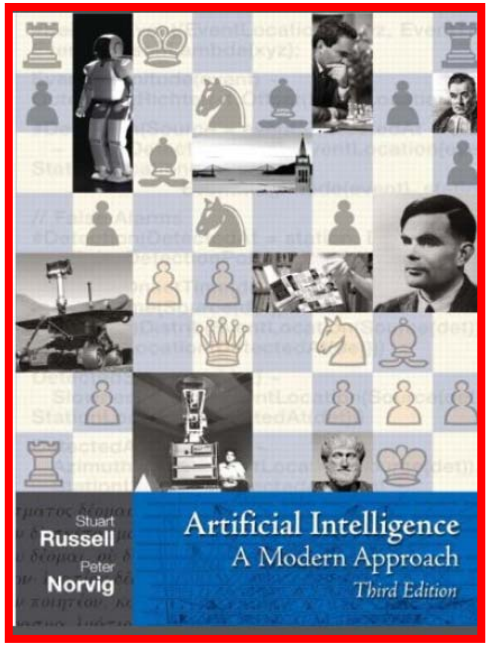


(Chapter-7)
Logical Agents

Yanmei Zheng

Outline

- Agent Case (Wumpus world)
- Knowledge-Representation
- Logic in general - models and entailment
- Propositional (Boolean) logic

Knowledge Bases

Inference Engine	←	Domain-independent Algorithm
Knowledge Base	←	Domain-Specific Content

- Knowledge base = set of sentences in a formal language
- Declarative approach to building an agent (or other system):
  1. Tell it what it needs to know
  - Then it can Ask itself what to do - answers should follow from the KB
- Agents can be viewed at the knowledge level
  i.e., what they know, regardless of how implemented
- Or at the implementation level
  1. i.e., data structures in KB and algorithms that manipulate them

Propositional Logic: A Very Simple Logic

Sentence → AtomicSentence | ComplexSentence

AtomicSentence → **True** | **False** | Symbol

Symbol → P | **Q** | **R** |...

ComplexSentence → ¬Sentence

| (Sentence ∧ Sentence)

| (Sentence ∨ Sentence)

| (Sentence □ Sentence)

| (Sentence □ Sentence)

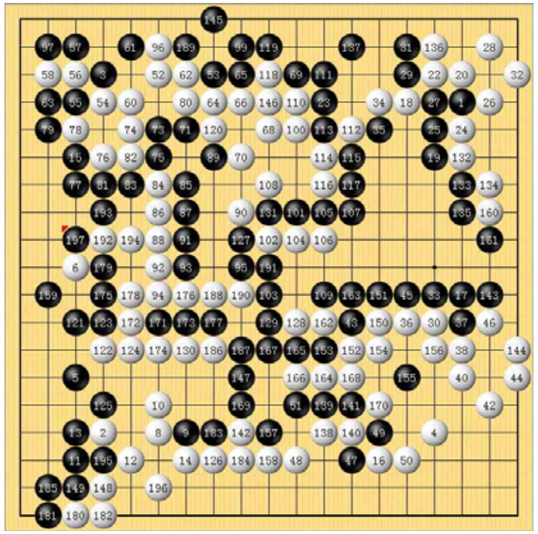
Dr.Yanmei Zheng

A simple knowledge-based agent

```
function KB-AGENT(percept)returns an action
    static:KB,a knowledge base
           t,a couter,initially 0,idicating time
    TELL(KB,MAKE-PERCEPT-SENTENCE(percept,t))
    action←ASK(KB,MAKE-ACTION-QUERY(t))
    TELL(KB,MAKE-ACTION-SENTENCE(action,t))
    t←t+1
    return action
```

The agent must be able to:

1. Represent states, actions, etc.
2. Incorporate new percepts
3. Update internal representations of the world
4. Deduce hidden properties of the world
5. Deduce appropriate actions



What will we learn?

- We design (knowledge based) agents that can form representations of a complex world, use a process of inference to derive new representation about the world, and use these new representations to deduce what to do.

What is the central component of knowledge based agent?

Knowledge Based Agents {7.1}

- The **knowledge base(KB)** is the central component
- KB is a set of **sentences** representing assertions about the world
- Sentences are represented with a **knowledge representation language**
- Two operations on KBs
- **Tell** and **Ask**
- Both may involve inferencing, deriving new sentences from old

Knowledge Based Agents {7.1}

- **Procedural**, e.g.: functions
  1. Such knowledge can only be used in one way --by executing it
- **Declarative**, e.g.: constraints
  1. It can be used to perform many different sorts of inferences
- Logic is a _____________ language

Knowledge Based Agents {7.1}

- **Procedural**, e.g.: functions
  1. Such knowledge can only be used in one way --by executing it
- **Declarative**, e.g.: constraints
  1. It can be used to perform many different sorts of inferences
- **Logic** is a Declarative language to：
  1. **Assert** sentences representing facts that hold in a world W (these sentences are given the value true)
  2. **Deduce** the true/false values to sentences representing other aspects of W

logic is a declarative language{7.1}

- **Propositional logic sentence**

$$A \wedge B \Rightarrow C$$

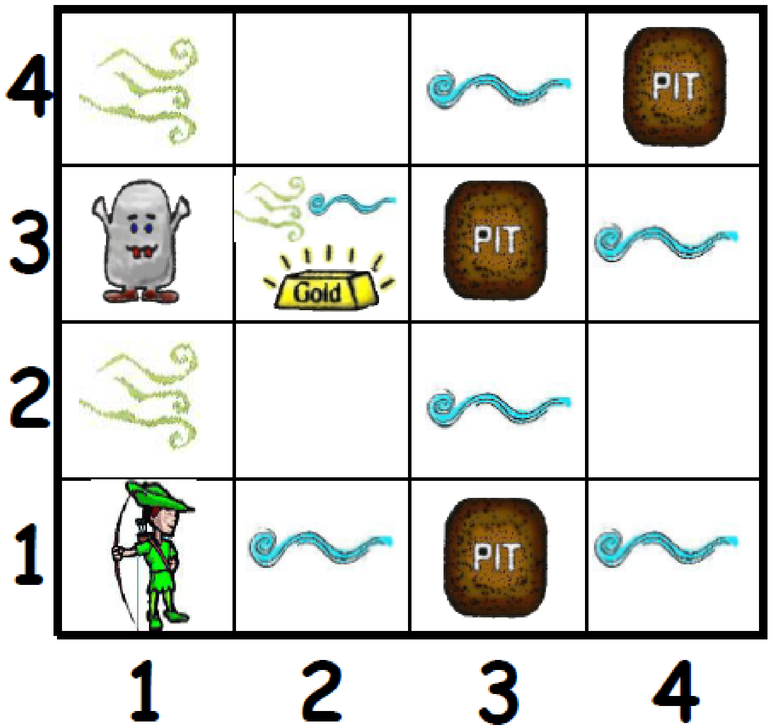
- **First-order predicate logic sentence**

$(\forall x)(\exists y)\mathrm{Mother}(y, x)$

$$(\forall x)(\exists y)\mathrm{Mother}(y, x)$$

Agent Case (Wumpus world)

The Wumpus World



✓The wumpus world is a cave consisting of rooms connected by passageways.

✓ Lurking somewhere is the **wumpus**, which eats anyone who enters its room.

✓The wumpus can be shot by **an agent**, but the agent has only **one arrow**.

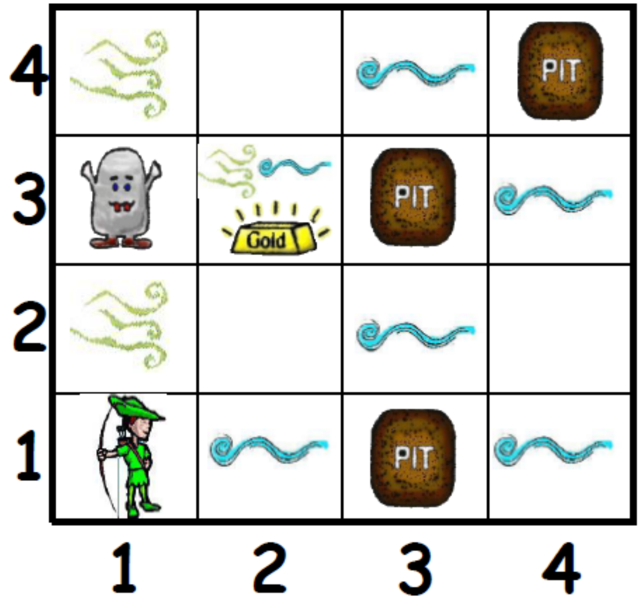
✓Some rooms contain **pits** that will trap anyone who wanders into these rooms (except for the wumpus, which is too big).

✓The only mitigating feature of this bleak environment is the possibility of finding a heap of **gold**.

✓**What is the PEAS of this world?**

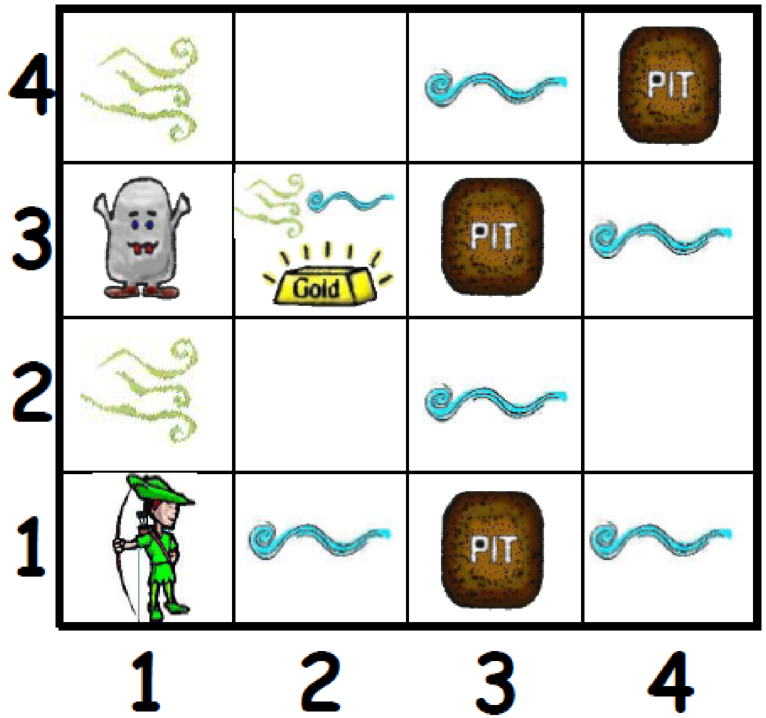
PEAS of Wumpus World {7.2}

- Performance measure
  - Gold→ +1000, death →-1000
  - -1 per step, -10 for using the arrow
- Environment
  - A 4×4 grid of rooms. Start at [1,1], facing to the right.
  - The locations of the gold and the wumpus are chosen randomly (and uniformly) from the squares other than the start square.
  - In addition, each square other than the start can be a pit, with probability 0.2.
  - Squares adjacent to wumpus are smelly
  - Squares adjacent to pit are breezy(windy)
  - Glitter iff gold is in the same square
  - 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



PEAS of Wumpus World {7.2}

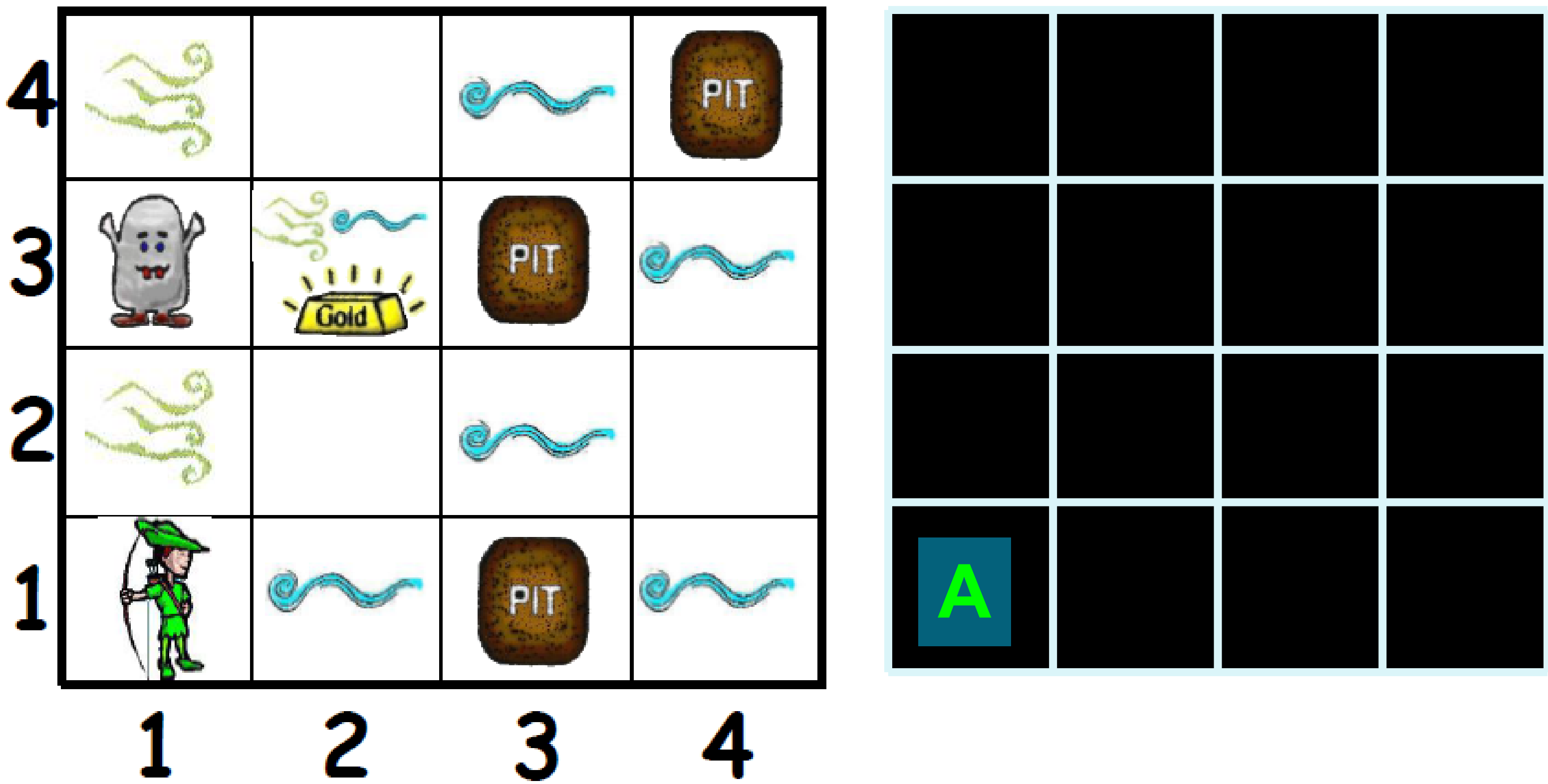
- Actuators
  1. Left turn
  2. Right turn
  3. Forward
  4. Grab
  5. Shoot
- Sensors
  1. Stench , in,4-adjacent
  2. Breeze (for movement) , 4-adjacent
  3. Glitter (gold), in
  4. Bump (hit), walking into a wall
  5. Scream, wumpus killed, anywhere



