



# Logical Agents: Knowledge Bases and the Wumpus World

R&N § 7.1-7.5

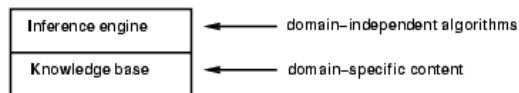
Jacques Fleuriot

School of informatics  
University of Edinburgh

Informatics 2D



## Knowledge bases



- Knowledge base = set of sentences in a formal language
- Declarative approach to building an agent (or other system):
  - 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
  - i.e., data structures in KB and algorithms that manipulate them

Informatics 2D



## Outline

- Knowledge-based agents
- Wumpus world
- Logic in general - models and entailment
- Propositional (Boolean) logic
- Equivalence, validity, satisfiability

Informatics 2D



## A simple knowledge-based agent

```

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 ← ASK(KB, MAKE-ACTION-QUERY(t))
  TELL(KB, MAKE-ACTION-SENTENCE(action, t))
  t ← t + 1
  return action
  
```

- The agent must be able to:
  - represent states, actions, etc.
  - incorporate new percepts
  - update internal representations of the world
  - deduce hidden properties of the world
  - deduce appropriate actions

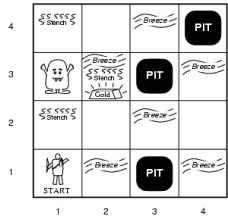
Informatics 2D



# Wumpus World PEAS description



- **Performance measure**
  - gold +1000, death -1000
  - -1 per step, -10 for using the arrow
- **Environment**
  - Squares adjacent to wumpus are smelly
  - Squares adjacent to pits are breezy
  - 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
- **Actuators:** Left turn, Right turn, Forward, Grab, Release, Shoot
- **Sensors:** Stench, Breeze, Glitter, Bump, Scream



Informatics 2D



# Wumpus world characterization

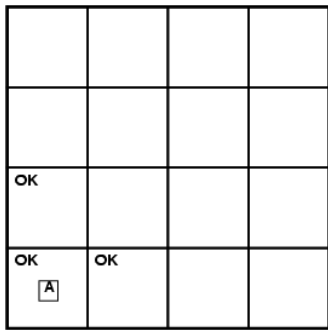


- **Fully Observable?** No – only local perception
- **Deterministic?** Yes – outcomes exactly specified
- **Episodic?** No – sequential at the level of actions
- **Static?** Yes – Wumpus and Pits do not move
- **Discrete?** Yes
- **Single-agent?** Yes – Wumpus is essentially a natural feature

