Propositional Logic: Logical Agents (Part I)

You will be expected to know:

- Basic definitions (section 7.1, 7.3)
- Models and entailment (7.3)
- Syntax, logical connectives (7.4.1)
- Semantics (7.4.2)
- Simple inference (7.4.4)

Complete architectures for intelligence?

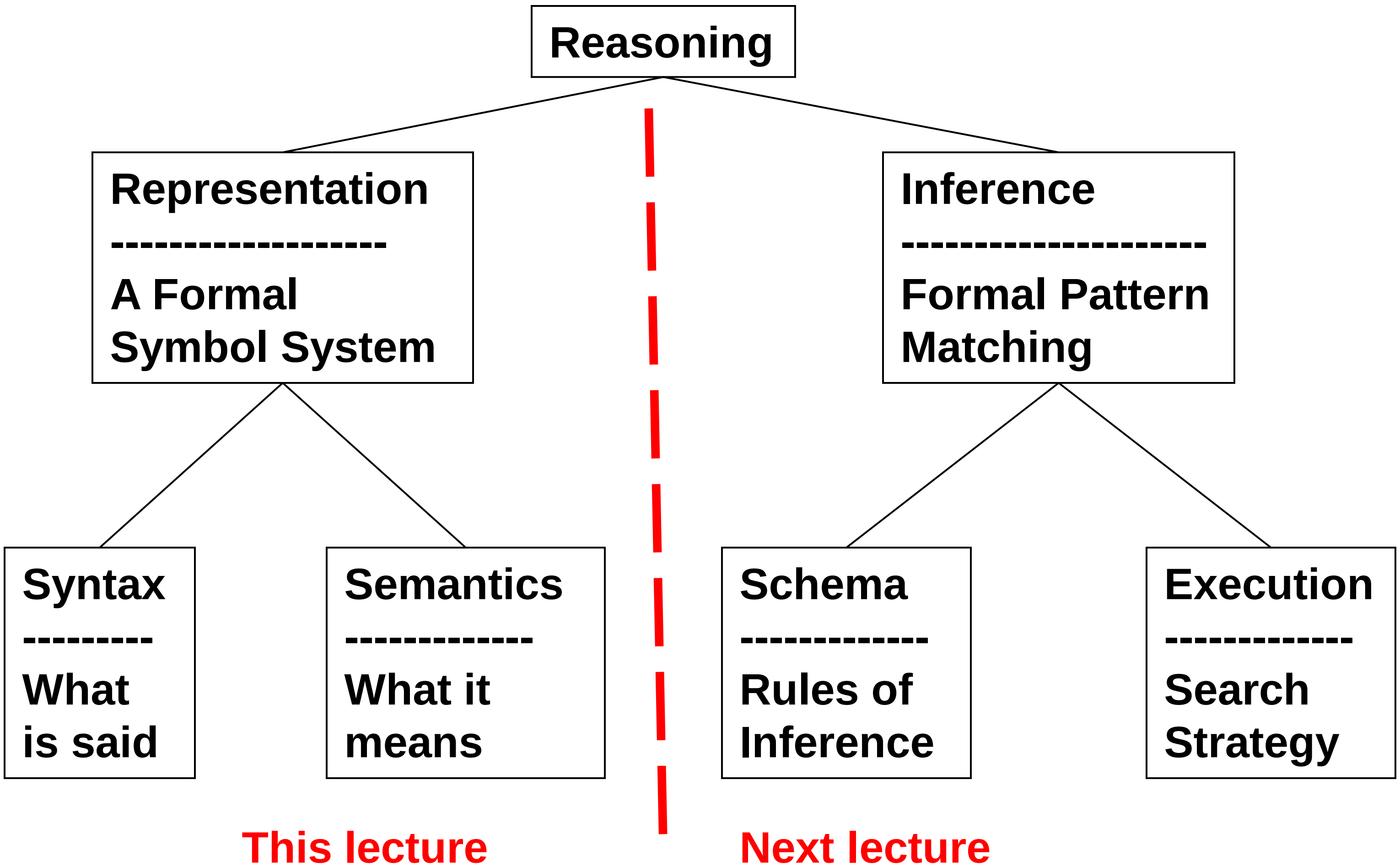
- <u>Search?</u>
  - Solve the problem of what to do.
- <u>Logic and inference?</u>
  - Reason about what to do.
  - Encoded knowledge/“expert” systems?
    - Know what to do.
- <u>Learning?</u>
  - Learn what to do.
- <u>Modern view: It’s complex & multi-faceted.</u>