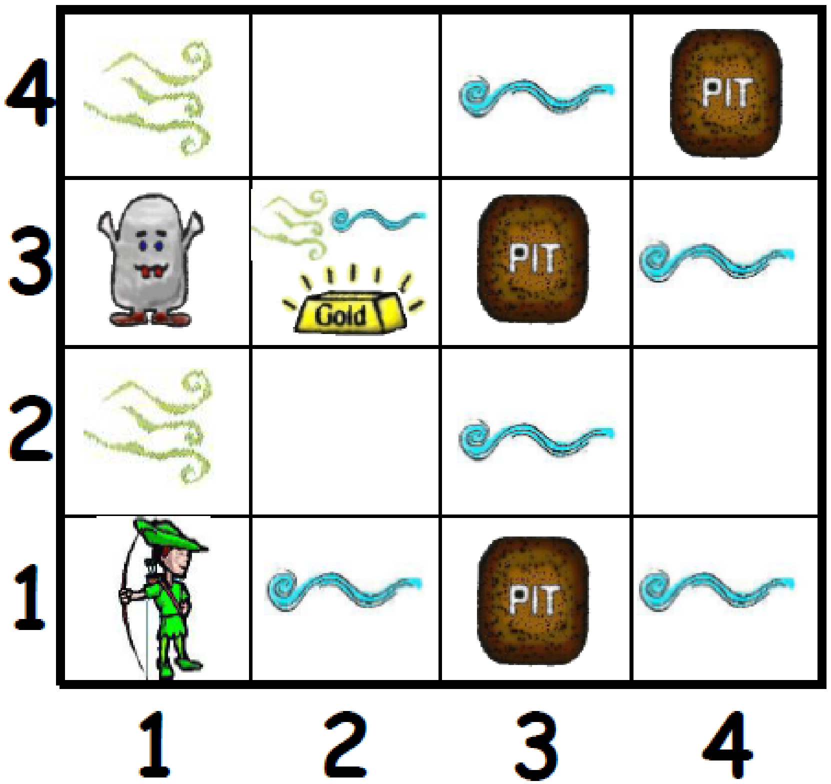
Wumpus world characterization

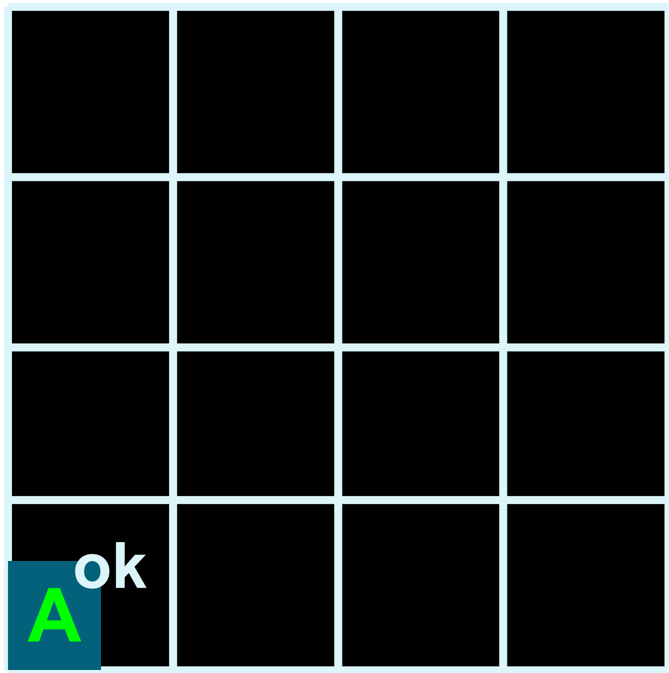
- **Fully Observable** No – only local perception
  Deterministic Yes – outcomes exactly specified
  Episodic No – sequential at the level of actions
  Static Yes – Wumpus and Pits do not move
  Discrete Yes
- **Single-agent?** Yes – Wumpus is essentially a natural feature

Exploring Wumpus World



Exploring Wumpus World



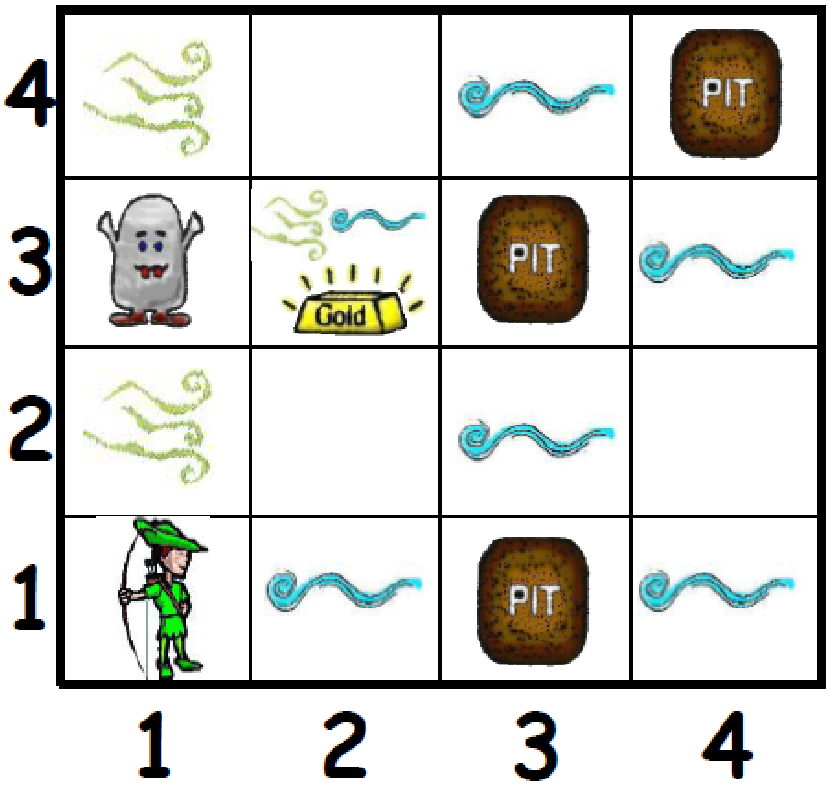


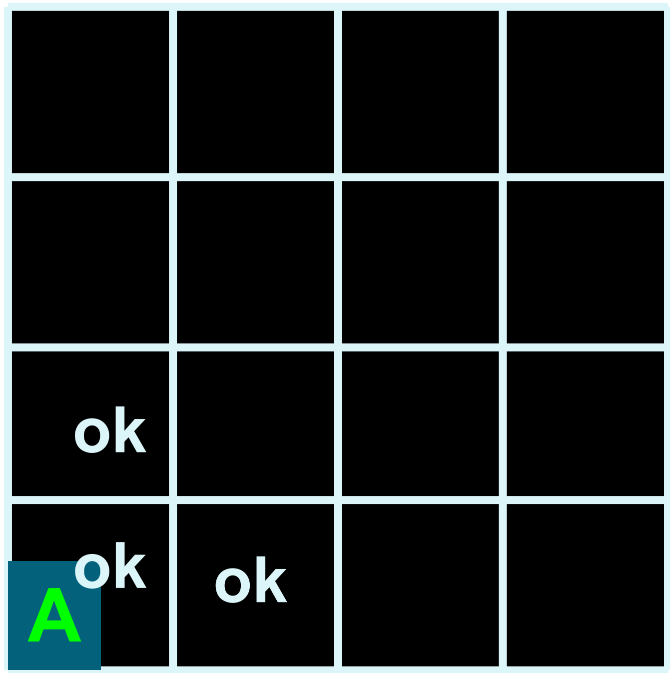
Ok because:

Haven't fallen into a pit.

Haven't been eaten by a Wumpus.

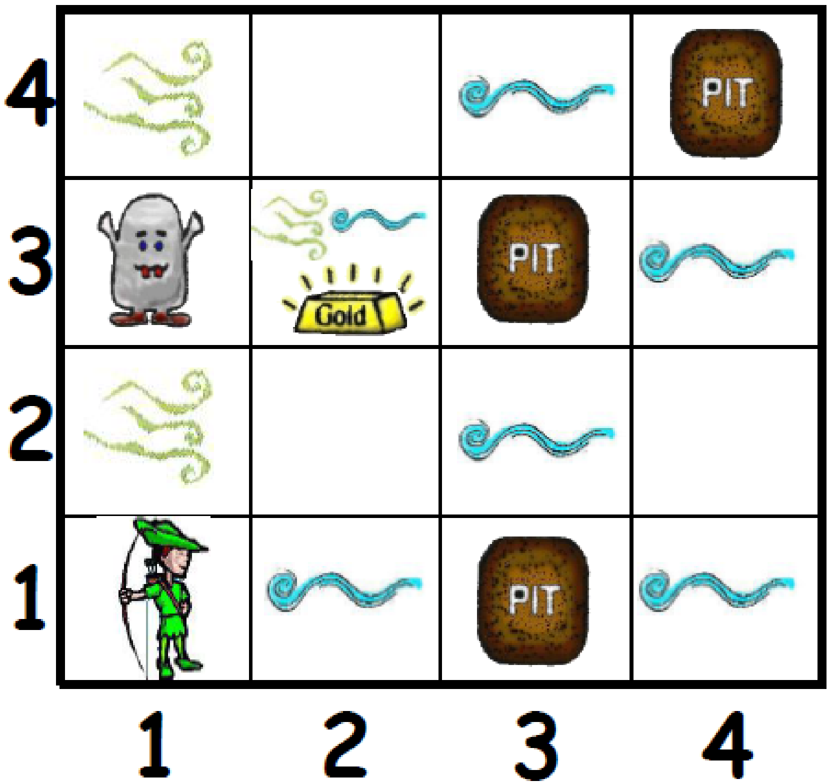
Exploring Wumpus World

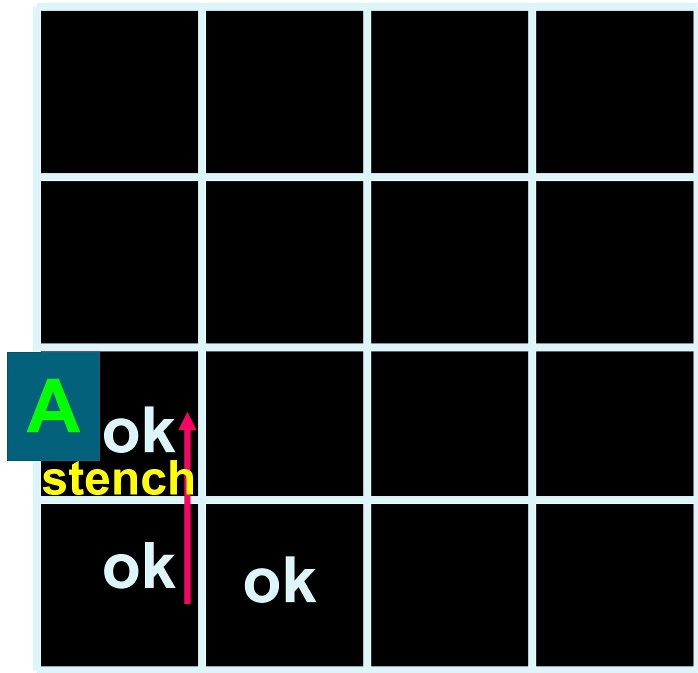




OK since
no Stench,
no Breeze,
neighbors are safe (OK).

