Logical Agents

Slides adopted from
Dr. Surangika Ranathunga

Week 6: Introduction to logical agents

Week 7: Logical reasoning

Week 8: Knowledge representation

Week 9: Planning

Outline

Fundamental concepts

General principles of logic

Propositional logic

First order logic

Agents

Problem solving agent
- Solution is given, the agent executes it
- Might not face dynamic problems well

Knowledge based / Planning agents
- Given explicit goals
- Can achieve competence quickly by being told or learning
- Adapt to changes in environment

Knowledge-based Agents

Maintain a representation of the world

infer new representations of the world

Use the representation to decide what to do

representation -> reason -> take action

Representations

Atomic:
State considered as a whole, No internal structure available to the agent

Factored:
Assignment of values to variables

Structured:
Objects and relations
Facts: knowledge about relations

Knowledge ?

“Knowing things which helps to reach goals efficiently”

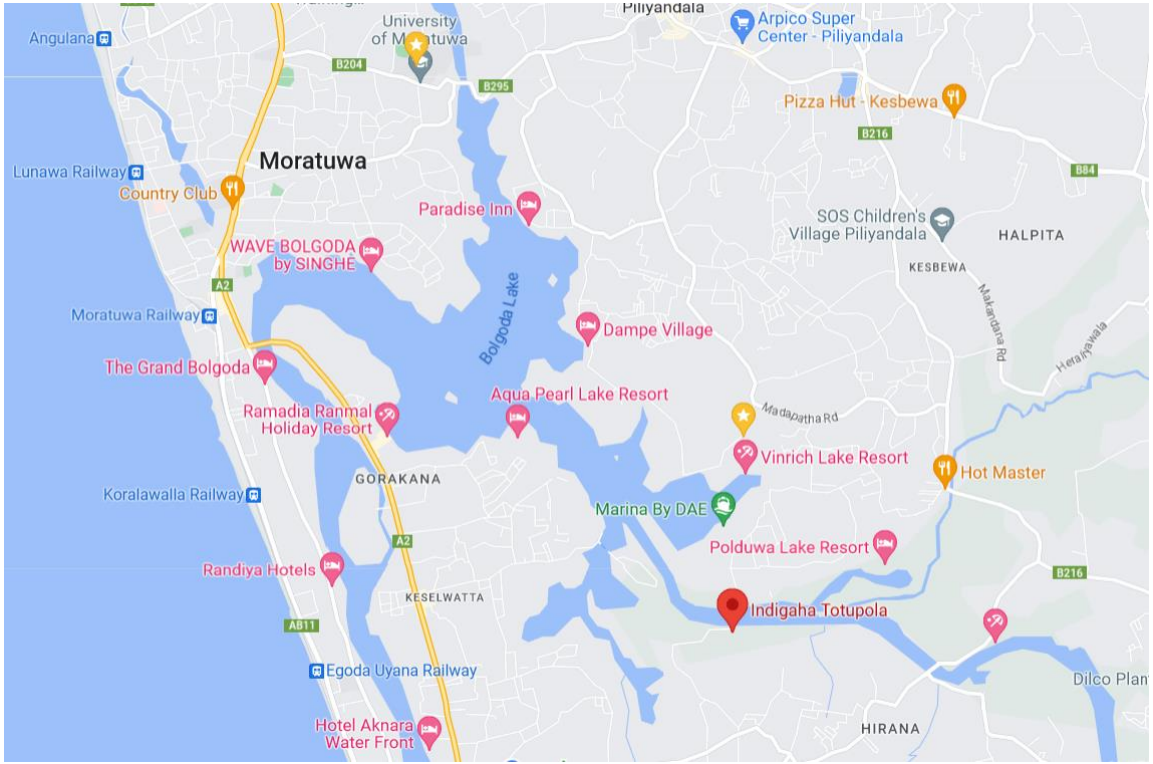
Example:

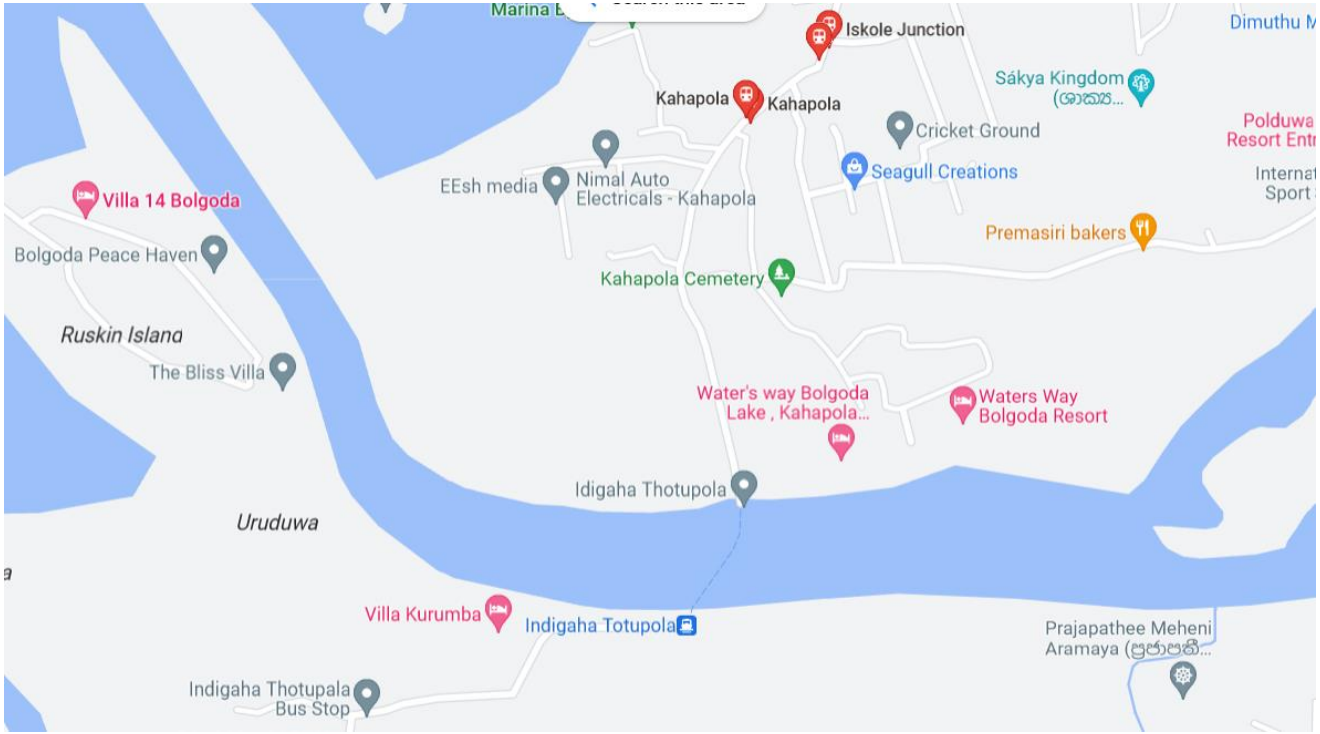
How to go from San Francisco to Marin County?

How to go from University of Moratuwa to Indigaha Thotupola

Procedural

Declarative





Learning Agents?

Knowledge Base (KB)

Where the representation of the world is maintained

Consists of a set of "sentences" -> written in a knowledge representation

language Base -> a set of axioms

Axiom: ?

KB

Need to

- Add new info to the KB (TELL)
- Retrieve info from KB (ASK)

both involve inference -> derive new info from old

knowledge level (world representation, agents goals) vs implementation level

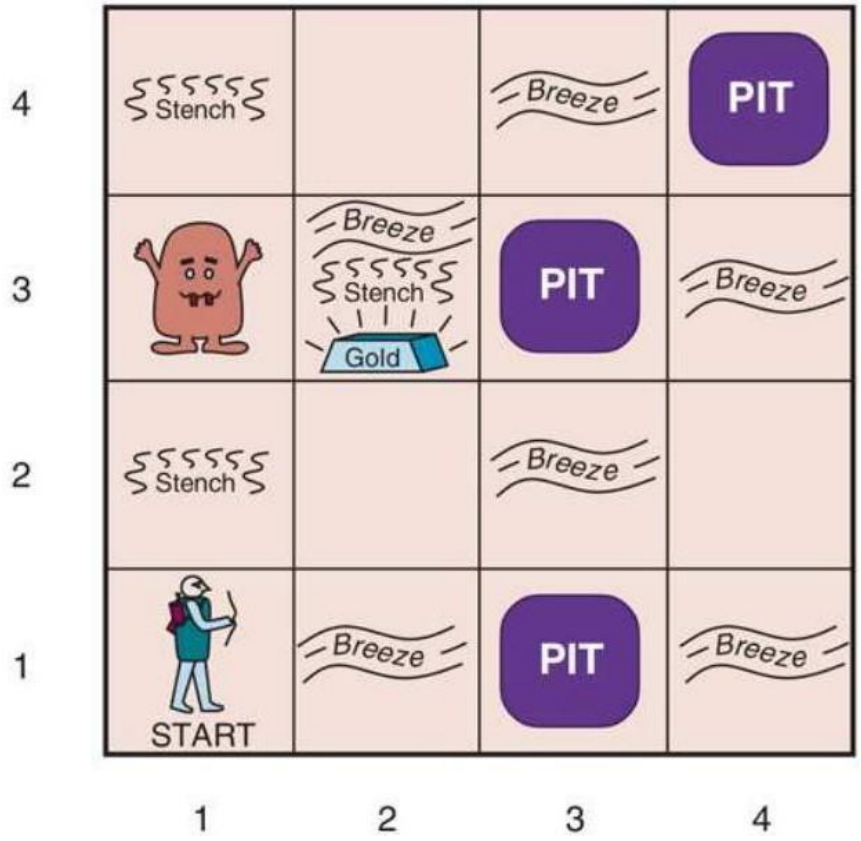
```
function KB-AGENT(percept) returns an action
  persistent: KB, a knowledge base
              t, a counter, initially 0, indicating time

  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
```

Real-world applications of the Wumpus World :

Designing intelligent agents for autonomous vehicles, robotics, and game creation.