Inference in Formal Symbol Systems: Ontology, Representation, Inference

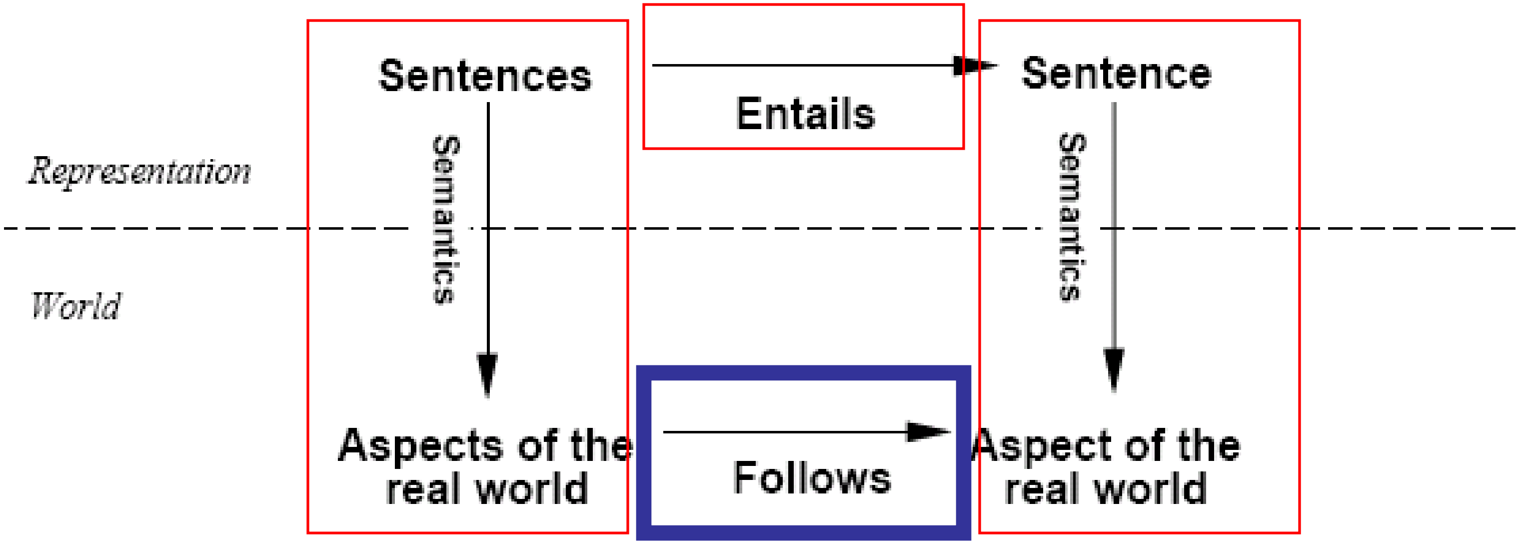
- **Formal Symbol Systems**
  - **Symbols** correspond to **things/ideas** in the world
  - **Pattern matching & rewrite** corresponds to **inference**

- **Ontology:** What exists in the world?
  - What must be represented?
- **Representation:** Syntax vs. Semantics
  - What’s Said vs. What’s Meant
- **Inference:** Schema vs. Mechanism
  - Proof Steps vs. Search Strategy

Ontology:
What kind of things exist in the world?
What do we need to describe and reason about?



Schematic perspective



If KB is true in the real world,

then any sentence α entailed by KB

is also true in the real world.

For example: If I tell you (1) Sue is Mary's sister, and (2) Sue is Amy's mother, then it **necessarily follows in the world** that Mary is Amy's aunt, even though I told you nothing at all about aunts. This sort of reasoning pattern is what we hope to capture.

Why Do We Need Logic?

- Problem-solving agents were very inflexible: hard code every possible state.
- Search is almost always exponential in the number of states.
- Problem solving agents cannot infer unobserved information.
- We want an algorithm that reasons in a way that resembles reasoning in humans.

Knowledge-Based Agents

- **KB = knowledge base**
  - A set of sentences or facts
  - e.g., a set of statements in a logic language

- **Inference**
  - Deriving new sentences from old
  - e.g., using a set of logical statements to infer new ones

- **A simple model for reasoning**
  - Agent is told or perceives new evidence
    - E.g., agent is told or perceives that A is true
  - Agent then infers new facts to add to the KB
    - E.g., KB = { (A -> (B OR C)); (not C) }

      then given A and not C the agent can infer that B is true
    - B is now added to the KB even though it was not explicitly asserted, i.e., the agent inferred B