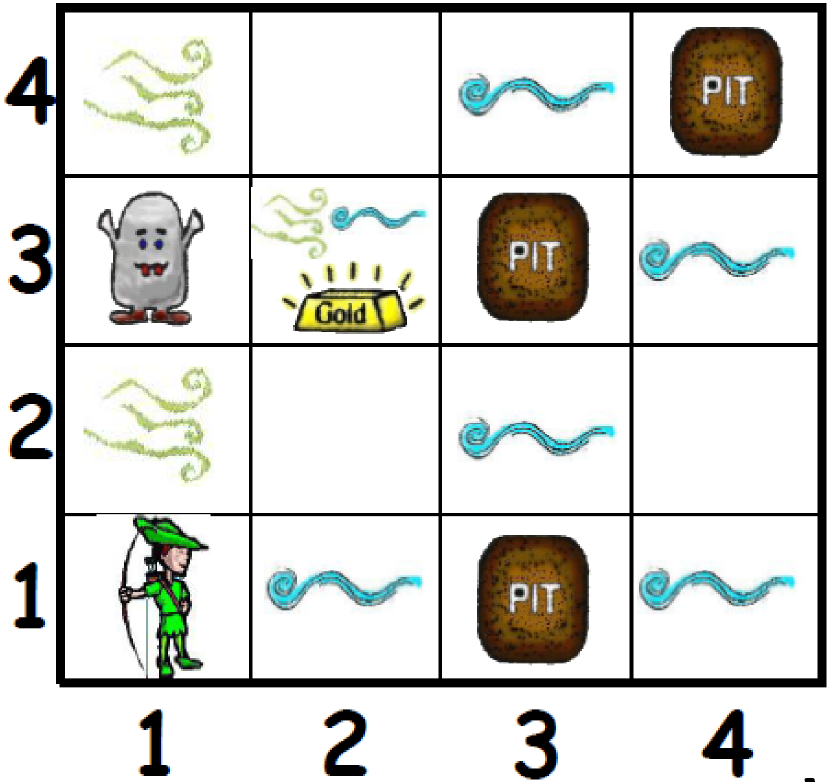
Exploring Wumpus World

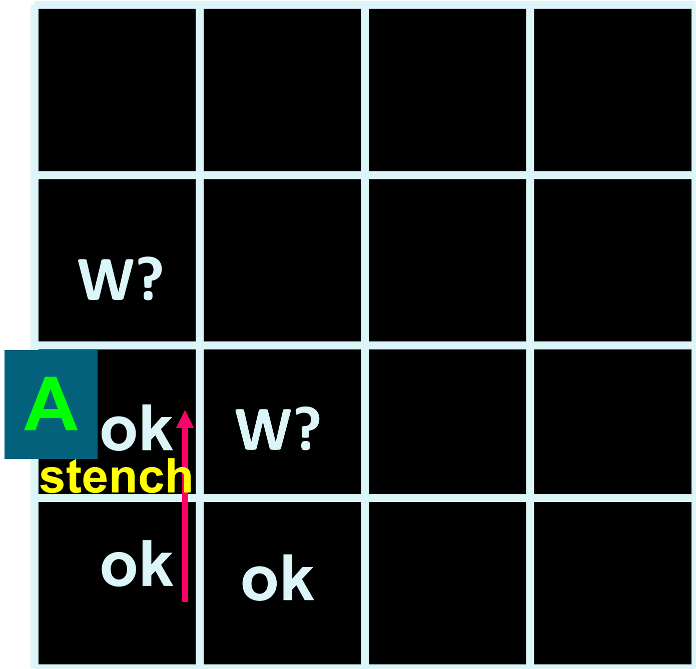




We move and smell a stench.

Exploring Wumpus World

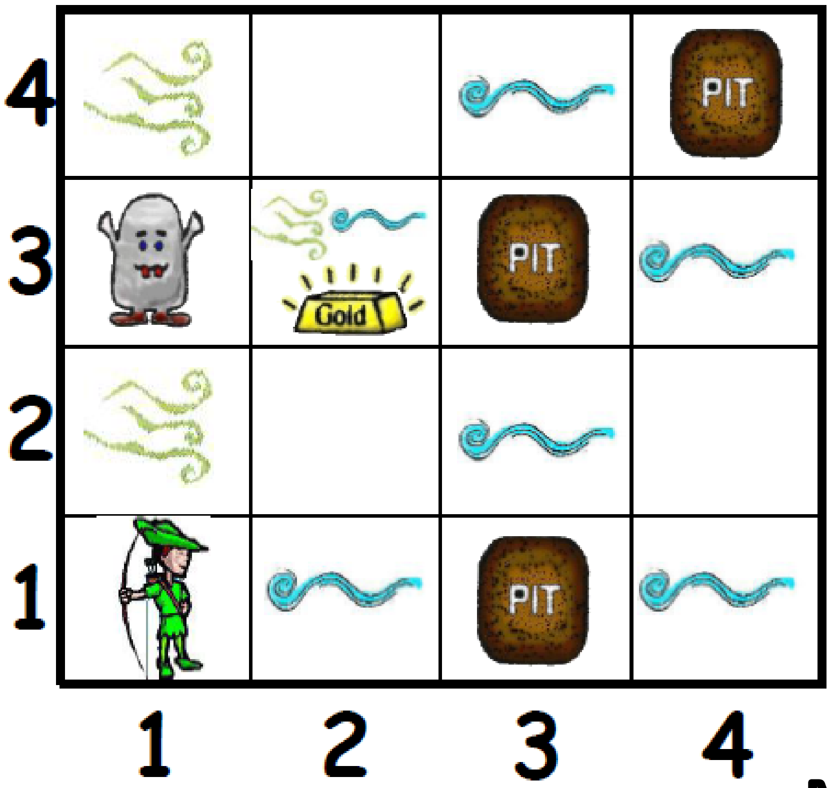




We can infer the following.

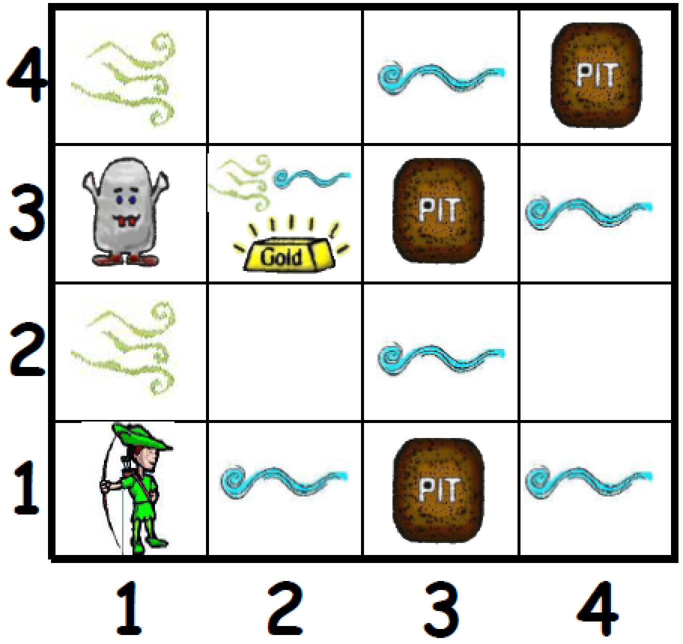
Note: square (1,1) remains OK.

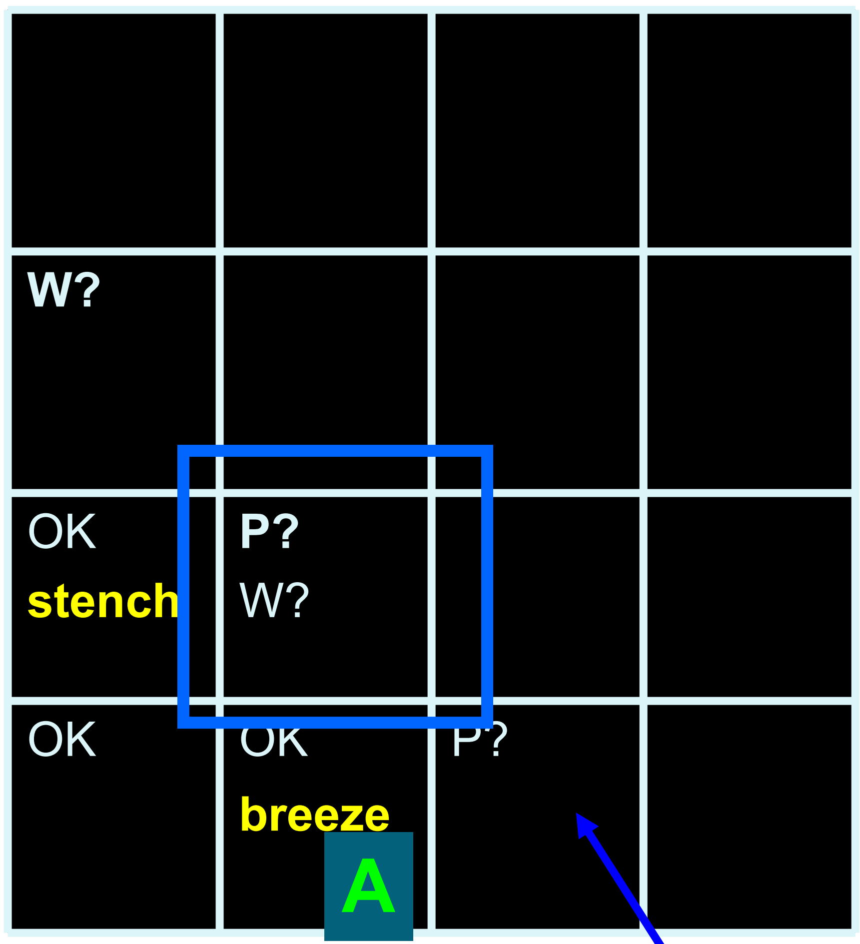
Exploring Wumpus World



Move and feel a breeze
What can we conclude?

Exploring Wumpus World





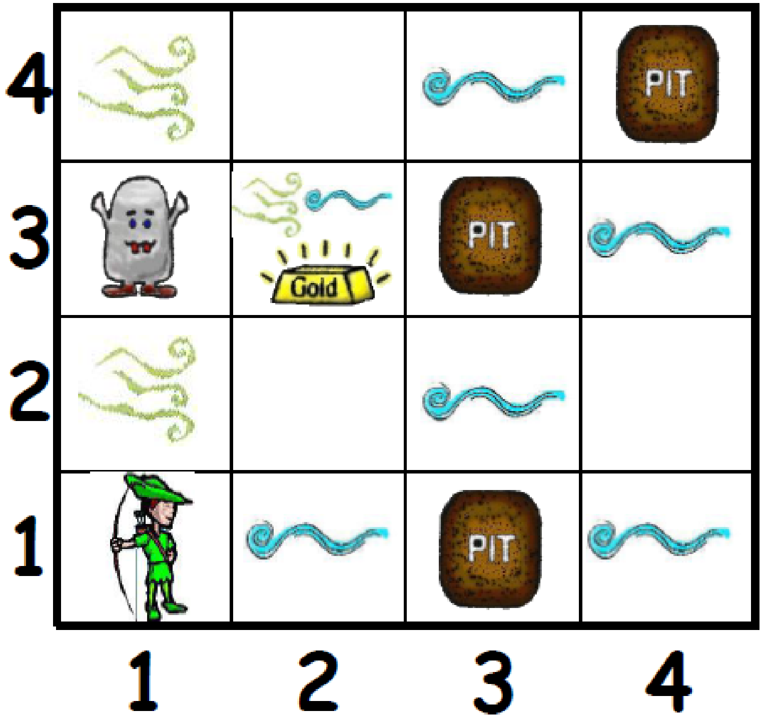
But, can the 2,2 square really have either a Wumpus or a pit?

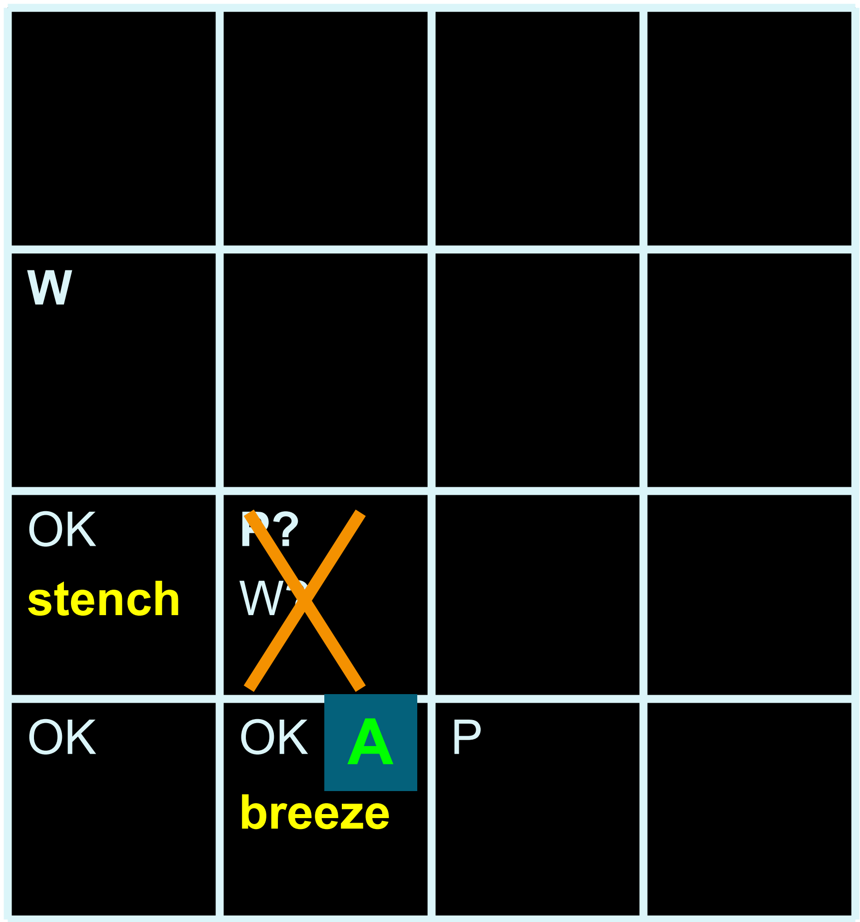
NO!