The Wumpus World



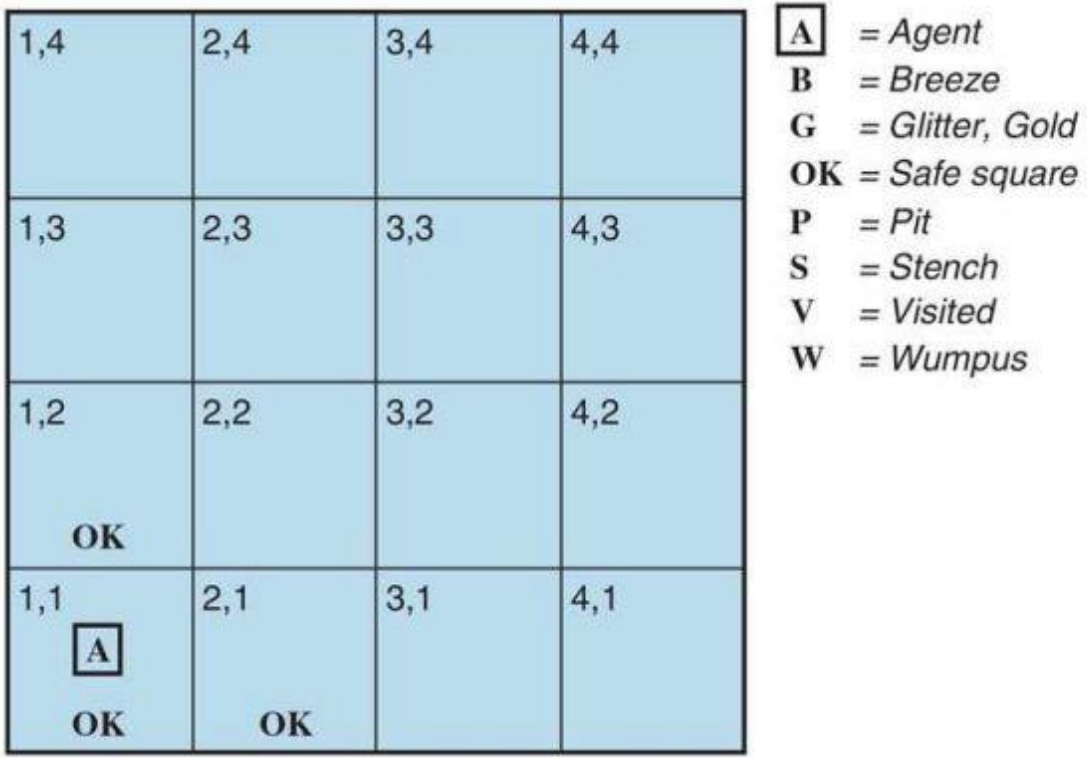
Describing the Environment

Properties of the Wumpus World

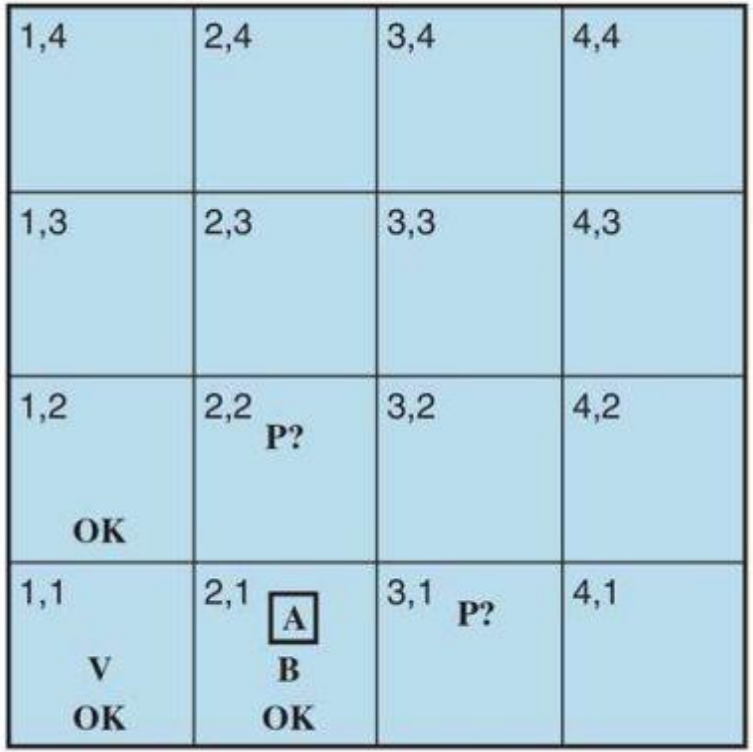
- Partially observable: The Wumpus world in AI is partially observable because the agent can only sense the immediate surroundings, such as an adjacent room.
- Deterministic: It is deterministic because the result and end of the world are already known.
- Sequential: It is sequential because the order is essential.
- Static: It is motionless because Wumpus and Pits are not moving.
- Discrete: The surroundings are distinct.
- One agent: The environment is a single agent because we only have one agent, and Wumpus is not regarded as an agent.