Types of Logics

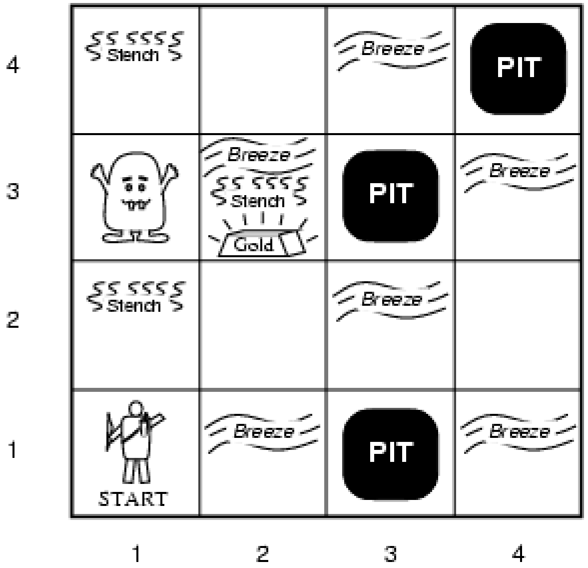
- **Propositional logic:** concrete statements that are either true or false
  - E.g., John is married to Sue.
- **Predicate logic (also called first order logic, first order predicate calculus):** allows statements to contain variables, functions, and quantifiers
  - For all X, Y: If X is married to Y then Y is married to X.
- **Probability:** statements that are possibly true; the chance I win the lottery?
- **Fuzzy logic:** vague statements; paint is <u>slightly grey</u>; sky is <u>very cloudy</u>.
- **Modal logic** is a class of various logics that introduce modalities:
  - **Temporal logic:** statements about time; John was a student at UCI for <u>four years</u>, and <u>before that</u> he spent <u>six years</u> in the US Marine Corps.
  - **Belief and knowledge:** Mary <u>knows</u> that John is married to Sue; a poker player <u>believes</u> that another player will fold upon a large bluff.
  - **Possibility and Necessity:** What <u>might</u> happen (possibility) and <u>must</u> happen (necessity); I <u>might</u> go to the movies; I <u>must</u> die and pay taxes.
  - **Obligation and Permission:** It is <u>obligatory</u> that students study for their tests; it is <u>permissible</u> that I go fishing when I am on vacation.

Other Reasoning Systems

- How to produce new facts from old facts?
- **Induction**
  - Reason from facts to the general law
  - Scientific reasoning, machine learning
- **Abduction**
  - Reason from facts to the best explanation
  - Medical diagnosis, hardware debugging
- **Analogy (and metaphor, simile)**
  - Reason that a new situation is like an old one

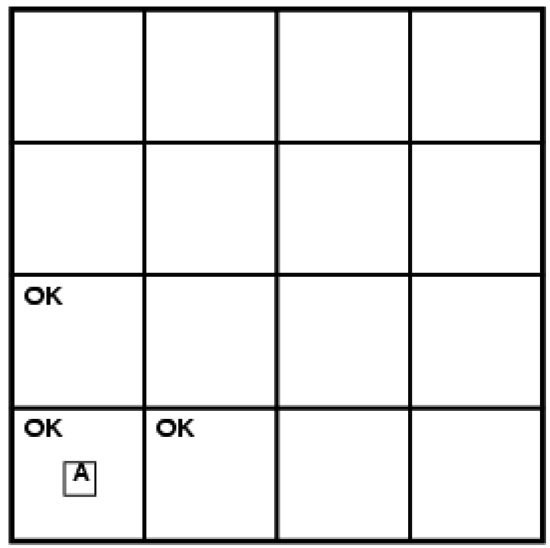
Wumpus World PEAS description

Would DFS work well? A*?

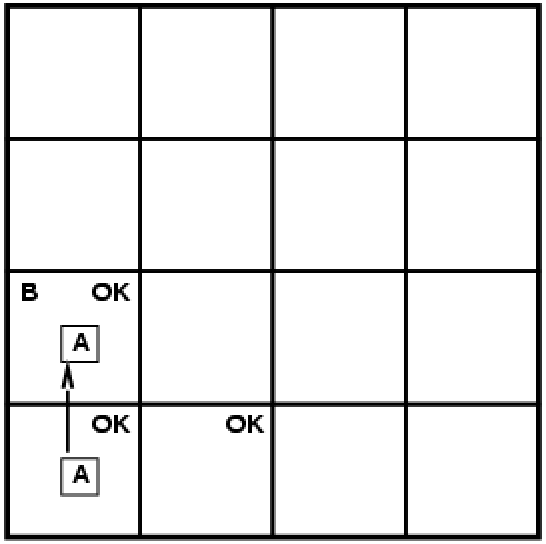


- Performance measure
  - gold: +1000, death: -1000
  - -1 per step, -10 for using the arrow
- Environment
  - Squares adjacent to wumpus are smelly
  - Squares adjacent to pit are breezy
  - Glitter iff gold is in the same square
  - Shooting kills wumpus if you are facing it
  - Shooting uses up the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