And what about the other P? and W? squares

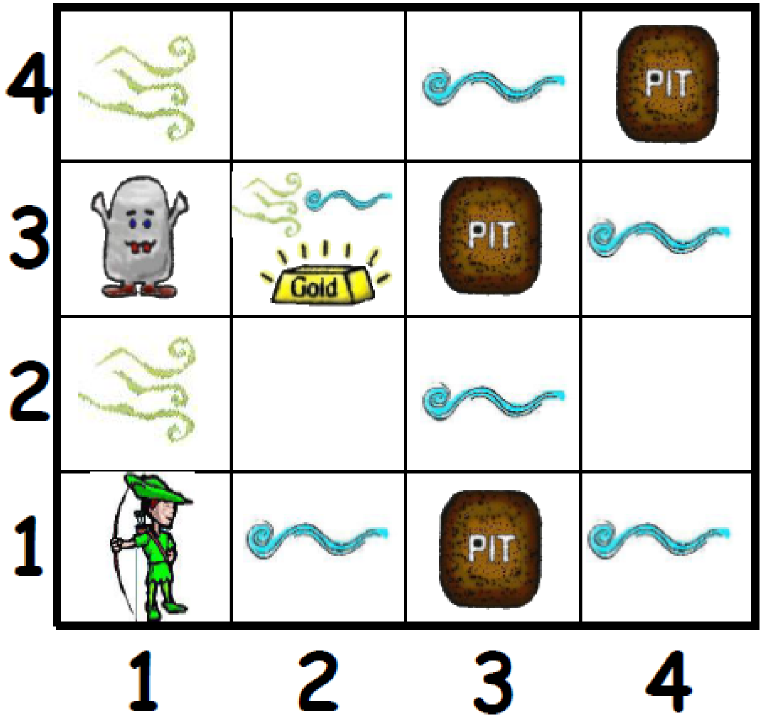
Dr.Yanmei Zheng

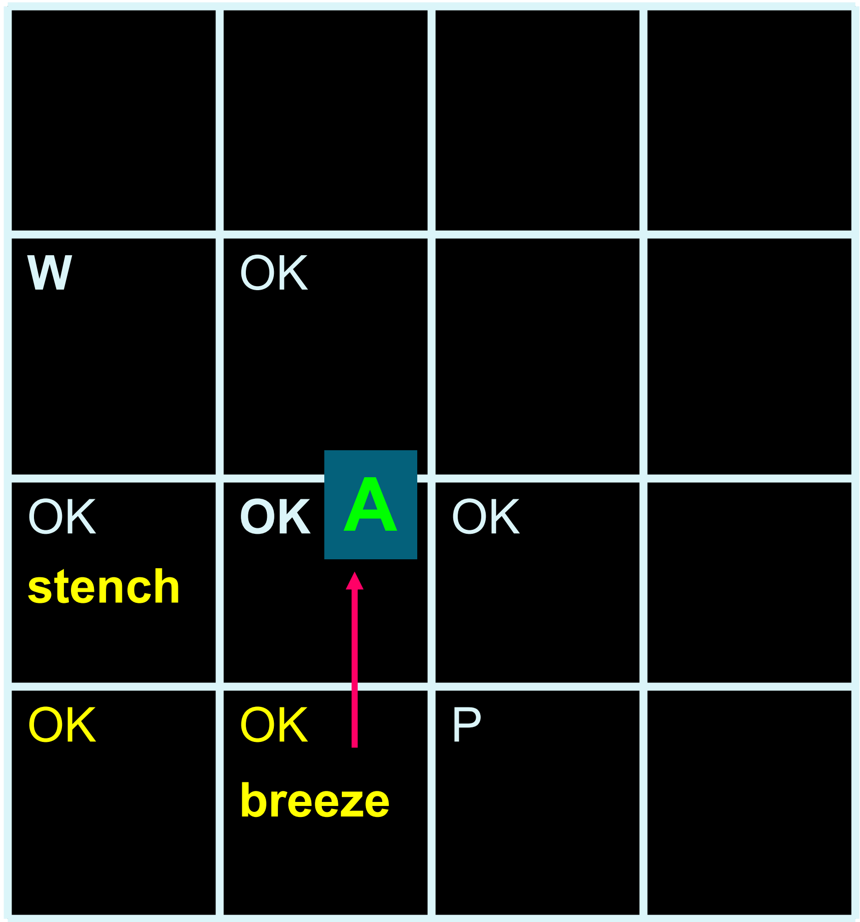
Exploring Wumpus World





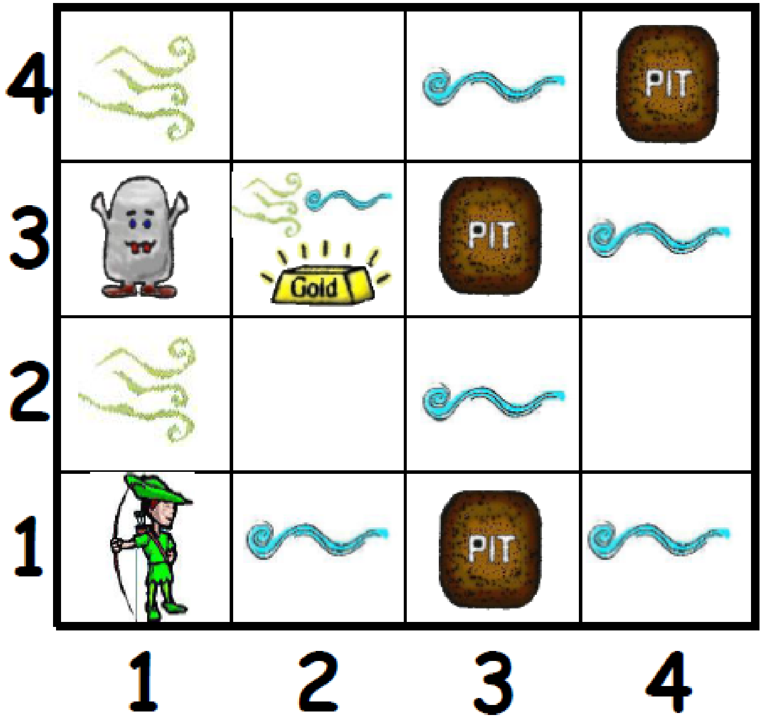
Exploring Wumpus World

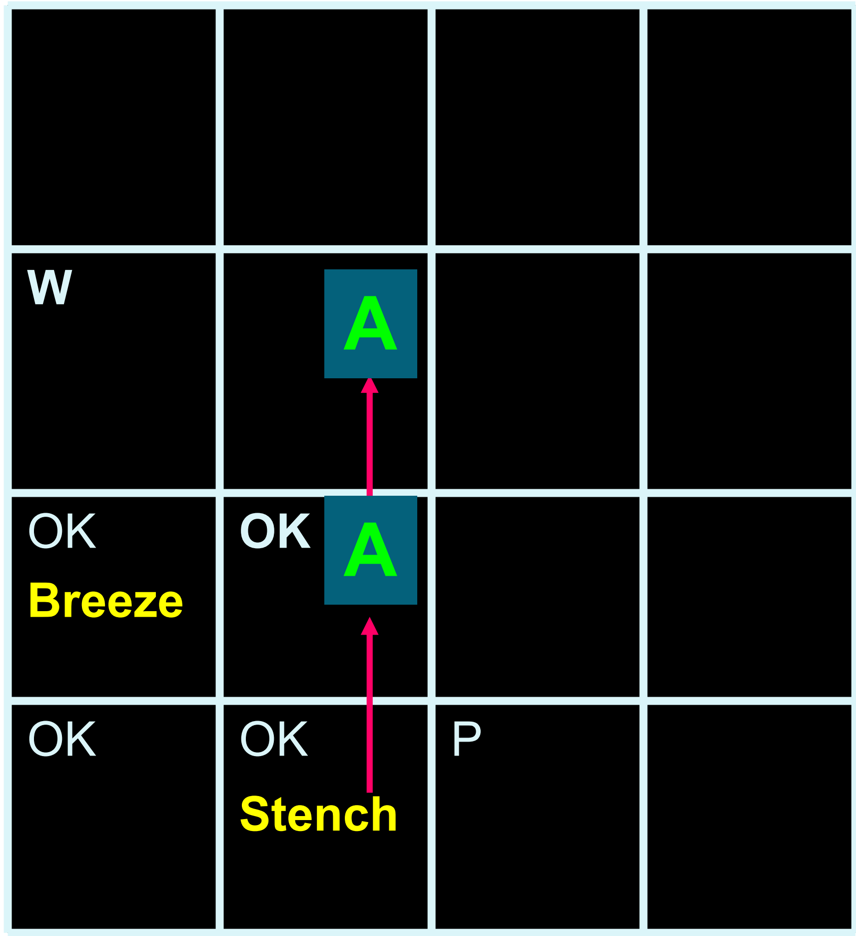




Dr.Yanmei Zheng

Exploring Wumpus World





…

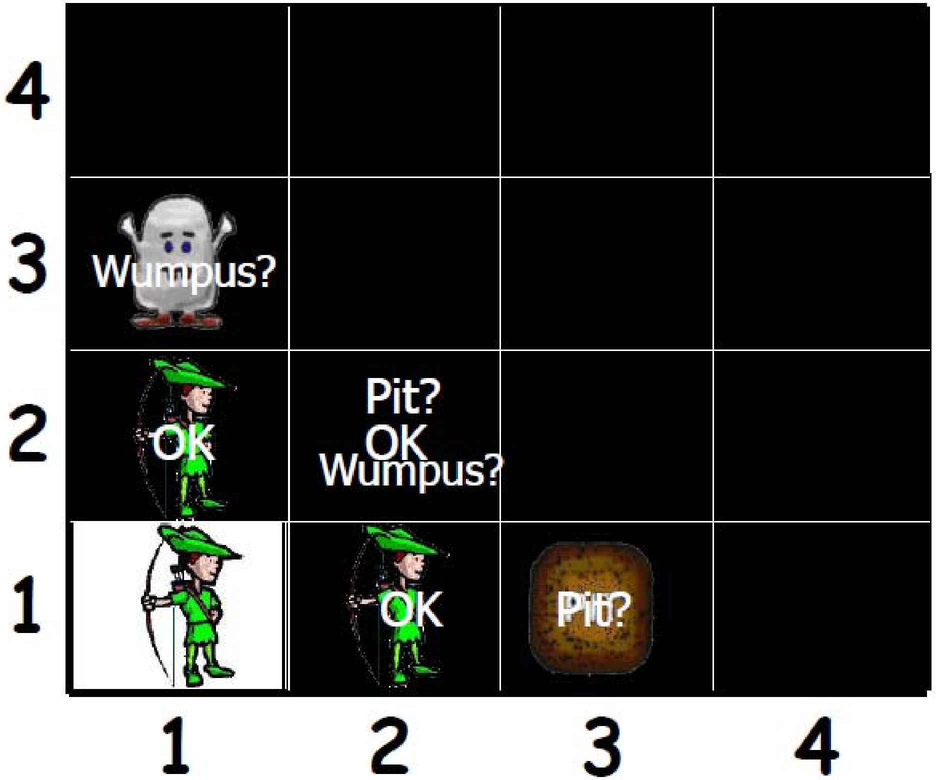
And the exploration continues onward until the gold is found.

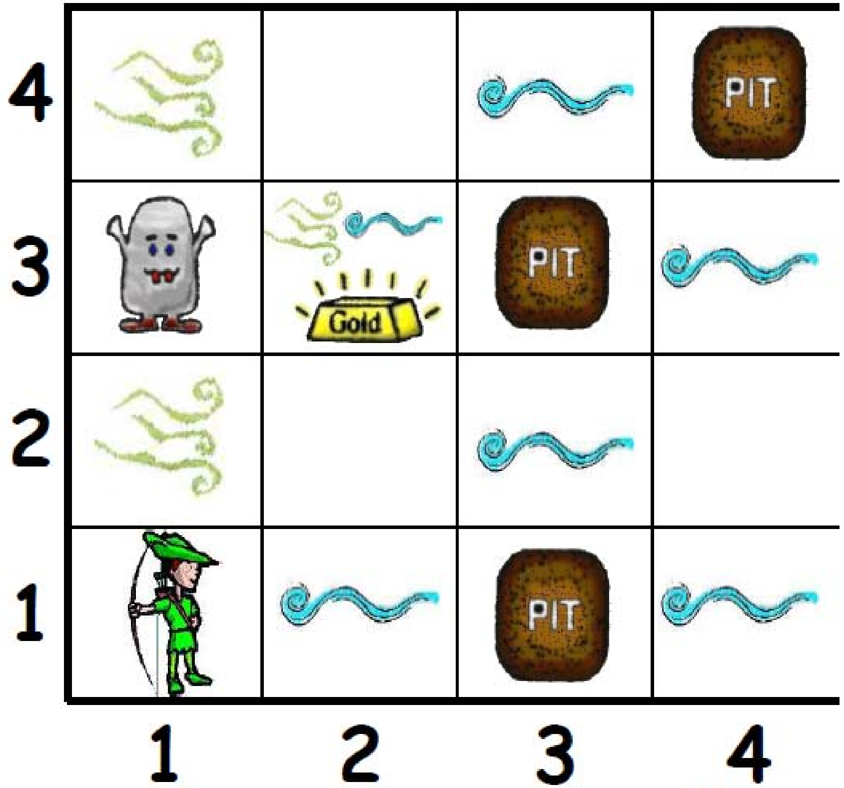
…

Dr.Yanmei Zheng

Exploring the Wumpus World {7.2}

- Agent needs to know which actions are safe.
- reasoning
- Wumpus world can be solved using logic.





What does an agent need to perform well in the wumpus world?

- It needs to represent the knowledge about the wumpus world: ex, stench → wumpus nearby.
- It needs to incorporate new knowledge that it discovers when exploring the cave →feeling a breeze in the current square.
- Deduce from the preceding information the appropriate actions (safe and rewarding) :
- The agent needs a knowledge-base.

Knowledge bases

What does an agent need to perform well in the wumpus world?

Each time the agent program is called, it does three things

- ✓ **First**, it **TELLS** the knowledge base what it perceives.
- ✓ **Second**, it **ASKS** the knowledge base what action it should perform
- ✓ **Third**, the agent program **TELLS** the knowledge base which action was chosen, and the agent executes the action.

Knowledge bases

- Knowledge base: set of sentences. Each sentence is expressed in a language called a knowledge representation language.