Agent’s initial KB

Rules of the environment

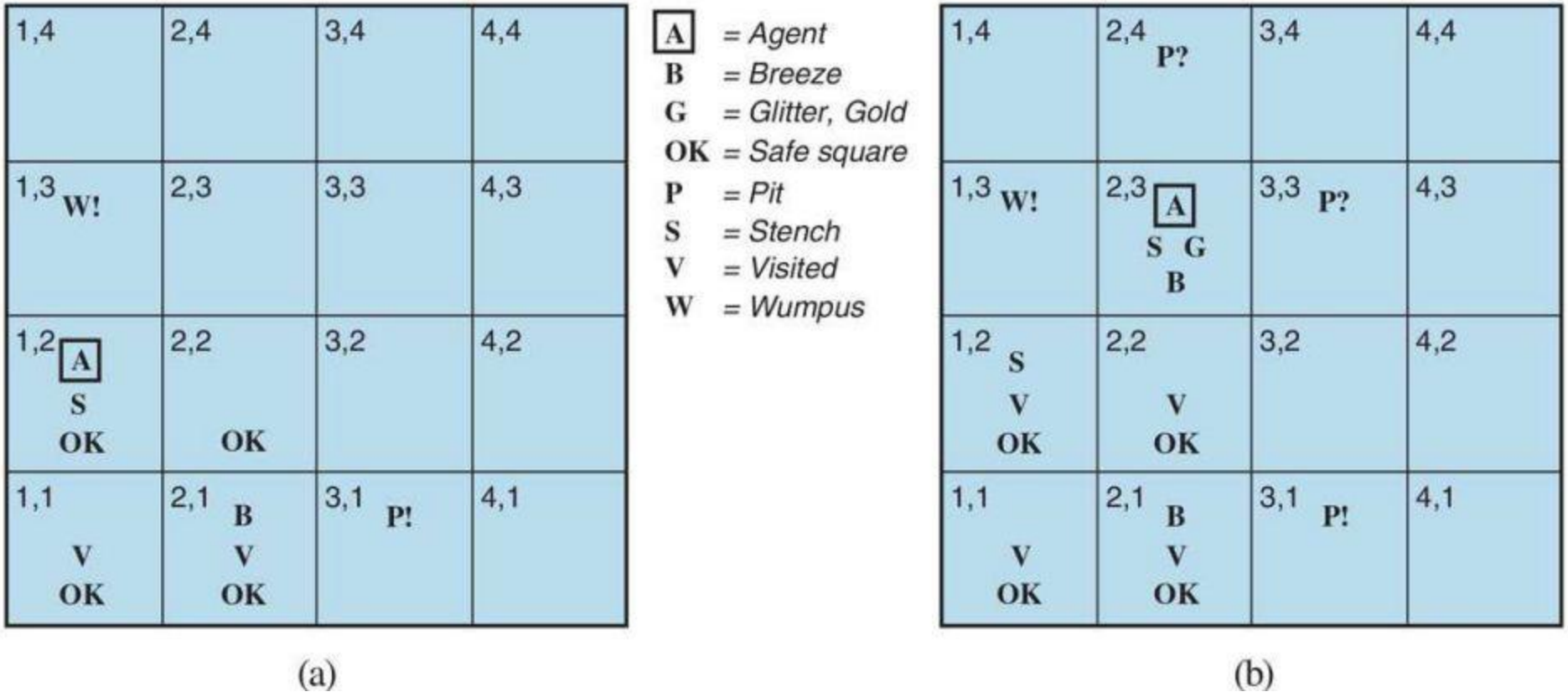


(a)



(b)

[stench, breeze, glitter, bump, scream]


1,4
2,4
3,4
4,4
1,3 W!
2,3
3,3
4,3
1,2 A S OK
2,2 OK
3,2
4,2
1,1 V OK
2,1 B V OK
3,1 P!
4,1
A = Agent
B = Breeze
G = Glitter, Gold
OK = Safe square
P = Pit
S = Stench
V = Visited
W = Wumpus
1,4
2,4 P?
3,4
4,4
1,3 W!
2,3 A S G B
3,3 P?
4,3
1,2 S V OK
2,2 V OK
3,2
4,2
1,1 V OK
2,1 B V OK
3,1 P!
4,1

(a) (b)

Logic

Fundamental concepts of logical representation and reasoning

Logical Languages

A KB has sentences

Expression of sentences should be syntactically correct

x+y =9

+2+= y

Syntax??