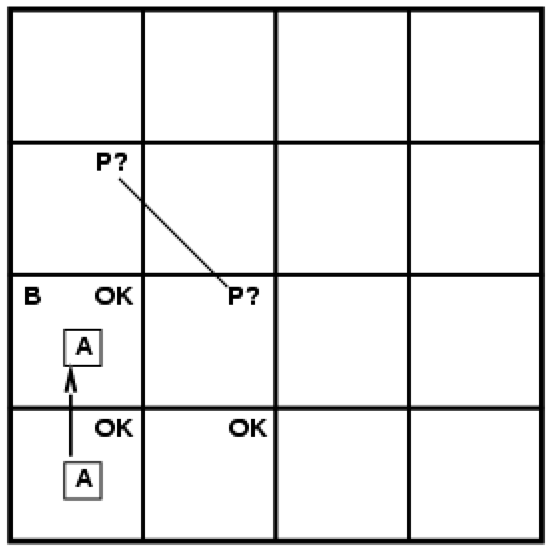
Exploring a wumpus world



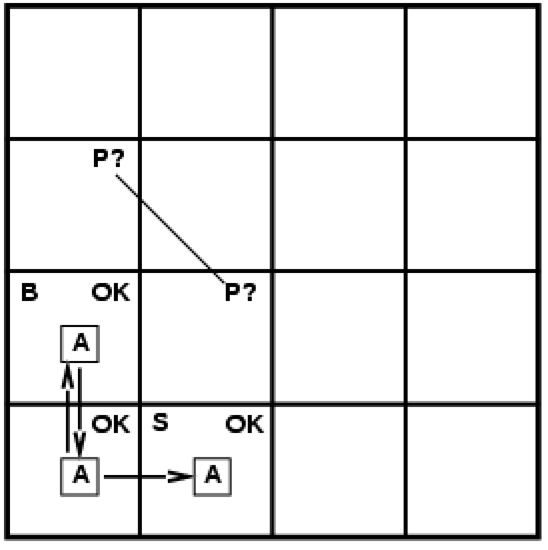
Exploring a wumpus world



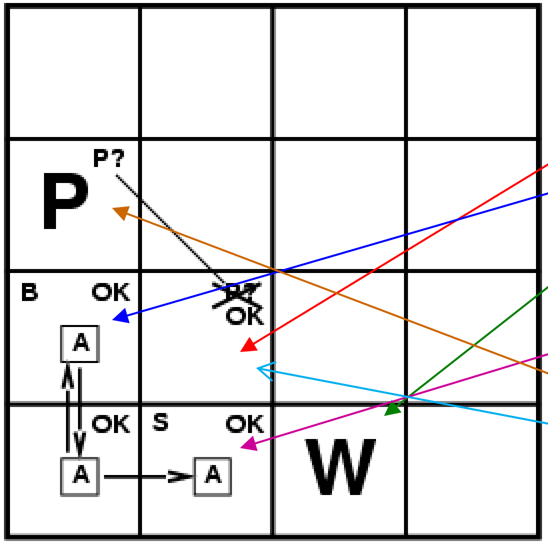
Exploring a wumpus world



Exploring a wumpus world



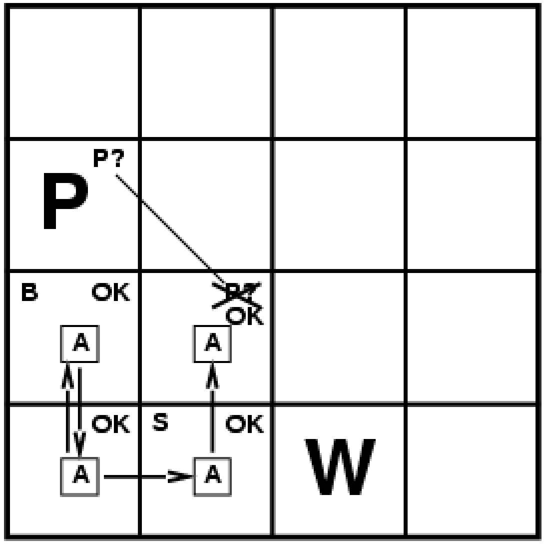
Exploring a Wumpus world



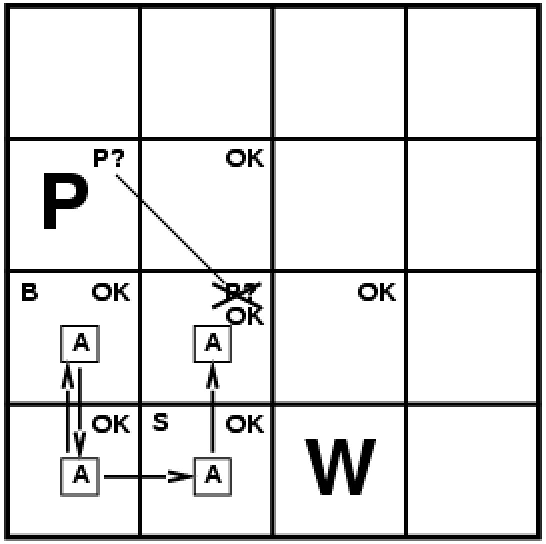
If the Wumpus were here, stench should be here. Therefore it is here.
Since, there is no breeze here, the pit must be there, and it must be OK here

We need rather sophisticated reasoning here!