Informatics 2D



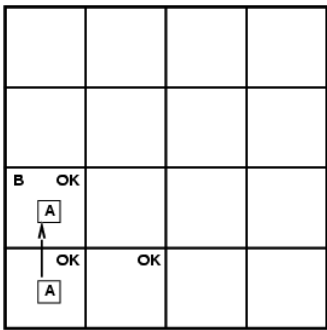
# Exploring a wumpus world



Informatics 2D



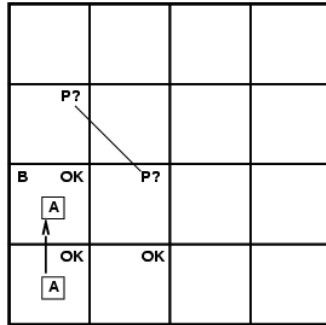
# Exploring a wumpus world



Informatics 2D



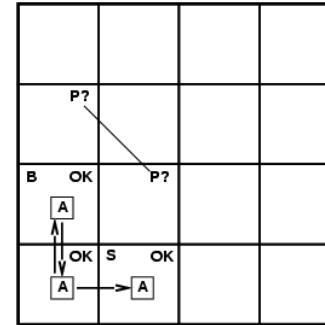
# Exploring a wumpus world



Informatics 2D



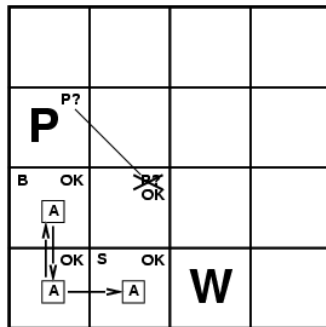
# Exploring a wumpus world



Informatics 2D



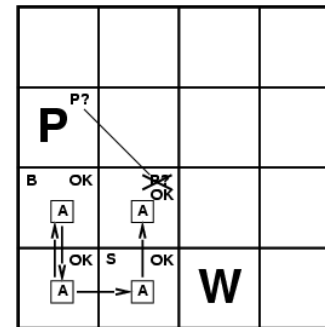
# Exploring a wumpus world



Informatics 2D



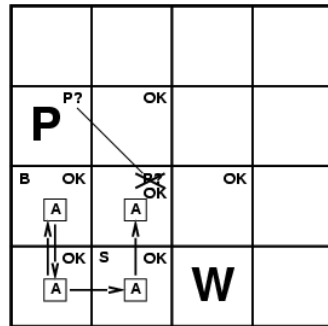
# Exploring a wumpus world



Informatics 2D



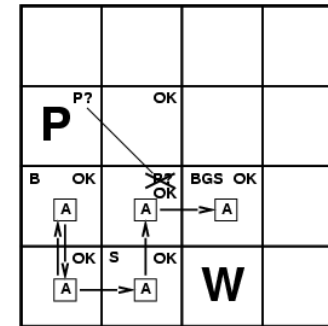
## Exploring a wumpus world



Informatics 2D



## Exploring a wumpus world



Informatics 2D



## Logic in general



- **Logics** are formal languages for representing information such that conclusions can be drawn
- **Syntax** defines the sentences in the language
- **Semantics** defines the "meaning" of sentences;
  - i.e., define truth of a sentence in a world
- E.g., the language of arithmetic
  - $x+2 \geq y$  is a sentence;  $x2+y > \{$  is not a sentence
  - $x+2 \geq y$  is true iff 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  $x = 0, y = 6$

Informatics 2D



## Entailment



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

$$KB \models \alpha$$

- Knowledge base  $KB$  entails sentence  $\alpha$  **if and only if**  $\alpha$  is true in all worlds where  $KB$  is true
  - e.g., the KB containing "Celtic won" and "Hearts won" entails "Either Celtic won or Hearts won"
  - e.g.,  $x+y = 4$  entails  $4 = x+y$
  - Entailment is a relationship between sentences (i.e., syntax) that is based on semantics

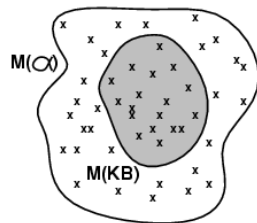
Informatics 2D





# 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  $\alpha$  if  $\alpha$  is true in  $m$
- $M(\alpha)$  is the set of all models of  $\alpha$
- Then  $KB \models \alpha$  iff  $M(KB) \subseteq M(\alpha)$
- The *stronger* an assertion, the fewer models it has.



Informatics 2D



# Entailment in the wumpus world

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

?	?		
A	B	?	

Consider possible models for  $KB$  assuming **only** pits

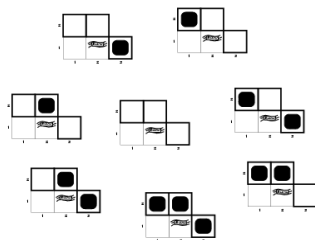
**3 Boolean choices  $\Rightarrow$  8 possible models**

**Mid-lecture Exercise: What are these 8 models?**

Informatics 2D



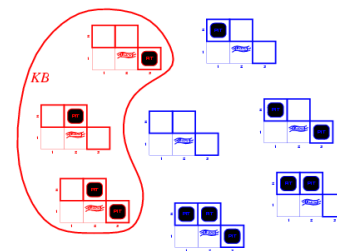
# Wumpus models



Informatics 2D



# Wumpus models

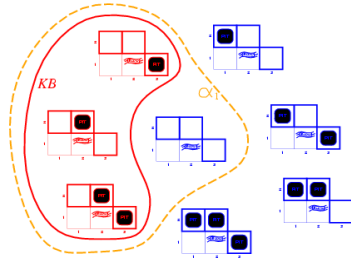


- $KB =$  wumpus-world rules + observations

Informatics 2D



## Wumpus models

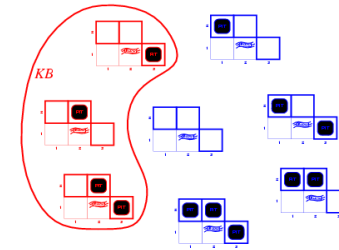


- $KB$  = wumpus-world rules + observations
- $\alpha_1 = "[1,2]$  has no pit",  $KB \models \alpha_1$ , proved by model checking
  - In every model in which  $KB$  is true,  $\alpha_1$  is also true

Informatics 2D



## Wumpus models

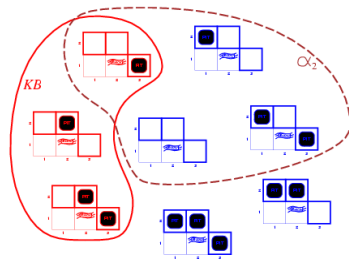


- $KB$  = wumpus-world rules + observations

Informatics 2D



## Wumpus models



- $KB$  = wumpus-world rules + observations
- $\alpha_2 = "[2,2]$  has no pit",  $KB \not\models \alpha_2$ 
  - In some models in which  $KB$  is true,  $\alpha_2$  is false

