
Introduction to Artificial Intelligence

Logical Agents

(Logic, Deduction, Knowledge Representation)

Bernhard Beckert



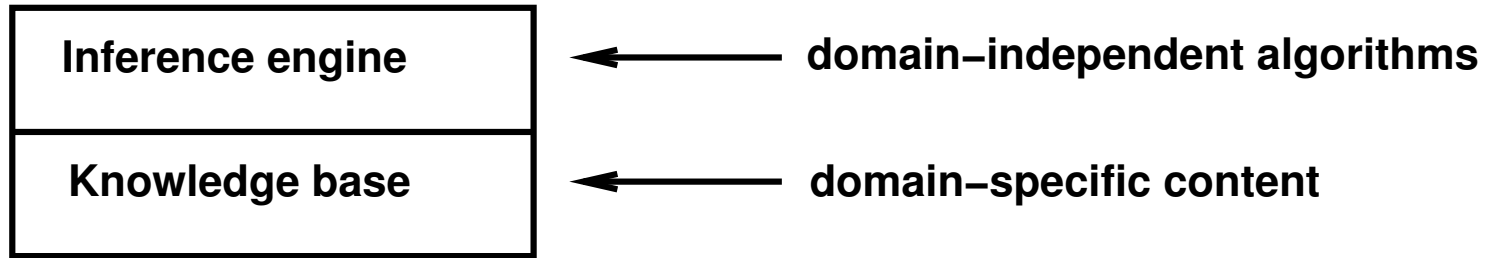
UNIVERSITÄT KOBLENZ-LANDAU

Winter Term 2004/2005

Outline

- **Knowledge-based agents**
- **Wumpus world**
- **Logic in general—models and entailment**
- **Propositional (Boolean) logic**
- **Equivalence, validity, satisfiability**
- **Inference rules and theorem proving**
 - **forward chaining**
 - **backward chaining**
 - **resolution**

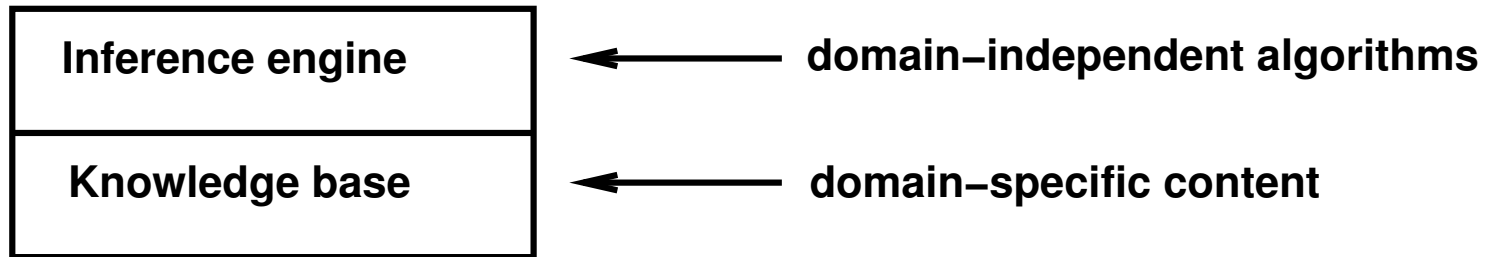
Knowledge bases



Knowledge base

Set of **sentences** in a **formal** language

Knowledge bases



Knowledge base

Set of **sentences** in a **formal** language

Declarative approach to building an agent

Tell it what it needs to know

Then it can ask itself what to do—answers follow from the knowledge base

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 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