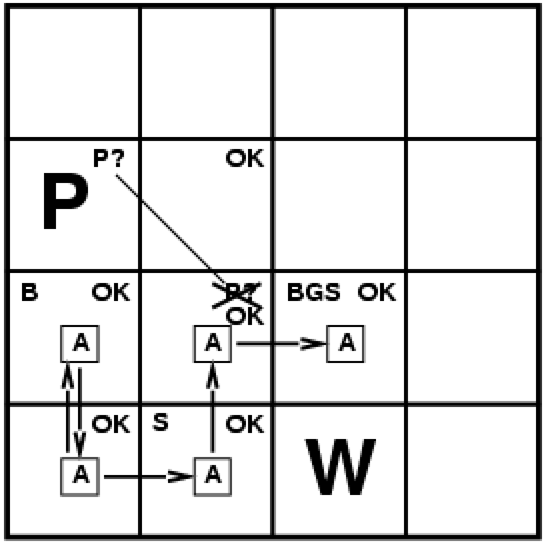
Exploring a wumpus world



Exploring a wumpus world



Exploring a wumpus world



Logic

- We used logical reasoning to find the gold.
- Logics are formal languages for representing information such that conclusions can be drawn from formal inference patterns
- Syntax defines the well-formed sentences in the language
- Semantics define the "meaning" or interpretation of sentences:
  - connect symbols to real events in the world
  - i.e., define truth of a sentence in a world

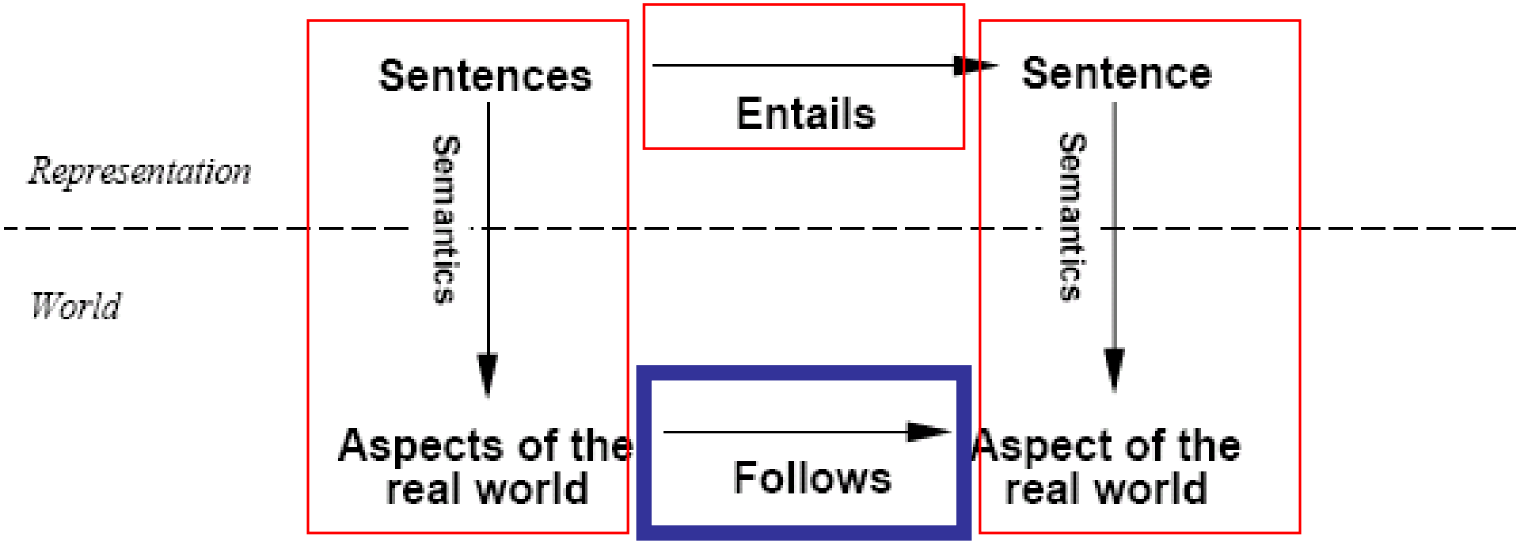
- E.g., the language of arithmetic:
  - $x+2 \geq y$ is a sentence
  - x2+y > {} is not a sentence

  syntax

  - $x+2 \geq y$ is true in a world where $x = 7$, $y = 1$
  - $x+2 \geq y$ is false in a world where $x = 0$, $y = 6$

  semantics

Schematic perspective



If KB is true in the real world,

then any sentence α entailed by KB

is also true in the real world.

For example: If I tell you (1) Sue is Mary's sister, and (2) Sue is Amy's mother, then it **necessarily follows in the world** that Mary is Amy's aunt, even though I told you nothing at all about aunts. This sort of reasoning pattern is what we hope to capture.