Semantics

Define the truth of sentences wrt each possible world

Possible world = model = an abstraction where each sentence is either true or false

x+y = 9 ???

If a sentence ***s*** is true in a model ***m*** => ***m*** satisfies ***s*** / ***m*** is a model of ***s***

Set of *m* which are models of ***s*** **:** M(s)

Logical Reasoning

Logical **entailment** between sentences

$\alpha \models \beta$ if and only if, in every model in which α is true, β is also true

$\alpha \models \beta$ if and only if $M(\alpha) \subseteq M(\beta)$

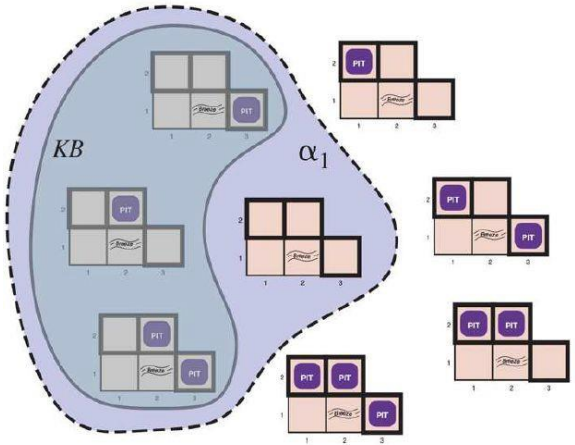
$x=0 \models xy=0$

A = Agent
B = Breeze
G = Glitter, Gold
OK = Safe square
P = Pit
S = Stench
V = Visited
W = Wumpus

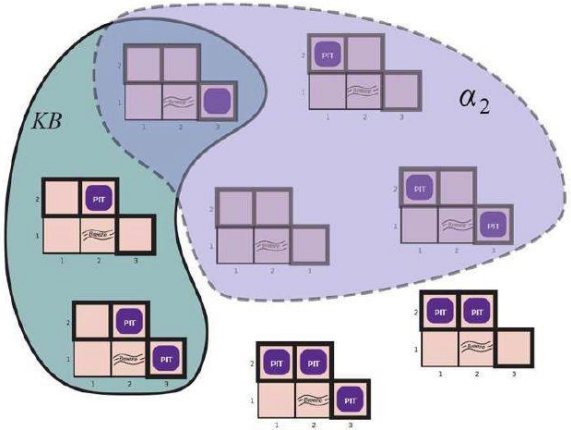
1,4	2,4	3,4	4,4
1,3	2,3	3,3	4,3
1,2 OK	2,2 P?	3,2	4,2
1,1 V OK	2,1 A B OK	3,1 P?	4,1

(b)

α_1 =" There is no pit in [1,2]." α_2 =" There is no pit in [2,2]."



(a)



(b)

Logical Inference

Derive conclusions using entailment

Model checking - enumerates all possible models to check that a sentence is true in all the models where the KB is true

