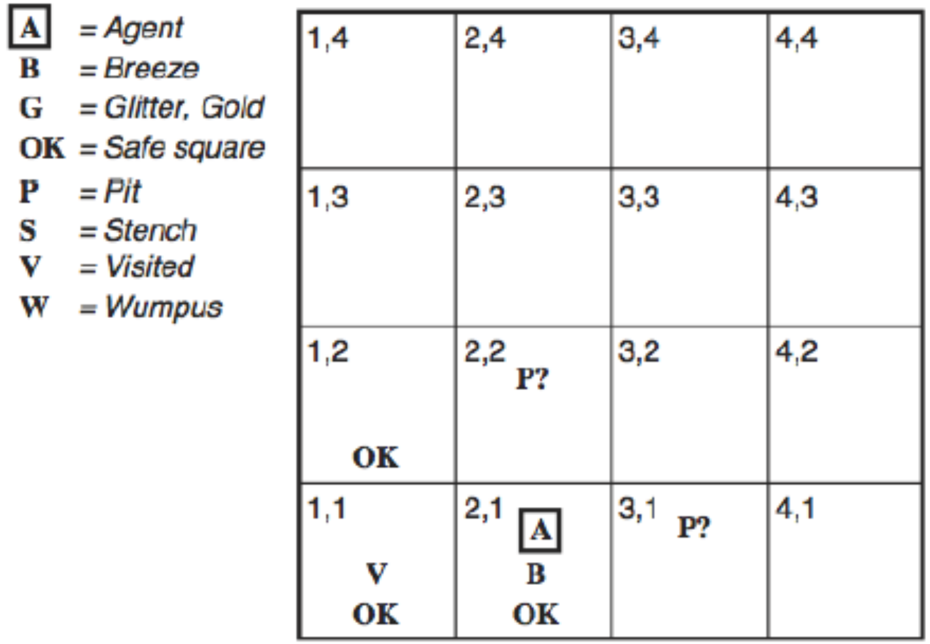
- Inference rules used to derive a proof
- Common Patterns:
  - Modus Pones $\dfrac{\alpha \Rightarrow \beta, \alpha}{\beta}$
  - And-Elimination $\dfrac{\alpha \wedge \beta}{\alpha}$
- Other rules can also be inference rules

$$\frac{\alpha \iff \beta}{(\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)}$$

$$\frac{(\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)}{\alpha \iff \beta}$$

Inference - In our Wumpus World

Is there a pit in 1,2?



- $R_1: \neg P_{1,1}$
- $R_2: B_{1,1} \leftrightarrow (P_{1,2} \vee P_{2,1})$
- $R_3: B_{2,1} \leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{3,1})$
- $R_4: \neg B_{1,1}$
- $R_5: B_{2,1}$

Inference – Applied to Wumpus World

We have KB = $R_1 \wedge R_2 \wedge R_3 \wedge R_4 \wedge R_5$. We want to prove $\neg P_{1,2}$

- R_6: $(B_{1,1} \rightarrow (P_{1,2} \vee P_{2,1})) \wedge ((P_{1,2} \vee P_{2,1}) \rightarrow B_{1,1})$ by biconditional elimination to R_2
- R_7: $((P_{1,2} \vee P_{2,1}) \rightarrow B_{1,1})$ by And-Elimination
- R_8: $(\neg B_{1,1} \rightarrow \neg(P_{1,2} \vee P_{2,1}))$ by Contrapositives
- R_9: $\neg(P_{1,2} \vee P_{2,1})$ by Modus Ponens with R_8 and R_4
- R_{10}: $\neg P_{1,2} \wedge \neg P_{2,1}$ by De Morgan's rule

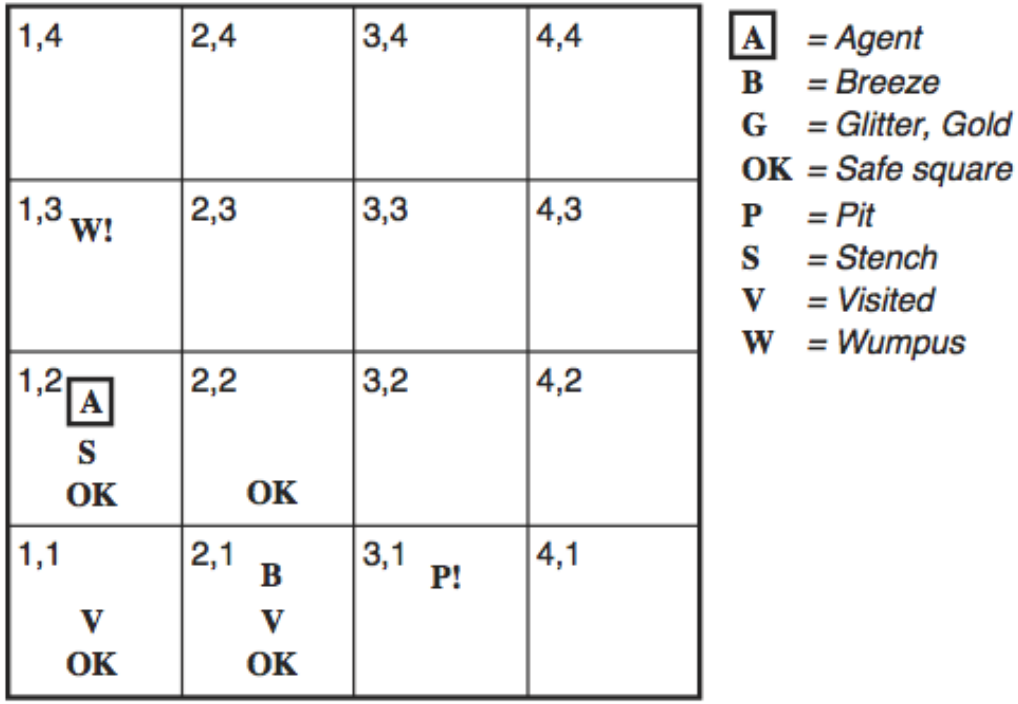
That is, neither $P_{1,2}$ nor $P_{2,1}$ contains a pit.