- Sentence: a sentence represents some assertion about the world.

- Inference: Process of deriving new sentences from old ones.

Types of Knowledge

Declarative knowledge	Concepts Facts Objects	Describes what is known about a problem. Simple statements that are asserted to be either true or false. A list of statements that more fully describes some object or concept (object-attribute-value triplet).

Types of Knowledge

Procedural knowledge	Rules Strategies Agendas Procedures	Describes how a problem is solved. This type of knowledge provides direction on how to do something.

Some General Knowledge Representations

1. **Logical Representations**
2. **Production Rules**
3. **Semantic Networks**
   - **Conceptual graphs, frames**
4. ***Description Logics*** **(not covered in this course)**

Logical Representation

Logic in general

- A language with concrete rules
- Many ways to translate between languages
  1. A statement can be represented in different logics
  2. And perhaps differently in same logic
- Not to be confused with logical reasoning
  1. Logics are languages, reasoning is a process (may use logic)

Logic in general

- **Logics** are formal languages for representing information such that conclusions can be drawn
  Syntax defines the **sentences** in the language
  Semantics define the "**meaning**" of sentences;
  1. i.e., define **truth** of a sentence in a world
- E.g., the language of arithmetic
  1. $x+2 \geq y$ is a sentence; $x2+y > \{\}$ is not a sentence
  2. $x+2 \geq y$ is true iff the number $x+2$ is no less than the number y
  3. $x+2 \geq y$ is true in a world where $x = 7, y = 1$
  4. $x+2 \geq y$ is false in a world where $x = 0, y = 6$

Logic in general

Syntax and Semantics

- **Syntax**
  1. Rules for constructing legal sentences in the logic
  2. Which symbols we can use (English: letters, punctuation)
  3. How we are allowed to combine symbols
- **Semantics**
  1. How we interpret (read) sentences in the logic
  2. Assigns a meaning to each sentence

Logic in general

Syntax and Semantics

- **Example: "All lecturers are 170 centimeters tall"**
  1. **A valid sentence (syntax)**
  2. **And we can understand the meaning (semantics)**
  3. **This sentence happens to be false (there is a counter example)**

Logic

- Oldest form of knowledge representation in a computer is logic
- Logic is concerned with the truthfulness of a chain of statements.
- An argument is true if and only if, when all assumptions are true, then all conclusions are also true.

Logic

- **Two kinds of logic:**
  - **Propositional Logic**
  - **Predicate Calculus**
- **Both use symbols to represent knowledge and operators applied to the symbols to produce logical reasoning**

Propositional Logic

Propositional Logic (PL)

- Propositional logic represents and reasons with propositions.
- P.L. assigns symbolic variable to a proposition such as
  1. A = The car will start
- In. P. L. if we are concern with the truth of the statement, we will check the truth of A.

Propositional Logic (PL)

Operators	Symbol
AND	$\wedge$, &, $\cap$
OR	$\vee$, $\cup$, +
NOT	$\neg$, ~
IMPLIES	$\supset$, $\rightarrow$
EQUIVALENT	$\equiv$

Propositional Logic (PL)

- Propositions that are linked together with connectives, such as AND, OR, NOT, IMPLIES, and EQUIVALENT, are called compound statements.

- Example:

IF	The Students Work Hard	$\rightarrow$	A
AND	Always come to lectures	$\rightarrow$	B
AND	Do all their homework's	$\rightarrow$	C
THEN	they will get an A	$\rightarrow$	D

- Using the symbols: $A \wedge B \wedge C \rightarrow D$

Propositional Logic (PL)

Truth Table

A	B	A and B	A or B	Not A	A ≡ B
F	F	F	F	T	T
F	T	F	T	T	F
T	F	F	T	F	F
T	T	T	T	F	T

Propositional Logic (PL)

- Implies Operator: $C = A \rightarrow B$
- For implication C, if A is true, then B is implied to be true
- The implies return a F when A is TRUE and B is FALSE
  Otherwise it return a TRUE.

A	B	C
F	F	T
F	T	T
T	F	F
T	T	T

Ex : A = it is raining
B= I have an umbrella

Ex : A = program code
B= output

Propositional Logic (PL)

Idempotent Laws	$A \rightarrow B \equiv \neg A \cup B$ $A \cap \neg A \equiv F$ $A \cup \neg A \equiv T$
Commutative Laws	$A \cap B \equiv B \cap A$ $A \cup B \equiv B \cup A$
Distributive Laws	$A \cap (B \cup C) \equiv (A \cap B) \cup (A \cap C)$ $A \cup (B \cap C) \equiv (A \cup B) \cap (A \cup C)$
Associative Laws	$A \cap (B \cap C) \equiv (A \cap B) \cap C$ $A \cup (B \cup C) \equiv (A \cup B) \cup C$
Absorptive Laws	$A \cup (A \cap B) \equiv A$ $A \cap (A \cup B) \equiv A$
DeMorgan's Laws	$\neg(A \cap B) \equiv \neg A \cup \neg B$ $\neg(A \cup B) \equiv \neg A \cap \neg B$

Propositional logic: Syntax

- Propositional logic is the simplest logic
  The proposition symbols P_1, P_2 etc are sentences
  1. 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)

Propositional logic: Syntax

Sentence → AtomicSentence|ComplexSentence

AtomicSentence → True | False | Symbol

Symbol → P | Q | R |…

ComplexSentence → ¬Sentence

| (Sentence Λ Sentence)

| (Sentence V Sentence)

| (Sentence □ Sentence)

| (Sentence □ Sentence)

Figure A BNF (Backus-Naur Form) grammar of sentences in propositionalogic

from highest to lowest

¬ , Λ , V , □ , and □

Propositional logic: Semantics

Rules for evaluating truth with respect to a model *m*:

$\neg S$	is true iff	S is false		
$S_1 \wedge S_2$	is true iff	S_1 is true	and	S_2 is true
$S_1 \vee S_2$	is true iff	S_1 is true	or	S_2 is true
$S_1 \Rightarrow S_2$	is true iff	S_1 is false	or	S_2 is true
$S_1 \Leftrightarrow S_2$	is true iff	$S_1 \Rightarrow S_2$ is true	and	$S_2 \Rightarrow S_1$ is true

Propositional logic: Semantics

Each model specifies true/false for each proposition symbol

E.g. $P_{1,2}$ $P_{2,2}$ $P_{3,1}$

false true false

With these symbols, 8 possible models, can be enumerated automatically.

Simple recursive process evaluates an arbitrary sentence, e.g.,

$\neg P_{1,2} \wedge (P_{2,2} \vee P_{3,1}) = \textit{true} \wedge (\textit{true} \vee \textit{false}) = \textit{true} \wedge \textit{true} = \textit{true}$

Model, Valid, Satisfiable, Entailment

Model of a sentence {7.3}

- use the term model in place of "possible world."
- We say m is a model of a sentence α (or m satisfies α)

if α is true in m,

  - "x=2,y=2"________ a model of $x^2+y^2<=16$
  - "x=3,y=3"________ a model of $x^2+y^2<=16$
  - "P=T,Q=T"________ a model of P∧Q
  - "P=T,Q=F"________ a model of P∧Q

Model of a sentence {7.3}

- use the term model in place of "possible world."
- We say m is a model of a sentence α (or m satisfies α) if α is true in m,
  - "x=2,y=2"________ a model of $x^2+y^2<=16$
  - "x=3,y=3"________ a model of $x^2+y^2<=16$
  - "P=T,Q=T"________ a model of P∧Q
  - "P=T,Q=F"________ a model of P∧Q
- M(α) is the set of all models of α
  - M(P∨Q) = ____________________________________

Model of a sentence {7.3}

- use the term model in place of "possible world."
- We say m is a model of a sentence α (or m satisfies α) if α is true in m,
  - "x=2,y=2"________ a model of $x^2+y^2<=16$
  - "x=3,y=3"________ a model of $x^2+y^2<=16$
  - "P=T,Q=T"________ a model of P∧Q
  - "P=T,Q=F"________ a model of P∧Q
- M(α) is the set of all models of α
  - M(P∨Q) = {"P=T,Q=T","P=T,Q=F","P=F,Q=T"}

Satisfiability of a sentence{}

- A sentence is valid if it is true in all models,
  - $A \vee B, x \geq 0$ are____________________
  - $A \vee \neg A, x^2 \geq 0$ are____________________
- A sentence is satisfiable if it is true in some model;A sentence is unsatisfiable if it is true in no models
  - $A \wedge \neg A, X^2 < 0$ are____________________
  - $A \vee B, X > 0$ are____________________
  - α is satisfiable iff $M(\alpha)$ is____________________
  - α is unsatisfiable iff $M(\alpha)$ is____________________

Satisfiability of a sentence{}

- A sentence is valid if it is true in all models,
  - $A \vee B, x \geq 0$ are not valid
  - $A \vee \neg A, x^2 \geq 0$ are valid
- A sentence is satisfiable if it is true in some model;A sentence is unsatisfiable if it is true in no models
  - $A \wedge \neg A, X^2 < 0$ are unsatisfiable
  - $A \vee B, X > 0$ are satisfiable
  - α is satisfiable iff $M(\alpha)$ is not empty
  - α is unsatisfiable iff $M(\alpha)$ is empty

Model of a KB {7.3}

- A KB is a set of sentences
- A model m is a model of KB iff it is a model of all sentences in KB, that is, all sentences in KB are true in m.
- KB = {P,P ∨ R}

  M(KB)=______________________________________

Model of a KB {7.3}

- A KB is a set of sentences
- A model m is **a model of KB** iff it is a model of all sentences in KB, that is, all sentences in KB are true in m.
- KB = {P,P V R}
  M(KB)={"P=T,R=T","P=T,R=F"}

Satisfiability of a KB {7.3}

- A KB is **satisfiable** iff it admits at least one model (M(KB) is not empty); otherwise it is **unsatisfiable** (M(KB) is empty)
  - KB1 = {P,¬Q∧R} is_____________
  - KB2 = {¬P ∨ P} is_____________
  - KB3 = {P,¬P} is_____________

Satisfiability of a KB {7.3}

- A KB is satisfiable iff it admits at least one model (M(KB) is not empty); otherwise it is unsatisfiable (M(KB) is empty)
  - KB1 = {P,¬Q∧R} is satisfiable
  - KB2 = {¬P∨P} is satisfiable
  - KB3 = {P,¬P} is unsatisfiable

Entailment

- **Entailment means that one thing follows from another:**

$$KB \models \alpha$$

- **Knowledge base KB entails sentence α if and only if α is true in all worlds where KB is true**

  1. **E.g., the KB containing "the Giants won" and "the Reds won" entails "Either the Giants won or the Reds won"**
     E.g., x+y = 4 entails 4 = x+y
     Entailment is a relationship between sentences (i.e., syntax) that is based on semantics

- $x + y = 4 \models 4 = x + y$? ______yes/no
- $x^2 + y^2 <= 4 \models X^2 + y^2 <= 16$? ______yes/no
- $P \models P \wedge R$? ______yes/no

- $x + y = 4 \models 4 = x + y$? (yes)
- $x^2 + y^2 <= 4 \models X^2 + y^2 <= 16$? (yes)
- $P \models P \wedge R$? (no)

Models

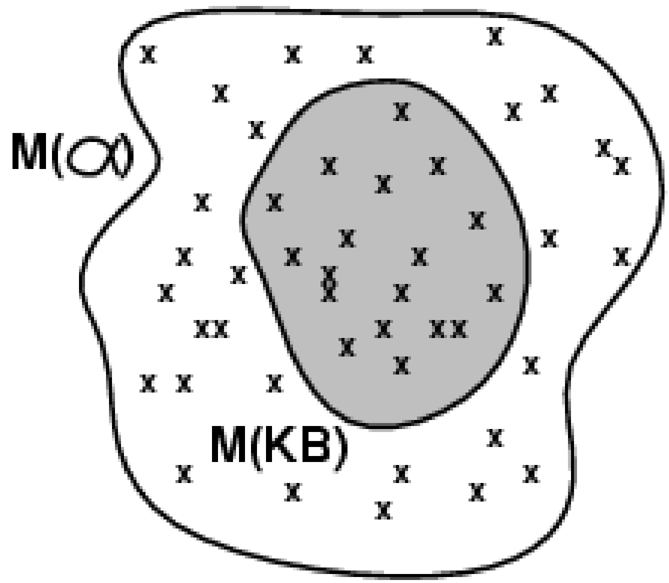
- **Logicians typically think in terms of models, which are formally structured worlds with respect to which truth can be evaluated**

- **We say *m* is a model of a sentence α if α is true in *m***

- ***M(α)* is the set of all models of α**

- **Then KB $\models$ α iff *M(KB)* $\subseteq$ *M(α)***

  1. **E.g. *KB* = Giants won and Reds won; α = Giants won**



Entailment

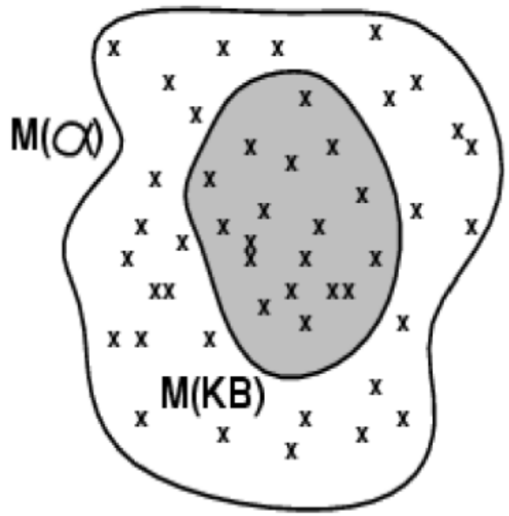
- KB $\models \alpha$
  - iff ...,
  - iff ...,
  - iff ...,
  - iff ...