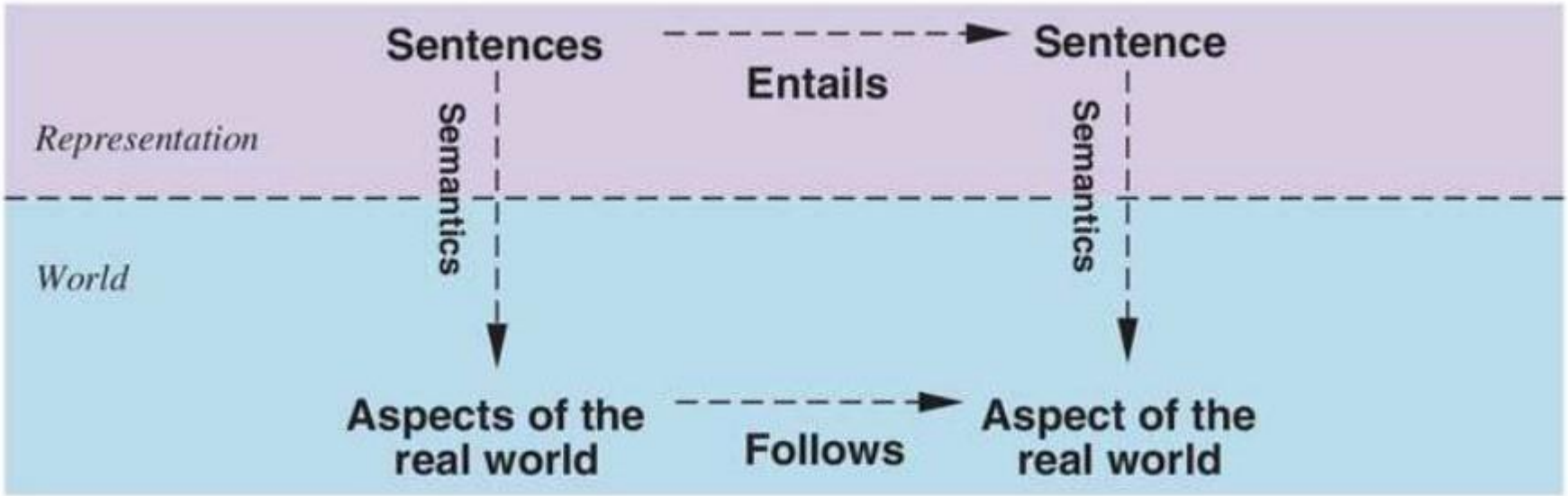
$KB \vdash_i \alpha$

i - inference algorithm, should be **sound**, **truth preserving** and **complete**

Model checking works if your space of models is finite

Later….. Theorem proving

Grounding



if KB is true in the real world, then any
sentence derived from KB by a sound inference procedure is also true in the real world

Propositional Logic - syntax

Propositional symbols - stands for a proposition that can be true or false

P,Q,R,$W_{1,2}$ and FacingEast, **True, False**

Atomic sentences - Consists of s single propositional symbol

Literal - atomic sentence or its negation (negative literal)

Complex sentences are constructed from simpler sentences, using parentheses and operators called logical connectives

¬ , ∧ , ∨ , ⇒ , ⇔

$$
\begin{aligned}
\mathit{Sentence} &\rightarrow \mathit{AtomicSentence} \mid \mathit{ComplexSentence} \\
\mathit{AtomicSentence} &\rightarrow \mathit{True} \mid \mathit{False} \mid P \mid Q \mid R \mid \ldots \\
\mathit{ComplexSentence} &\rightarrow (\mathit{Sentence}) \\
&\mid \neg \mathit{Sentence} \\
&\mid \mathit{Sentence} \wedge \mathit{Sentence} \\
&\mid \mathit{Sentence} \vee \mathit{Sentence} \\
&\mid \mathit{Sentence} \Rightarrow \mathit{Sentence} \\
&\mid \mathit{Sentence} \Leftrightarrow \mathit{Sentence}
\end{aligned}
$$

OPERATOR PRECEDENCE : $\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow$

Propositional Logic - semantics

Defines the rules for determining the truth of a sentence with respect to a particular model

Models in propositional logic - simply sets the truth value—true or false— for every proposition symbol

E.g. proposition symbols P,Q, R

One possible model m_1 = {P= true, Q=true, R=false}

Computing the Truth Value of Sentences

The semantics for propositional logic must specify how to compute the truth value of any sentence, given a model

¬ P

Q ∧ P

P ∨ Q

P ⇒ Q

P ⇔ Q

P	*Q*
false	*false*
false	*true*
true	*false*
true	*true*

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 Knowledge Base

$P_{x,y}$ is true if there is 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 there is a breeze in $[x,y]$.

$S_{x,y}$ is true if there is a stench in $[x,y]$.

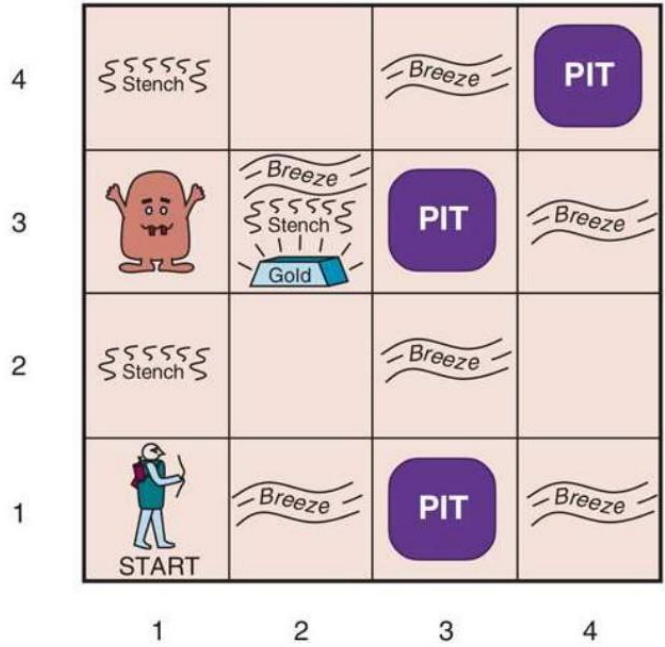
$L_{x,y}$ is true if the agent is in location $[x,y]$.

New sentences R_1 R_2 R_3
There is no pit in [1,1] R_1 =

A square is breezy if and only if there is a pit in a neighboring square(consider [1,1]) R_2 : ??