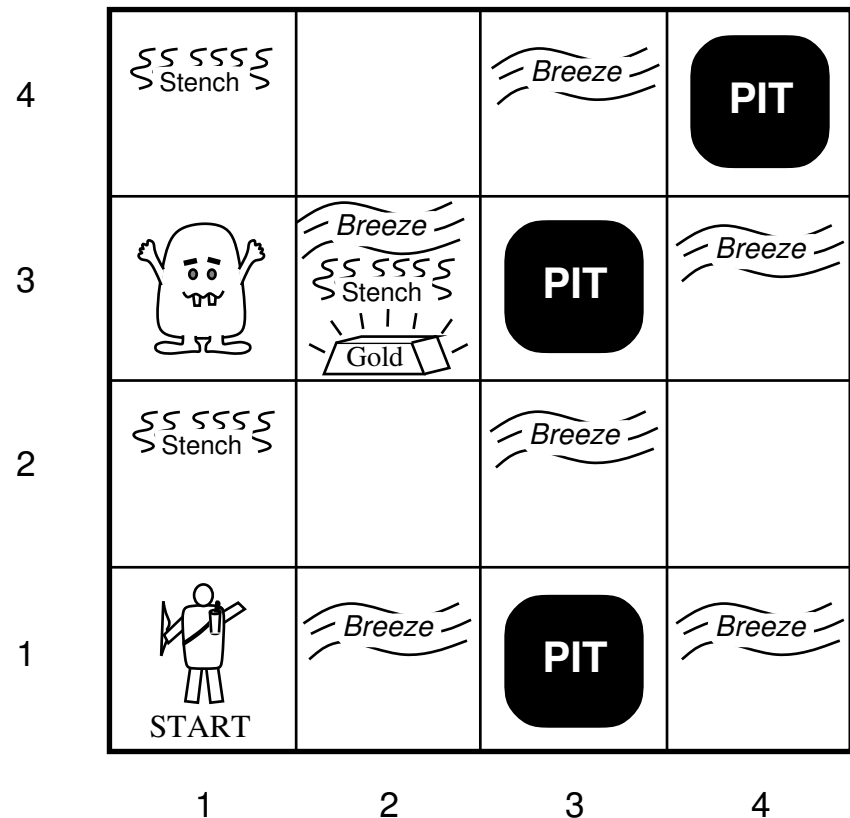
Actuators

Left turn, Right turn,

Forward, Grab, Release, Shoot

Sensors

Breeze, Glitter, Smell



Wumpus World Characterization

Observable

Deterministic

Episodic

Static

Discrete

Single agent

Wumpus World Characterization

Observable **No** – only local perception

Deterministic

Episodic

Static

Discrete

Single agent

Wumpus World Characterization

Observable **No** – only local perception

Deterministic **Yes** – outcome of action exactly specified

Episodic

Static

Discrete

Single agent

Wumpus World Characterization

Observable **No** – only local perception

Deterministic **Yes** – outcome of action exactly specified

Episodic **No** – sequential at the level of actions

Static

Discrete

Single agent

Wumpus World Characterization

Observable **No** – only local perception

Deterministic **Yes** – outcome of action exactly specified

Episodic **No** – sequential at the level of actions

Static **Yes** – wumpus and pits do not move

Discrete

Single agent

Wumpus World Characterization

Observable	No – only local perception
Deterministic	Yes – outcome of action exactly specified
Episodic	No – sequential at the level of actions
Static	Yes – wumpus and pits do not move
Discrete	Yes
Single agent	