Informatics 2D



## Inference



- $KB \vdash_i \alpha$  = sentence  $\alpha$  can be **derived** from  $KB$  by **procedure**  $i$
- **Soundness**:  $i$  is sound if whenever  $KB \vdash_i \alpha$ , it is also true that  $KB \models \alpha$
- **Completeness**:  $i$  is complete if whenever  $KB \models \alpha$ , it is also true that  $KB \vdash_i \alpha$
- **Preview**: we will define first-order logic:
  - expressive enough to say almost anything of interest,
  - sound and complete inference procedure exists.
  - But first...

Informatics 2D



# Propositional logic: Syntax



Propositional logic is the simplest logic – illustrates basic ideas:

- The proposition symbols  $P_1, P_2$  etc are sentences
- If  $S$  is a sentence,  $\neg S$  is a sentence (**negation**)
- If  $S_1$  and  $S_2$  are sentences,  $S_1 \wedge S_2$  is a sentence (**conjunction**)
- If  $S_1$  and  $S_2$  are sentences,  $S_1 \vee S_2$  is a sentence (**disjunction**)
- If  $S_1$  and  $S_2$  are sentences,  $S_1 \Rightarrow S_2$  is a sentence (**implication**)
- If  $S_1$  and  $S_2$  are sentences,  $S_1 \Leftrightarrow S_2$  is a sentence (**biconditional**)

Informatics 2D



# 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. 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  
 i.e., is false iff  $S_1$  is true and  $S_2$  is false  
 $S_1 \Leftrightarrow S_2$  is true iff  $S_1 \Rightarrow S_2$  is true and  $S_2 \Rightarrow S_1$  is true

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

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

Informatics 2D



# Truth tables 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

Informatics 2D



# Wumpus world sentences



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

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

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

- “Pits cause breezes in adjacent squares”

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

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

Informatics 2D



## Truth tables for inference



$B_{1,1}$	$B_{2,1}$	$P_{1,1}$	$P_{1,2}$	$P_{2,1}$	$P_{2,2}$	$P_{3,1}$	$KB$	$\alpha_1$
false	false	false	false	false	false	false	false	true
false	false	false	false	false	false	true	false	true
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
false	true	false	false	false	false	false	false	true
false	true	false	false	false	false	true	true	true
false	true	false	false	false	true	false	true	true
false	true	false	false	false	true	true	true	true
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
true	true	true	true	true	true	true	false	false

Informatics 2D



## Logical equivalence



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

- $(\alpha \wedge \beta) \equiv (\beta \wedge \alpha)$  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  $\vee$
- $\neg(\neg\alpha) \equiv \alpha$  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  $\wedge$

Informatics 2D



## Inference by enumeration



- Depth-first enumeration of all models is sound and complete

```

function TT-ENTAILS?(KB,  $\alpha$ ) returns true or false
  symbols  $\leftarrow$  a list of the proposition symbols in KB and  $\alpha$ 
  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
  else do
    P  $\leftarrow$  FIRST(symbols); rest  $\leftarrow$  REST(symbols)
    return TT-CHECK-ALL(KB,  $\alpha$ , rest, EXTEND(P, true, model) and
      TT-CHECK-ALL(KB,  $\alpha$ , rest, EXTEND(P, false, model))
    
```

- PL-TRUE? returns true if a sentence holds within a model
- EXTEND( $P, val, model$ ) returns a new partial model in which  $P$  has value  $val$
- For  $n$  symbols, time complexity is  $O(2^n)$ , space complexity is  $O(n)$

Informatics 2D



## 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

Informatics 2D





## Proof methods

- Proof methods divide into (roughly) two kinds:
  - 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
    - Example: resolution
  - Model checking
    - truth table enumeration (always exponential in  $n$ )
    - improved backtracking, e.g., Davis-Putnam-Logemann-Loveland (DPLL) method
    - heuristic search in model space (sound but incomplete)  
e.g., min-conflicts-like hill-climbing algorithms

Informatics 2D



## Summary

- Logical agents apply inference to a knowledge base to derive new information and make decisions
- Basic concepts of logic:
  - syntax: formal structure of sentences
  - semantics: truth of sentences wrt models
  - entailment: necessary truth of one sentence given another
  - inference: deriving sentences from other sentences
  - soundness: derivations produce only entailed sentences
  - completeness: derivations can produce all entailed sentences
- Wumpus world requires the ability to represent partial and negated information, reason by cases, etc.
- Propositional logic lacks expressive power

Informatics 2D