Entailment

- KB $\models \alpha$
  - iff M(KB)__________M(α),
  - iff {KB,¬α}is___________,
  - iff KB=>α is__________,
  - iff KB,¬α $\models$____________

Entailment

- KB $\models$ α
  - iff M(KB) $\subseteq$ M(α),
  - iff {KB,¬α}is unsatisfiable,
  - iff KB=>α is valid,
  - iff KB,¬α $\models$ False

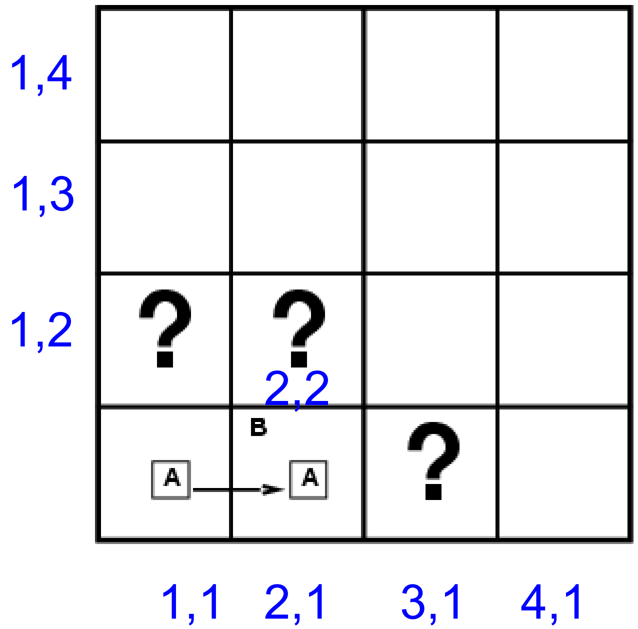


Entailment in the wumpus world

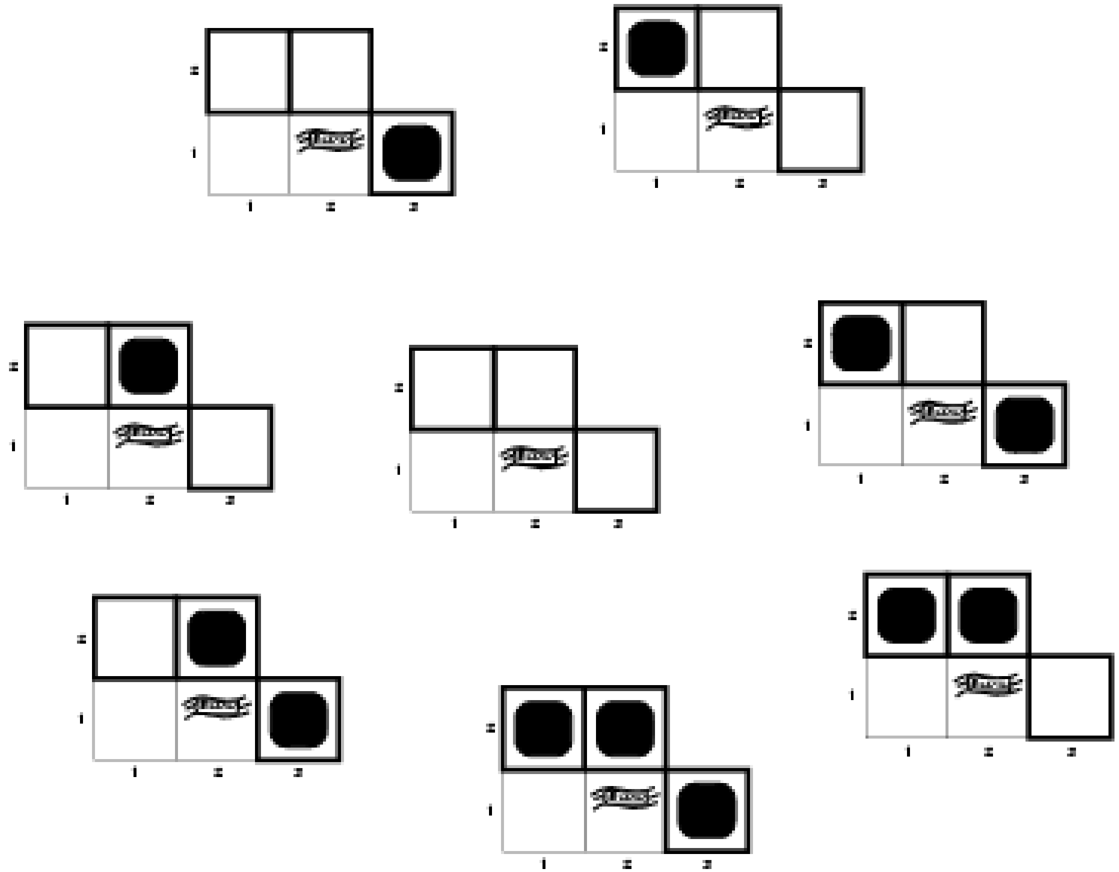
- Situation after detecting nothing in [1,1], moving right, breeze in [2,1]

- Consider possible models for *KB* assuming only pits

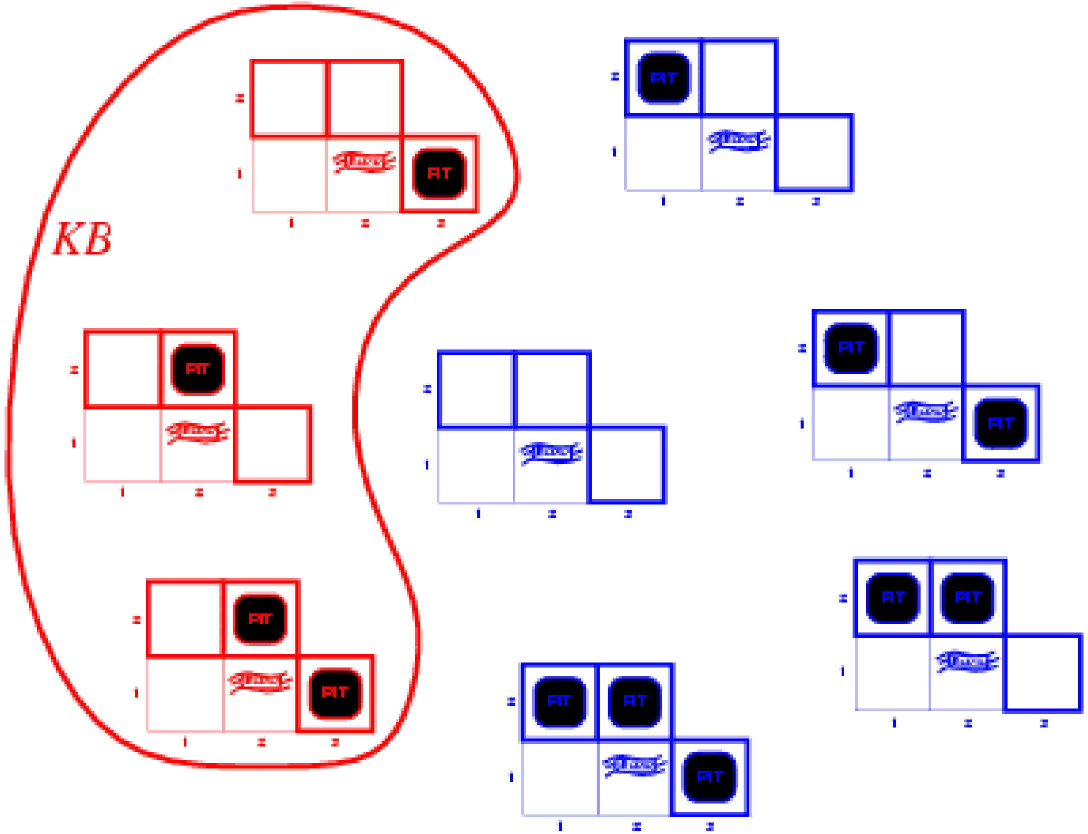
- Boolean choices ⇒ 8 possible models



Wumpus models

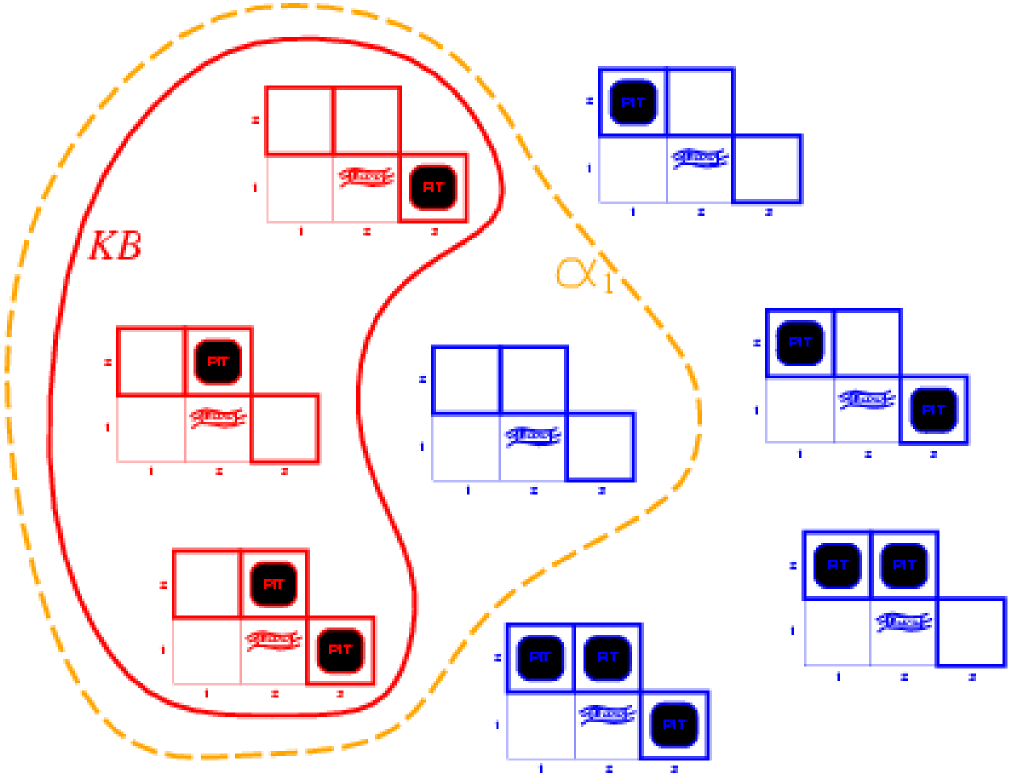


Wumpus models



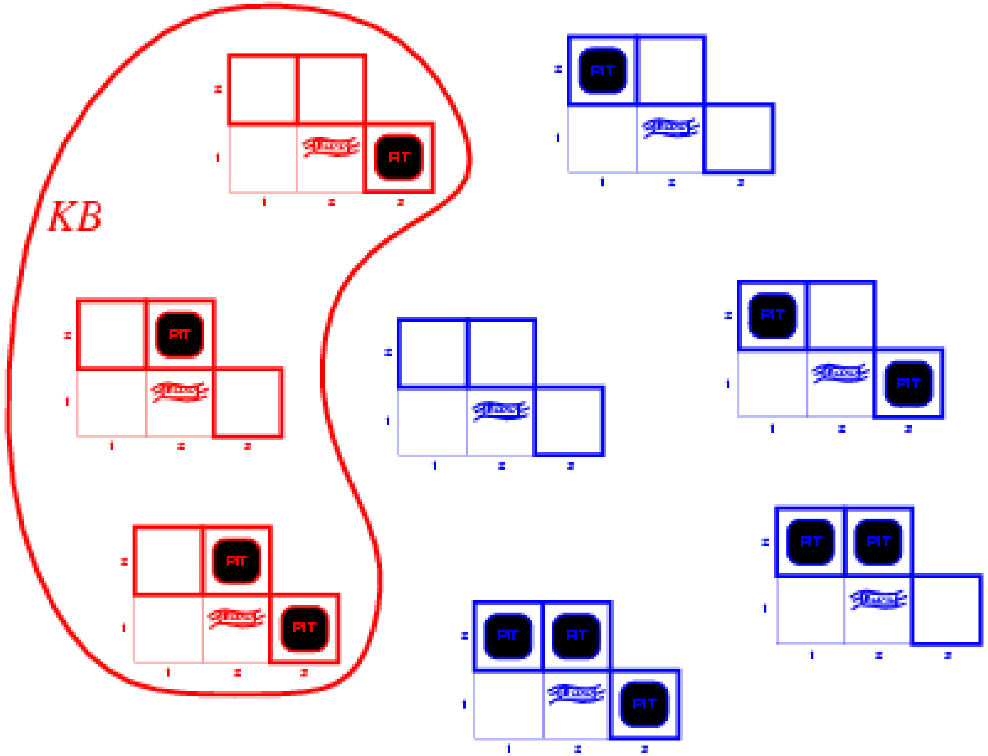
- *KB* = wumpus-world rules + observations