R_3 :??

$$R_4 : \quad \neg B_{1,1}$$
$$R_5 : \quad B_{2,1} .$$



Inferencing

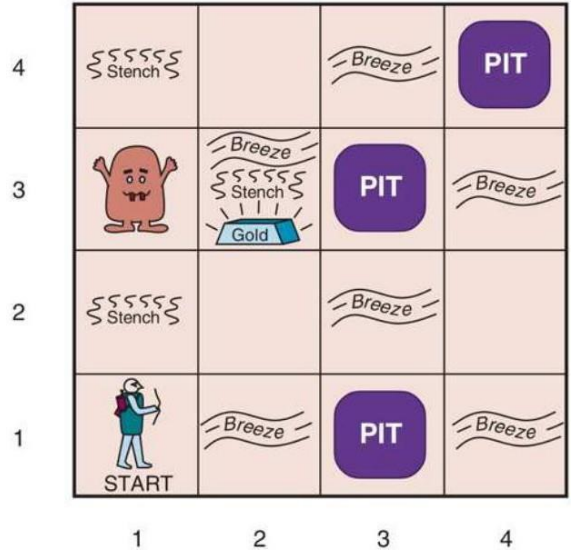
Is a sentence α entailed by the KB??

$KB \models \alpha$

One way to implement an inference algorithm is model checking

Enumerate all the models

Check if α is true when KB is true



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

the agent has detected nothing in [1,1] and a breeze in [2,1]

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

Theorem Proving

applying rules of inference directly to the sentences in our knowledge base to construct a proof of the desired sentence without consulting models

Beneficial over model checking if there are a large number of models

Logical Equivalence

Two sentences are logically equivalent if they are true in the same set of models

$\alpha \equiv \beta$ if and only if $\alpha \models \beta$ and $\beta \models \alpha$

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

Validity

sentence is valid if it is true in all models

Valid sentences are also known as tautologies

Useful in developing deduction theorems

Satisfiability

A sentence is satisfiable if it is true in, or satisfied by, some model

Satisfiability can be checked by enumerating the possible models until one is found that satisfies the sentence

Validity and satisfiability are connected:

α is valid if ¬α is unsatisfiable; contrapositively, α is satisfiable iff ¬α is not valid

Monotonicity

The set of entailed sentences can only increase as information is added to the knowledge base

$$KB \models \alpha \quad \text{then} \quad KB \wedge \beta \models \alpha$$

Inference and Proofs

inference rules that can be applied to derive a proof —a chain of conclusions that leads to the desired goal

Modes Ponens $$\frac{\alpha \Rightarrow \beta, \quad \alpha}{\beta}$$

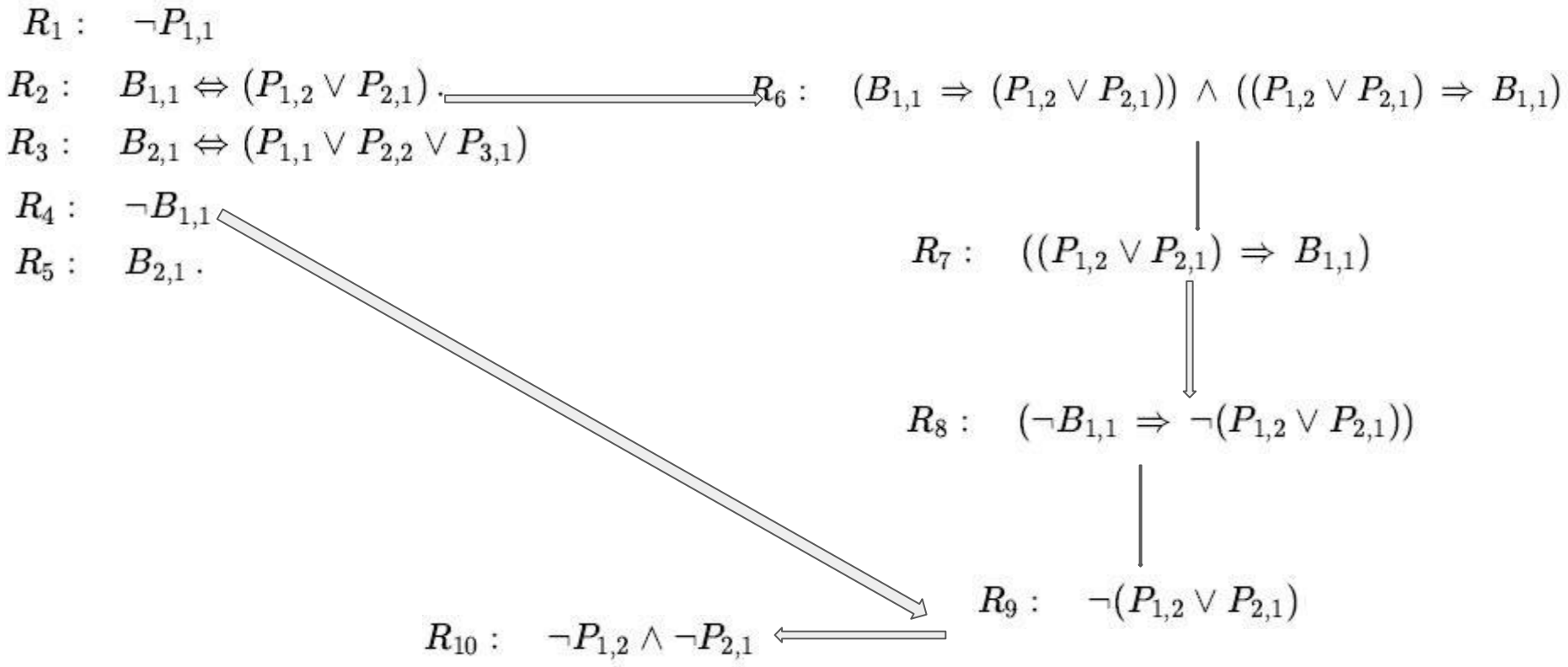
And elimination $$\frac{\alpha \wedge \beta}{\alpha}$$