Entailment

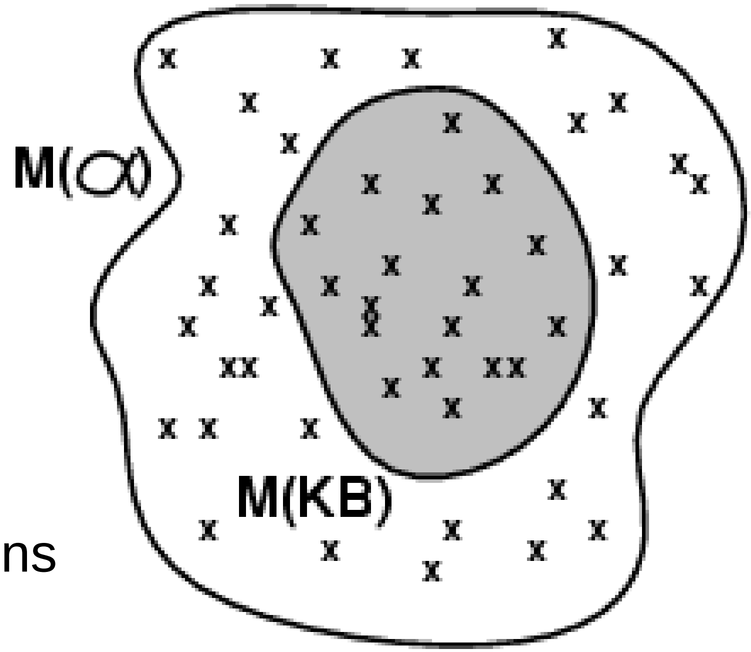
- Entailment means that one thing follows from another set of things:

$$KB \models \alpha$$

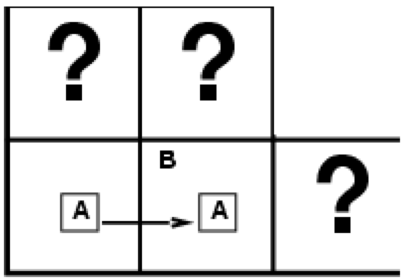
- Knowledge base *KB* entails sentence α if and only if α is true in all worlds wherein *KB* is true
  - E.g., the KB = “the Giants won and the Reds won” entails α = “The Giants won”.
  - E.g., KB = “x+y = 4” entails α = “4 = x+y”
  - E.g., KB = “Mary is Sue’s sister and Amy is Sue’s daughter” entails α = “Mary is Amy’s aunt.”
- The entailed α MUST BE TRUE in ANY world in which KB IS TRUE.

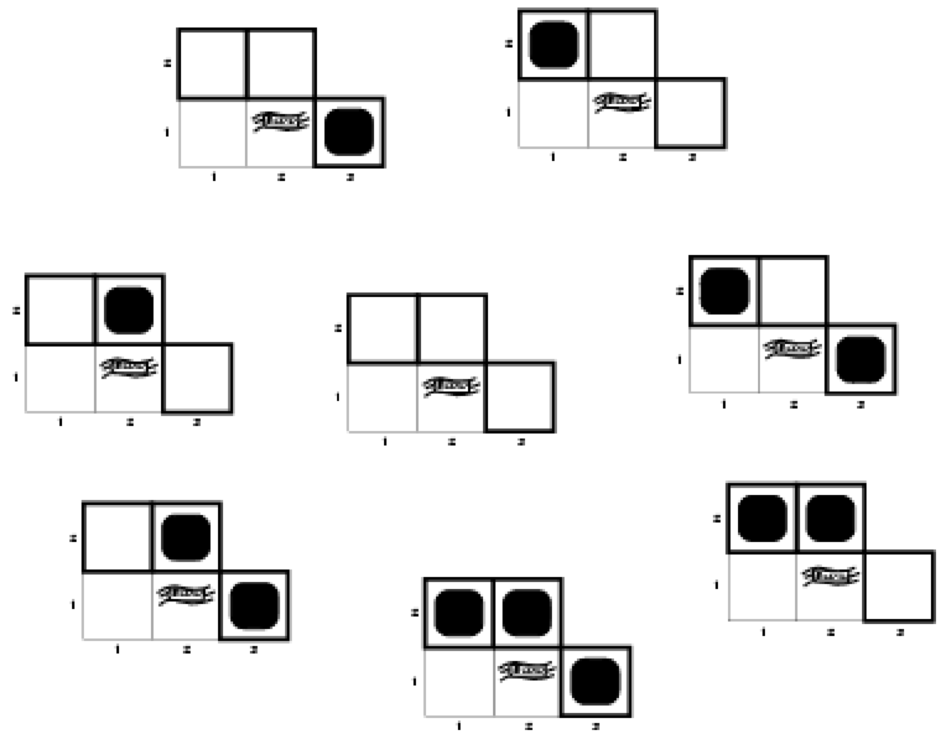
Models

- 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)
  - E.g. *KB* = Giants won and Reds won entails α = Giants won
- Think of KB and α as collections of constraints and of models m as possible states. M(KB) are the solutions to KB and M(α) the solutions to α. Then, KB $\models$ α when all solutions to KB are also solutions to α.