Wumpus models



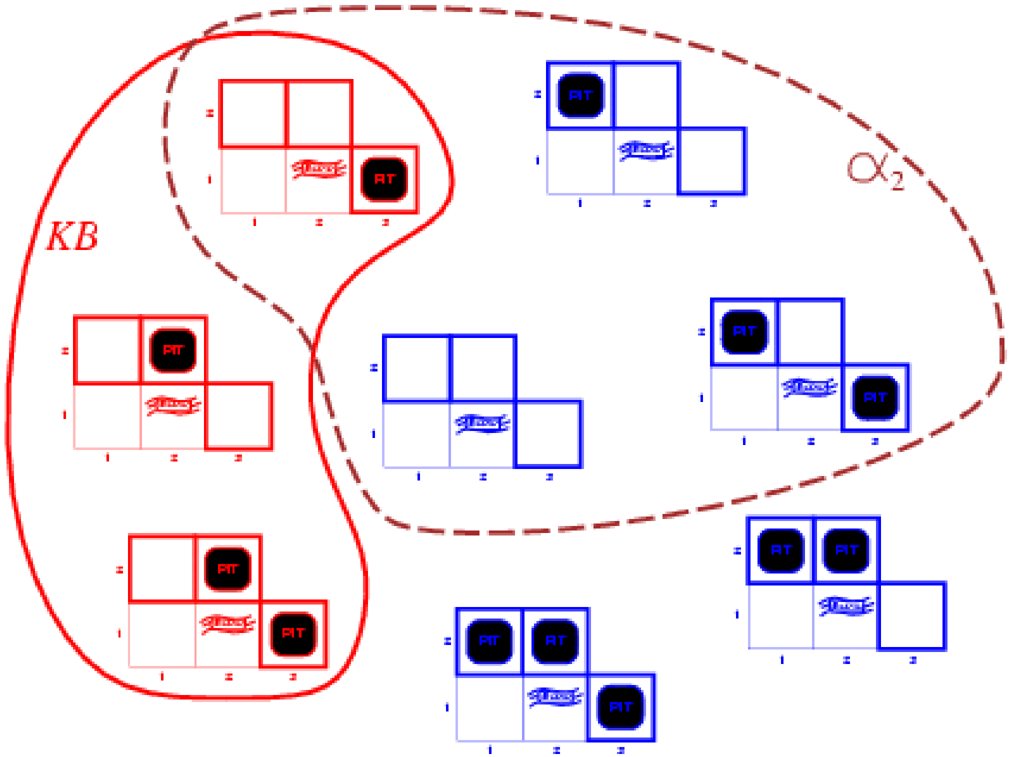
- *KB* = wumpus-world rules + observations
- α_1 = "[1,2] is safe", *KB* $\models \alpha_1$, proved by **model checking**

Wumpus models



- *KB* = wumpus-world rules + observations

Wumpus models



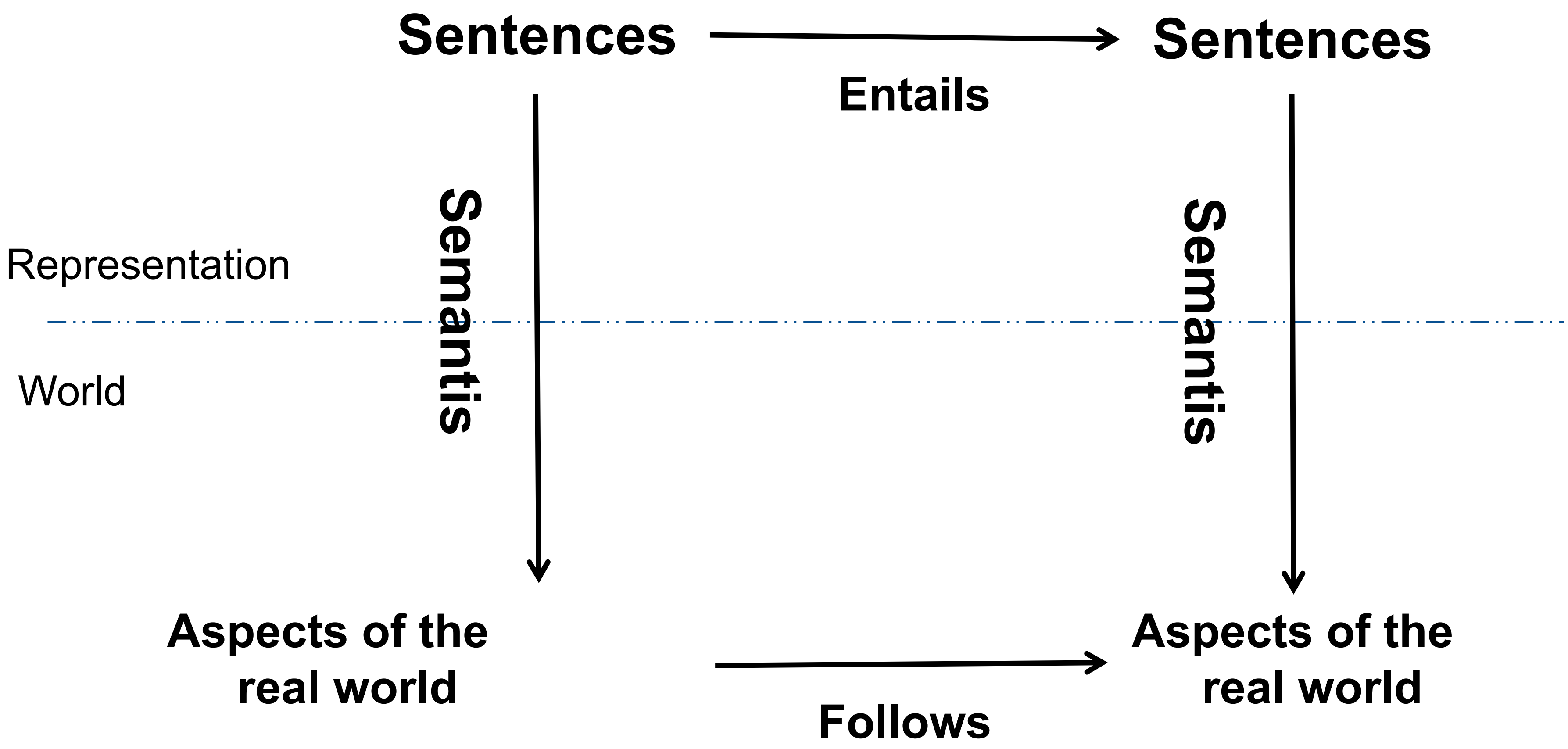
- *KB* = wumpus-world rules + observations
- α_2 = "[2,2] is safe", *KB* can not $\models \alpha_2$

Property of Inference Algorithm

- An inference algorithm that derives only entailed sentences is called **sound or truth-preserving**.
- An inference algorithm is **complete** if it can derive any sentence that is entailed.
- If KB is true in the real world, then any sentence **Alpha** derived from KB by a **sound** inference procedure is also true in the real world.

Property of Inference Algorithm

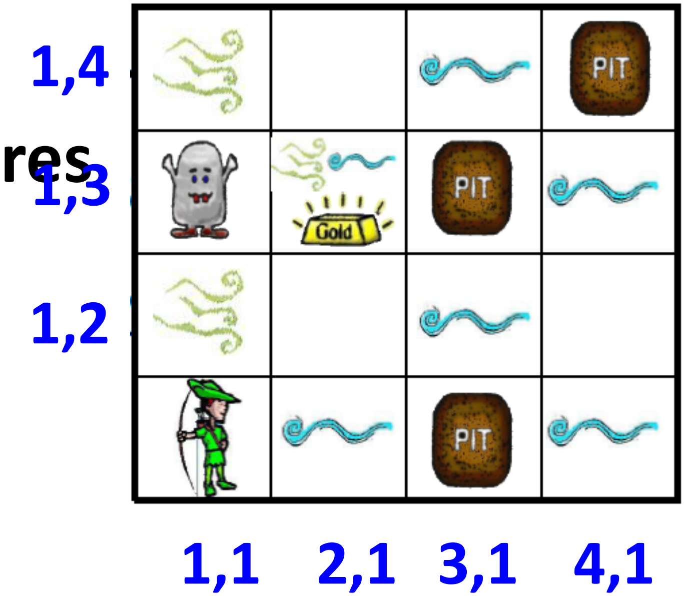
- Sensors and learning



How to prove Entailment

Wumpus world sentences

- how to represent wumpus world using PL?
- Let Pi,j be true if there is a pit in [i, j]
- Let Bi,j be true if there is a breeze in [i,j]
- Start
  - R1: ¬P1,1
- Pits cause breezes in adjacent squares
  - R2: B1,1 □ (P1,2 ∨ P2,1)
  - R3: B2,1 □ (P1,1 ∨ P2,2 ∨ P3,1)
- Perceiving
  - R4: ¬B1,1
  - R5: B2,1
- is P1,2 true????



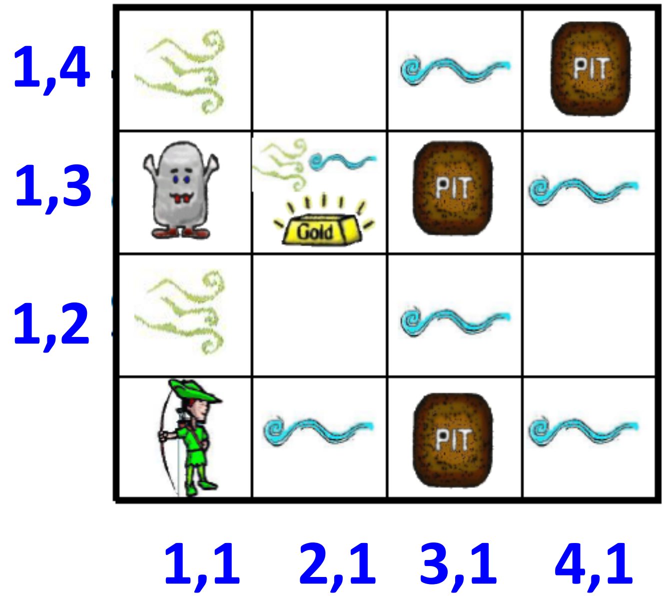
Dr.Yanmei Zheng

How to prove KB $\models \alpha$

Proof methods

- TELL(KB,R1)
- ...
- TELL(KB,R5)
- ASK(KB,¬P1,2)

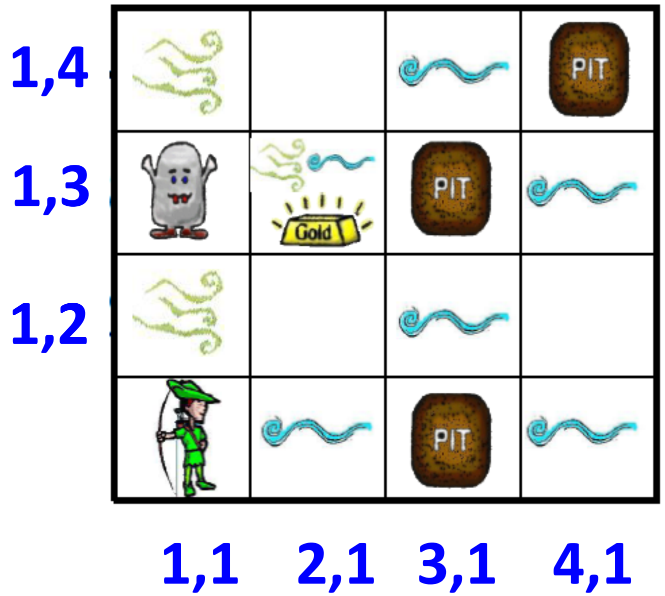
  KB $\models$ ¬P1,2 ?
- how to prove KB $\models$¬P1,2



Proof methods

- KB $\models \neg P1,2$?

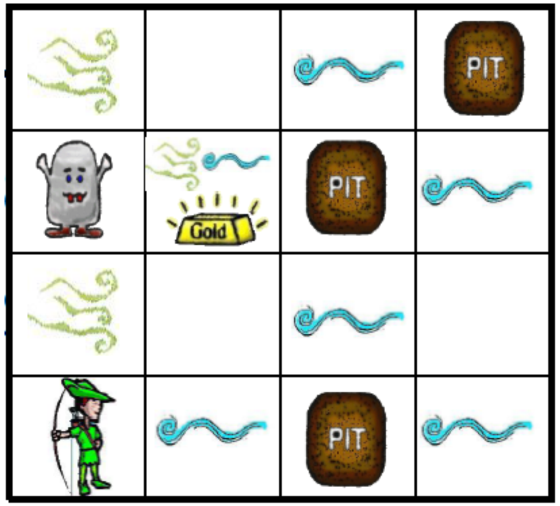
- KB $\models \alpha$ iff ...
- KB $\models \alpha$ iff ...
- KB $\models \alpha$ iff ...
- KB $\models \alpha$ iff ...



Proof methods

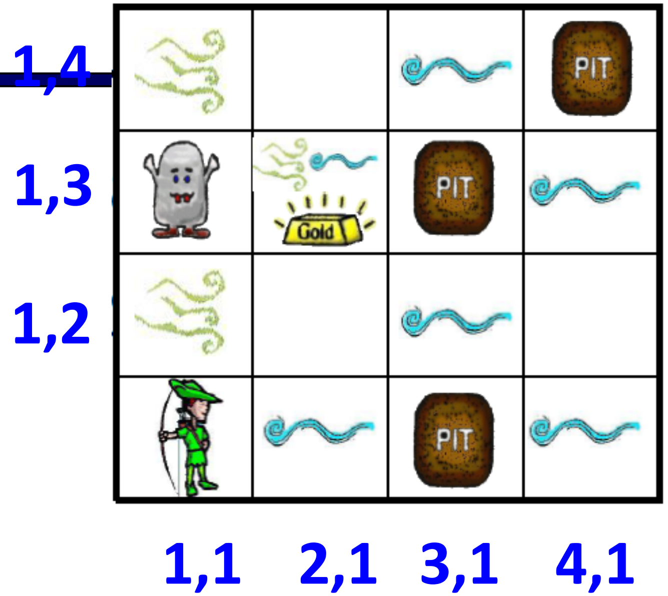
- KB ⊨¬P1,2?