All of the logical equivalences can be used as inference rules

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

This biconditional elimination yields the following inference rules

$$\frac{\alpha \Leftrightarrow \beta}{(\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)} \quad \text{and} \quad \frac{(\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)}{\alpha \Leftrightarrow \beta}$$


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 : $B_{2,1}$.
R_6 : $(B_{1,1} \Rightarrow (P_{1,2} \vee P_{2,1})) \wedge ((P_{1,2} \vee P_{2,1}) \Rightarrow B_{1,1})$
R_7 : $((P_{1,2} \vee P_{2,1}) \Rightarrow B_{1,1})$
R_8 : $(\neg B_{1,1} \Rightarrow \neg(P_{1,2} \vee P_{2,1}))$
R_9 : $\neg(P_{1,2} \vee P_{2,1})$
R_{10} : $\neg P_{1,2} \wedge \neg P_{2,1}$

INITIAL STATE: the initial knowledge base.

ACTIONS: the set of actions consists of all the inference rules applied to all the sentences that match the top half of the inference rule.

RESULT: the result of an action is to add the sentence in the bottom half of the inference rule.

GOAL: the goal is a state that contains the sentence we are trying to prove

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

Proof by Resolution

$R_1 : \quad \neg P_{1,1}$

$R_2 : \quad B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})\,.$

$R_3 : \quad B_{2,1} \Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{3,1})$

$R_4 : \quad \neg B_{1,1}\,.$

$R_5 : \quad B_{2,1}\,.$

$R_6 : \quad (B_{1,1} \Rightarrow (P_{1,2} \vee P_{2,1})) \wedge ((P_{1,2} \vee P_{2,1}) \Rightarrow B_{1,1})$

$R_7 : \quad ((P_{1,2} \vee P_{2,1}) \Rightarrow B_{1,1})$

$R_8 : \quad (\neg B_{1,1} \Rightarrow \neg(P_{1,2} \vee P_{2,1}))$

$R_9 : \quad \neg(P_{1,2} \vee P_{2,1})$

$R_{10} : \quad \neg P_{1,2} \wedge \neg P_{2,1}$

$R_{11} : \quad \neg B_{1,2}.$

$R_{12} : \quad B_{1,2} \Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{1,3})$

$R_{13} : \quad \neg P_{2,2}$

$R_{14} : \quad \neg P_{1,3}$

$R_{15} : \quad P_{1,1} \vee P_{2,2} \vee P_{3,1}$

$R_{16} : \quad P_{1,1} \vee P_{3,1}$

$R_{17} : \quad P_{3,1}$

CNF

Resolution rule applies only to disjunctions of literals (clauses)

Every sentence of propositional logic is logically equivalent to a conjunction of clauses

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

Eliminate the biconditional $(B_{1,1} \Rightarrow (P_{1,2} \vee P_{2,1})) \wedge ((P_{1,2} \vee P_{2,1}) \Rightarrow B_{1,1})$

Eliminate => $(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge (\neg(P_{1,2} \vee P_{2,1}) \vee B_{1,1})$

CNF

CNF requires the negation to appear only in literals

$$\neg(\neg\alpha) \equiv \alpha \text{ (double-negation elimination)}$$
$$\neg(\alpha \wedge \beta) \equiv (\neg\alpha \vee \neg\beta) \text{ (De Morgan)}$$
$$\neg(\alpha \vee \beta) \equiv (\neg\alpha \wedge \neg\beta) \text{ (De Morgan)}$$

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

Apply Distributive Law $(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge (\neg P_{1,2} \vee B_{1,1}) \wedge (\neg P_{2,1} \vee B_{1,1})$