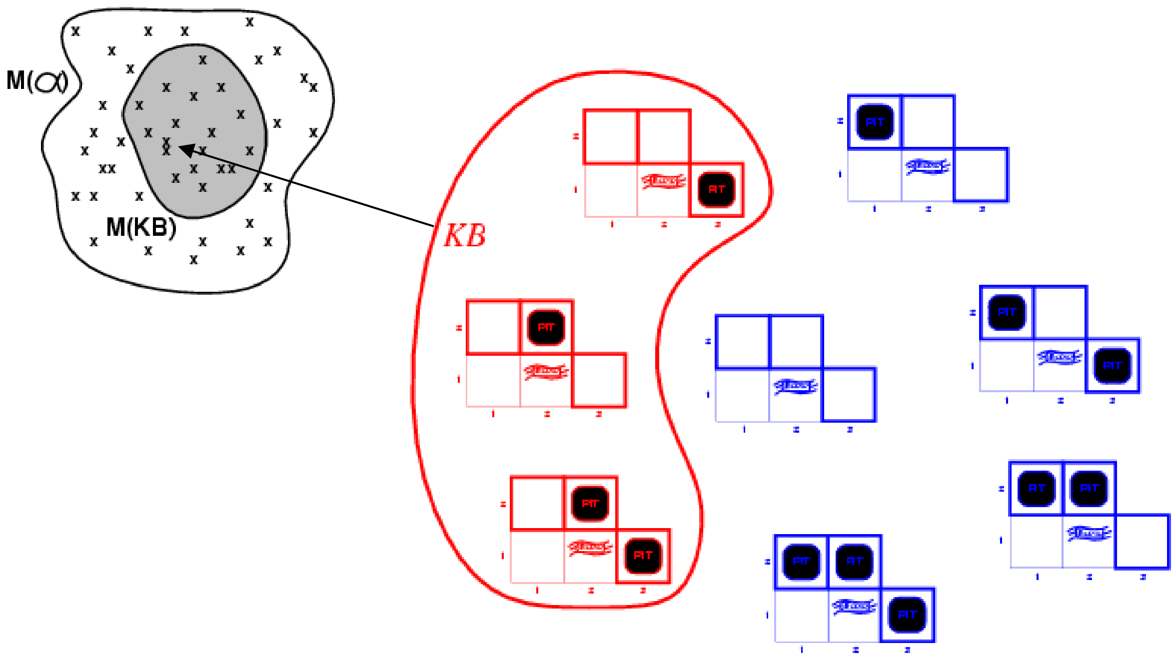
Wumpus models





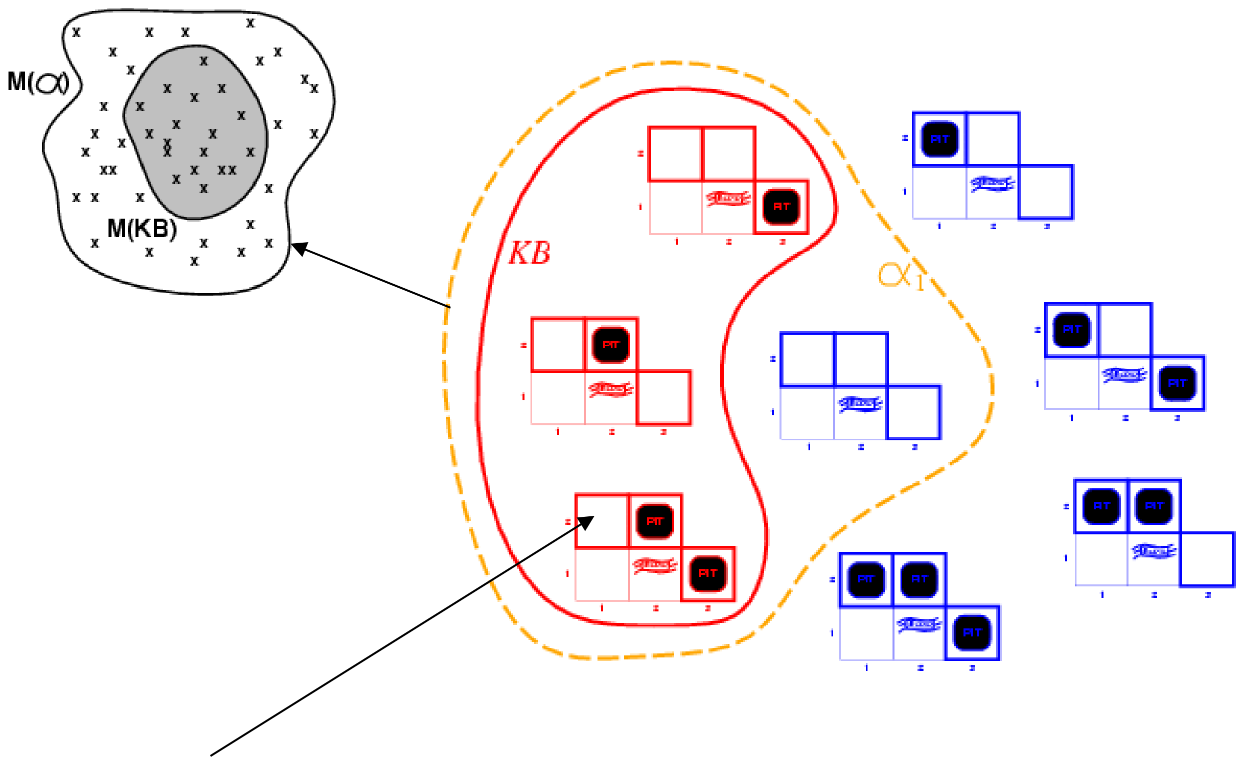
All possible models in this reduced Wumpus world. What can we infer?