- KB ⊨α iff M(KB)□ M(α)
- KB ⊨α iff (KB∧¬α) is unsatisfiable
- KB ⊨α iff KB=>α is valid
- KB ⊨α iff KB,¬α ⊨False



1,4
1,3
1,2
1,1 2,1 3,1 4,1

Proof methods



- KB ╞ ¬ P1,2?
- KB ╞α iff M(KB)□ M(α)
- KB ╞α iff (KB Λ α) is unsatisfiable
- KB ╞α iff KB=>α is valid
- KB ╞α iff KB,¬α ╞ False
- to prove KB ╞ α :
- for any m, if m is the model of KB, then m is also the model of α.
- for any m,KBΛα is false
- for any m,KB=>α is true.
- KB = >a is logically equivalent to True.
- From KB, derive α
- FromKB , ¬α, derive False

Model checking
truth table enumeration

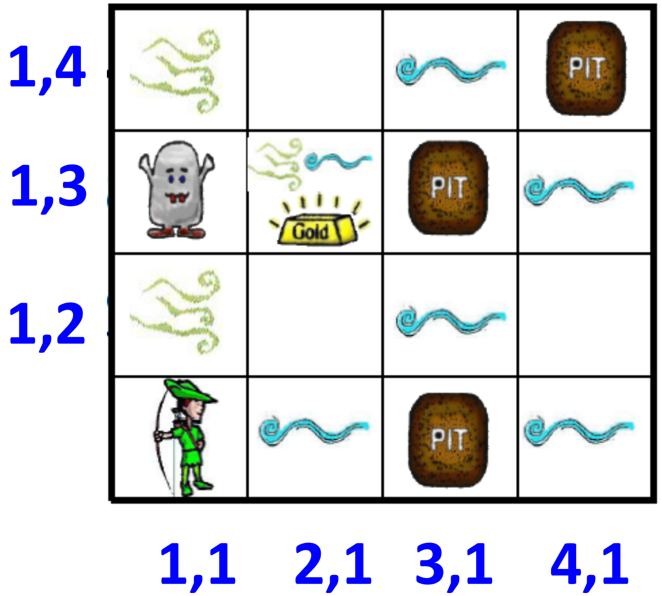
logical equivalence

inference rules

Proof methods

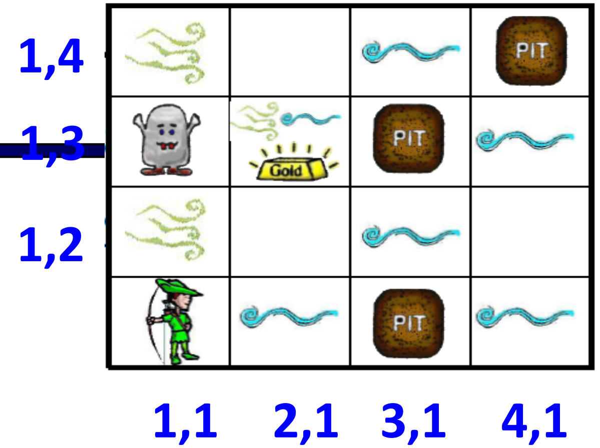
- $KB \models \neg P_{1,2}$?

- $KB \models \alpha$ iff $M(KB)$ □ $M(\alpha)$
- $KB \models \alpha$ iff $(KB \wedge \alpha)$ is unsatisfiable
- $KB \models \alpha$ iff $KB \Rightarrow \alpha$ is valid
- $KB \models \alpha$ iff $KB, \neg\alpha \models$ False



- proof by:
  - Model checking
    - truth table enumeration(always exponential in n)
  - logical equivalence
  - inference rules

proof by model checking{7.4.4}



- KB ╞ ¬P1,2 ?

B1,1	B2,1	P1,1	P1,2	P2,1	P2,2	P3,1	R1	R2	R3	R4	R5	KB
false false . false	false false . false	false false . false	false false . false	false false . false	false false . false	false true . false	true true . true	true true . true	true false . false	true true . true	false false . true	false false . false
false false false	true true true	false false false	false false false	false false false	false true false	true false true	true true true	true true true	true true true	true true true	true true true	true true true
false . true	true . true	false . true	false . true	true . true	false . true	false . true	true . false	false . true	false . true	true . false	true . true	false . false

Proof by model checking

- (A ∨ ¬B) ∧(B ∨ ¬C) □ A ∨ ¬C is valid?

A	B	C	A∨¬B	B∨¬C	A∨¬C	(A∨¬B) ∧(B∨¬C) □ A∨¬C
0	0	0	1	1	1	1
0	0	1	1	0	0	1
0	1	0	0	1	1	1
0	1	1	0	1	0	1
1	0	0	1	1	1	1
1	0	1	1	0	1	1
1	1	0	1	1	1	1
1	1	1	1	1	1	1

Logical equivalence{7.5}

- Logical equivalence: Two sentences are logically **equivalent** iff they are true in same models

$(\alpha \land \beta) \equiv (\beta \land \alpha)$ commutativity of $\land$

$(\alpha \lor \beta) \equiv (\beta \lor \alpha)$ commutativity of $\lor$

$((\alpha \land \beta) \land \gamma) \equiv (\alpha \land (\beta \land \gamma))$ associativity of $\land$

$((\alpha \lor \beta) \lor \gamma) \equiv (\alpha \lor (\beta \lor \gamma))$ associativity of $\lor$

$\neg(\neg\alpha) \equiv \alpha$ double-negation elimination

$(\alpha \Rightarrow \beta) \equiv (\neg\beta \Rightarrow \neg\alpha)$ contraposition

$(\alpha \Rightarrow \beta) \equiv (\neg\alpha \lor \beta)$ implication elimination

$(\alpha \Leftrightarrow \beta) \equiv ((\alpha \Rightarrow \beta) \land (\beta \Rightarrow \alpha))$ biconditional elimination

$\neg(\alpha \land \beta) \equiv (\neg\alpha \lor \neg\beta)$ de Morgan

$\neg(\alpha \lor \beta) \equiv (\neg\alpha \land \neg\beta)$ de Morgan

$(\alpha \land (\beta \lor \gamma)) \equiv (\alpha \land \beta) \lor (\alpha \land \gamma))$ distributivity of $\land$ over $\lor$

$(\alpha \lor (\beta \land \gamma)) \equiv (\alpha \lor \beta) \land (\alpha \lor \gamma)$ distributivity of $\lor$ over $\land$

Validity and satisfiability

A sentence is valid if it is true in all models,

e.g., *True*, $A \vee \neg A$, $A \Rightarrow A$, $(A \wedge (A \Rightarrow B)) \Rightarrow B$

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., $A \vee B$, C

A sentence is unsatisfiable if it is true in no 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

$\alpha \models \beta$ if and only if the sentence $(\alpha \wedge \neg \beta)$ is unsatisfiable

poof by logical equivalence:

- $(A \vee \neg B) \wedge (B \vee \neg C) \Rightarrow A \vee \neg C$ is valid ?

$$(A \vee \neg B) \wedge (B \vee \neg C) \Rightarrow A \vee \neg C$$

$$\equiv \neg((A \vee \neg B)) \wedge (B \vee \neg C) \vee A \vee \neg C$$

$$\equiv (\neg(A \vee \neg B) \vee \neg(B \vee \neg C)) \vee A \vee \neg C$$

$$\equiv (\neg A \wedge B) \vee (\neg B \wedge C) \vee A \vee \neg C$$

$$\equiv (\neg A \wedge B) \vee A \vee (\neg B \wedge C) \vee \neg C$$

$$\equiv ((\neg A \vee A) \wedge (B \vee A)) \vee ((\neg B \vee \neg C) \wedge (C \vee \neg C))$$

$$\equiv (B \vee A) \vee (\neg B \vee \neg C)$$

$$\equiv T$$

Proof methods

- Proof methods divide into (roughly) two kinds:

1. **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
     Typically require transformation of sentences into a normal form
2. **Model checking**
   - truth table enumeration (always exponential in *n*)
     improved backtracking, e.g., Davis--Putnam-Logemann-Loveland (DPLL)
     Heuristic search in model space (sound but incomplete)
       e.g., min-conflicts-like hill-climbing algorithms

Inference Rules

- **Modus Ponens**

$$\frac{\alpha \;=>\; \beta,\;\; \alpha}{\beta}$$

- **And-Elimination**

$$\frac{\alpha \Lambda \beta}{\alpha}$$

- **logical equivalence**

  α□ β≡(α□ β)Λ(β□ α)

$$\frac{\alpha <=> \beta}{(\alpha => \beta)\Lambda(\beta => \alpha)}$$

$$\frac{(\alpha => \beta)\Lambda(\beta => \alpha)}{\alpha <=> \beta}$$

Inference

- **Example:**
- “Gary is either intelligent or a good actor.
- If Gary is intelligent, then he can count from 1 to 10.
- Gary can only count from 1 to 2.
- Therefore, Gary is a good actor.”
- **Propositions:**
- I: “Gary is intelligent.”
- A: “Gary is a good actor.”
- C: “Gary can count from 1 to 10.”

Inference

- I: "Gary is intelligent."

A: "Gary is a good actor."

C: "Gary can count from 1 to 10."

- Step 1: $\neg C$ Hypothesis
- Step 2: $I \supset C$ Hypothesis
- Step 3: $\neg I$ Modus Tollens Steps 1 & 2
- Step 4: $A \vee I$ Hypothesis
- Step 5: A Disjunctive Syllogism Steps 3 & 4

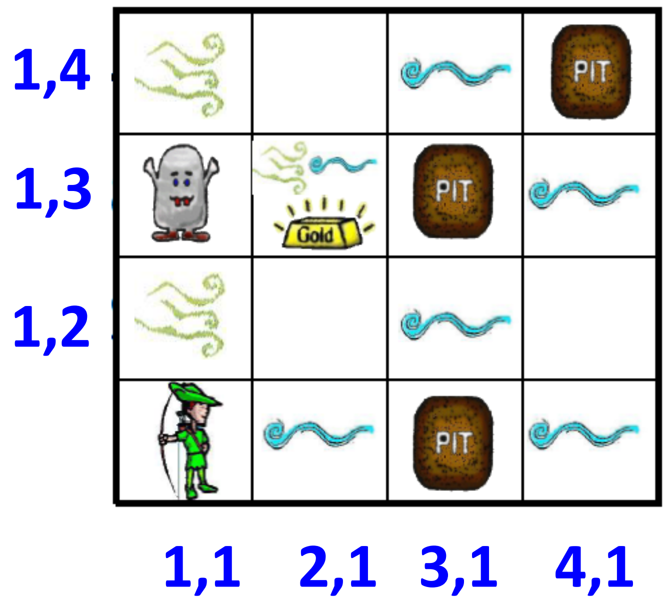
- Conclusion: A ("Gary is a good actor.")

Inference{7.5.1}

The preceding derivation a sequence of applications of inference rules is called a proof.

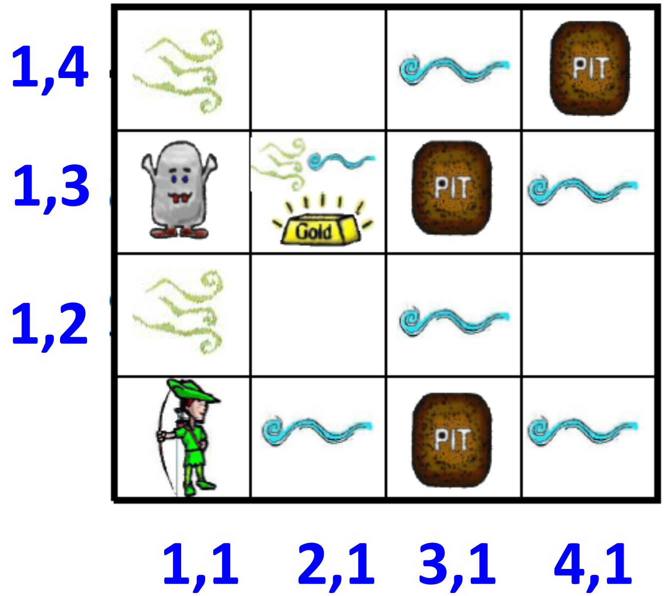
Finding proofs is exactly like finding solutions to search problems.

Searching for proofs is an alternative to enumerating models.



- Start inference[7.5.1)
  - R1: ¬ P1,1

- Pits cause breezes in adjacent squares
  - R2: B1,1 <=> (P1,2VP2,1)
  - R3: B2,1 <=>(P1,1VP2,2V P3,1)
- Perceiving
  - R4:¬B1,1
  - R5: B2,1
- KB ⊨¬P1,2?
- R6:(B1,1 => (P1,2VP2,1))Λ((P1,2VP2,1)=> B1,1) R2, biconditional-elimination
- R7:((P1,2VP2,1)=> B1,1) R6, And- Elimination
- R8:(¬B1,1=>¬(P1,2vP2,1)) R7, contraposition
- R9: ¬(P1,2VP2,1) R8,R4 , Modus-ponens
- R10:¬P1,2ΛP2,1 R9, De Morgan
- R11: ¬P1,2 R10, And- Elimination



Inference{7.5.1}

- Car problem:
  (1) Batter-OK、
  (2) ¬Empty-Gas-Tank、
  (3) ¬Car-OK
  (4) Battery-OK∧¬Empty-Gas-Tank Engine-Starts
  (5) Engine-Starts∧¬Flat-Tire Car-OK
- What's the problem?
  - ______________________

Inference{7.5.1}

- Car problem:

  (1) Batter-OK、

  (2) ¬Empty-Gas-Tank、

  (3) ¬Car-OK

  (4) Battery-OK∧¬Empty-Gas-Tank Engine-Starts

  (5) Engine-Starts∧¬Flat-Tire Car-OK

- What's the problem?
  - Flat-Tire

Formalize the inference as a search problem

inference by searching{7.5.1}

- apply a search algorithm
  - Initial state:____________________
  - Actions:______________________
  - Result:_______________________
  - Goal:_________________________________

inference by searching{7.5.1}

- apply a search algorithm
  - Initial state: the initial KB
  - Actions: inference rules
  - Result: add inferred sentences to KB
  - Goal: the sentence we are trying to prove is in the KB.

from KB. derive α

from [KB,- -a), derive False

[KB, a) is unsatisfiable

using resolution inference rule

Thank you

End of Chapter 7-1