Inference - As Search

- **Intial State:** The initial Knowledge Base
- **Actions:** The set of all the inference rules applied to all sentences that match top half
- **Result:** Add sentence in the bottom half of the inference rule
- **Goal:** The goal is a state that contains sentence we want to prove

Inference – By Resolution

- Let's say agent returns to [1,1] from [2,1] and goes to [1,2]



We add:

- $R_{11} : \neg B_{1,2}$
- $R_{12} : B_{1,2} \leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{1,3})$

Inference – By Resolution

We can continue using same process as earlier.

- R_{13} : $\neg P_{2,2}$ Contrapositive R_{12} and AND elimination
- R_{14} : $\neg P_{1,3}$ Same as above.
- R_{15} : $P_{1,1} \vee P_{2,2} \vee P_{3,1}$ bi-conditional elimination R_3 and modus ponens R_5

And the literal $\neg P_{2,2}$ in R_{13} resolves with $P_{2,2}$ in R_{15} to give the resolvent

- R_{16} : $P_{1,1} \vee P_{3,1}$
- more generally...

$$\frac{A \vee B, \neg A \vee C}{B \vee C}$$

Resolution - Conjunctive Normal Form (CNF)

- Every sentence in propositional logic can be expressed as conjunctions of disjunctions of literals.
- e.g. $(A \lor B) \land (\neg C \lor D \lor \neg E) \land \ldots$

$B_{1,1} \leftrightarrow (P_{1,2} \lor P_{2,1})$ in CNF?

- Eliminate $\leftrightarrow$ replacing $\alpha \leftrightarrow \beta$ with $(\alpha \rightarrow \beta)$ ^ $(\alpha \rightarrow \beta)$
  – $(B_{1,1} \rightarrow (P_{1,2} \lor P_{2,1})) \land ((P_{1,2} \lor P_{2,1})\ ) \rightarrow B_{1,1})$
- Eliminate $\rightarrow$ by replacing $\alpha \rightarrow \beta$ with $\neg\alpha \lor \beta$
  – $(\neg B_{1,1} \lor P_{1,2} \lor P_{2,1}) \land (\neg(P_{1,2} \lor P_{2,1}) \lor B_{1,1})$
- Symbol $\neg$ should appear next to each literal: DeMorgan
  – $\neg(\alpha \lor \beta) \equiv \neg\alpha \land \neg\beta$ and $\neg(\alpha \land \beta) \equiv \neg\alpha \lor \neg\beta$
  – $(\neg B_{1,1} \lor P_{1,2} \lor P_{2,1}) \land ((\neg P_{1,2} \land \neg P_{2,1}) \lor B_{1,1})$
- Distribute $\lor$ over $\land$ and flatten
  – $(\neg B_{1,1} \lor P_{1,2} \lor P_{2,1}) \land (\neg P_{1,2} \lor B_{1,1}) \land (\neg P_{2,1} \lor B_{1,1})$

Resolution

Algorithm works using **proof by contradiction**.

To show KB $\models \alpha$ we show that KB $\wedge \neg\alpha$ is not satisfiable

Apply resolution to KB $\wedge \neg\alpha$ in CNF and Resolve pairs with complementary literals

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

If l_i and m_j are complimentary literals

and add new clauses

until

- there are no new clauses to be added
- two clauses resolve to the empty class, which means KB $\models \alpha$

Resolution algorithm

- Proof by contradiction, i.e., show $KB \land \neg\alpha$ is unsatisfiable

```
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 ∧ ¬α
   new ← { }
   loop do
      for each Ci, Cj in clauses do
         resolvents ← PL-RESOLVE(Ci, Cj)
         if resolvents contains the empty clause then return true
         new ← new ∪ resolvents
      if new ⊆ clauses then return false
      clauses ← clauses ∪ new
```

Resolution example

Say the agent is in [1,1], no breeze, so no pits can be in there.

- $\alpha = \neg P_{1,2}$
- $KB = R_2 \land R_4$
- $KB = (B_{1,1} \leftrightarrow (P_{1,2} \lor P_{2,1})) \land \neg B_{1,1}$
- $KB \land \neg\alpha = (\neg P_{2,1} \lor B_{1,1}) \land (\neg B_{1,1} \lor P_{1,2} \lor P_{2,1}) \land (\neg P_{1,2} \lor B_{1,1}) \land (\neg B_{1,1}) \land (P_{1,2})$