Wumpus models



- *M(KB)* = all possible wumpus-worlds consistent with the observations and the “physics” of the Wumpus world.

Wumpus models



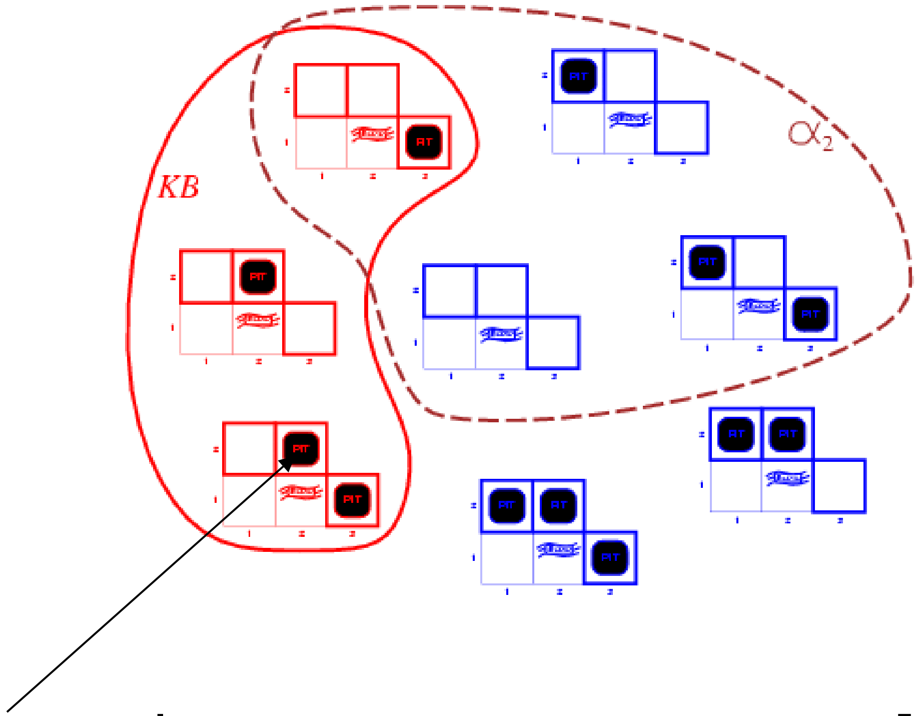
Now we have a query sentence, α_1 = "[1,2] is safe“

KB $\models \alpha_1$, proved by **model checking**

M(KB) (red outline) is a subset of M(α_1) (orange dashed outline)

$\Rightarrow$ **α_1 is true in any world in which KB is true**

Wumpus models



Now we have another query sentence, α_2 = "[2,2] is safe"

KB $\not\models$ α_2, proved by **model checking**

M(KB) (red outline) is a **not** a subset of M(α_2) (dashed outline)

$\Rightarrow$ α_2 is false in some world(s) in which KB is true

Recap 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)

Recap propositional logic: Semantics

Each model/world specifies true or 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.,

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

* iff = if and only if

Recap truth tables for connectives

P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \Rightarrow Q$	$P \Leftrightarrow Q$
$false$	$false$	$true$	$false$	$false$	$true$	$true$
$false$	$true$	$true$	$false$	$true$	$true$	$false$
$true$	$false$	$false$	$false$	$true$	$false$	$false$
$true$	$true$	$false$	$true$	$true$	$true$	$true$

OR: P or Q is true or both are true.
XOR: P or Q is true but not both.

Implication is always true when the premises are False!

Inference by enumeration
(generate the truth table = model checking)

- Enumeration of all models is sound and complete.
- For *n* symbols, time complexity is $O(2^n)$...
- We need a smarter way to do inference!
- In particular, we are going to infer new logical sentences from the data-base and see if they match a query.

Logical equivalence

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

You need to know these !

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

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 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)

Summary (Part I)

- 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
  - valid: sentence is true in every model (a tautology)

- Logical equivalences allow syntactic manipulations

- Propositional logic lacks expressive power
  - Can only state specific facts about the world.
  - Cannot express general rules about the world (use First Order Predicate Logic)