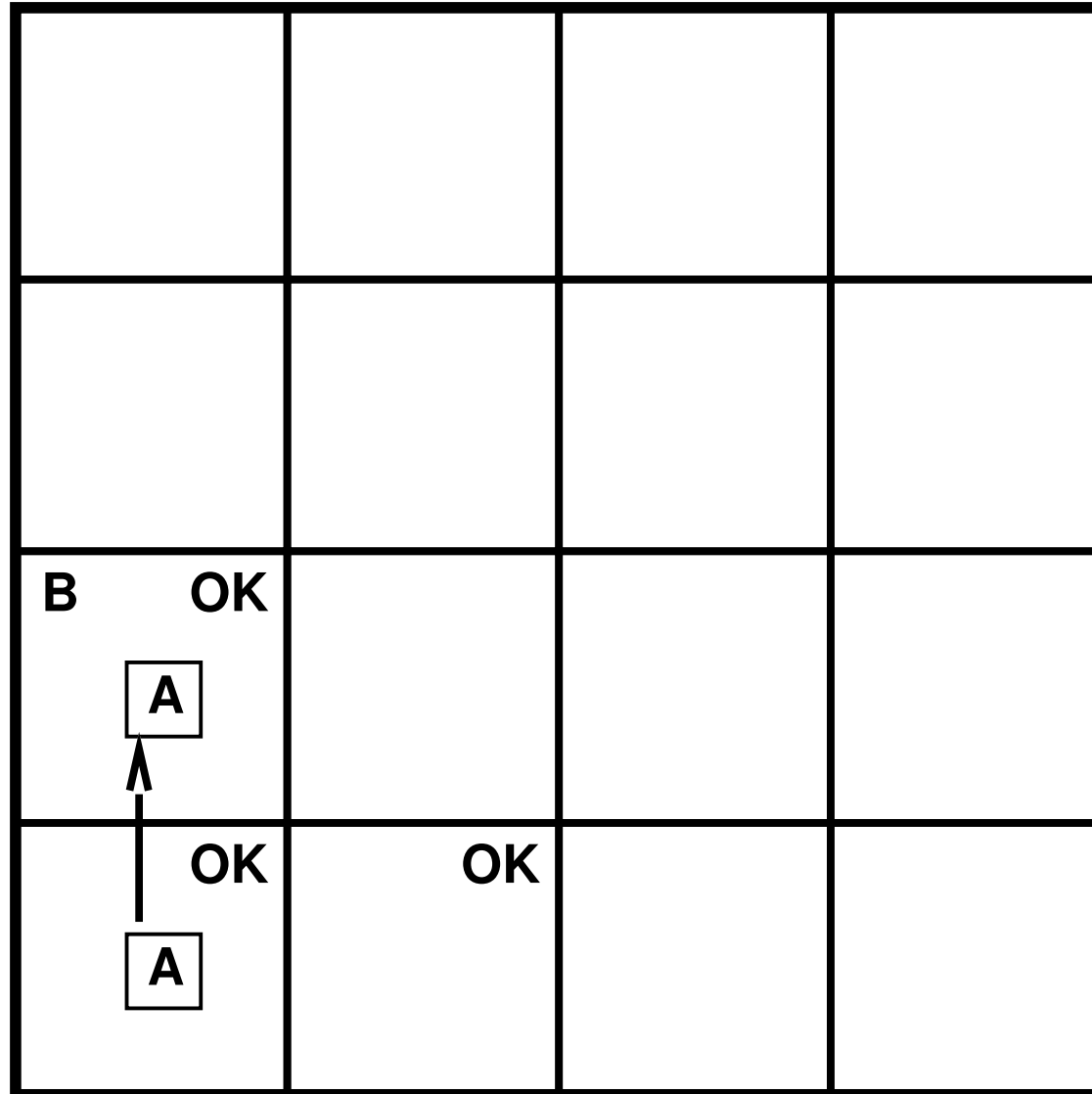
Wumpus World Characterization

Observable	No – only local perception
Deterministic	Yes – outcome of action 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

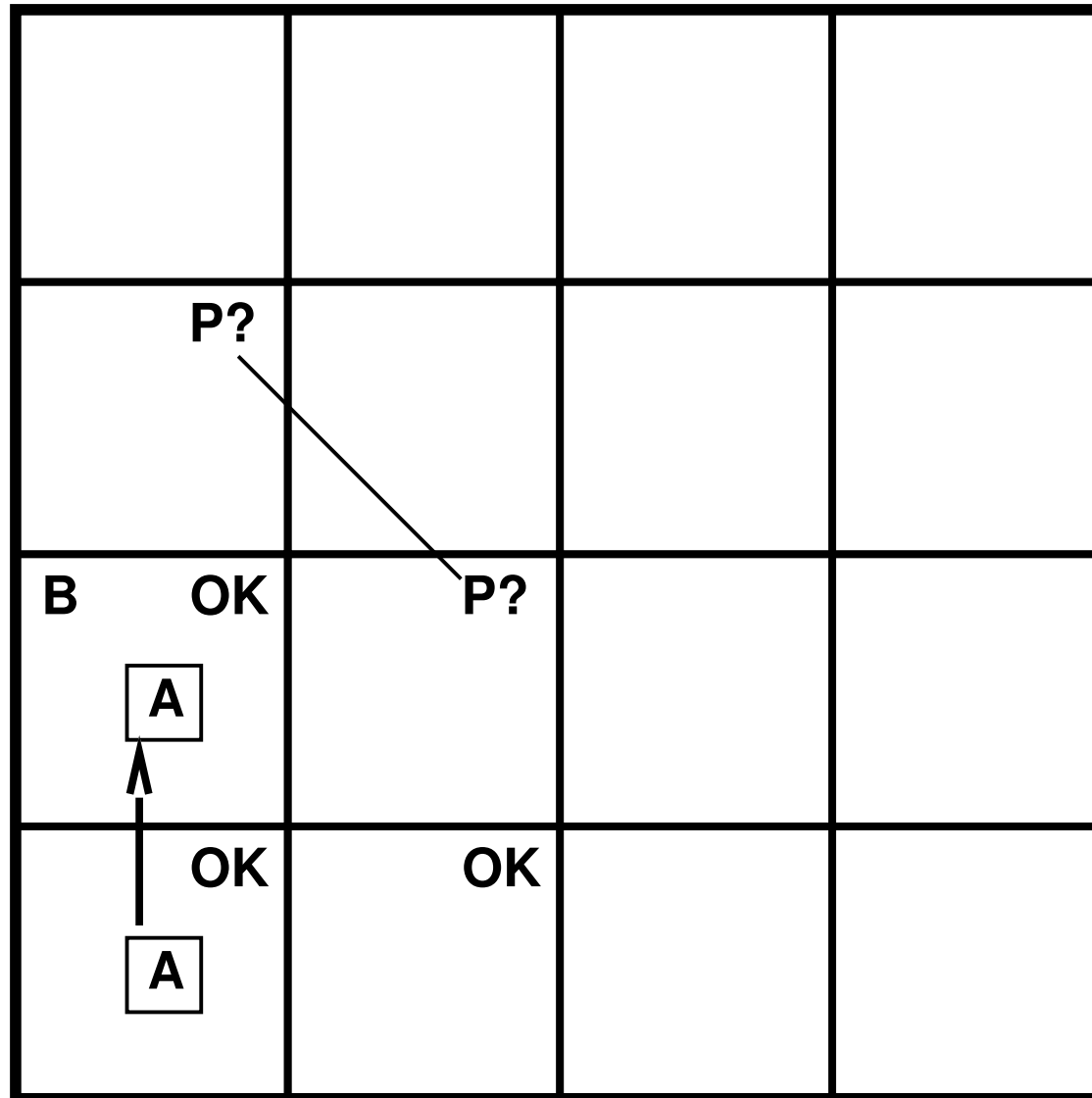
Exploring a Wumpus World

OK			
OK A	OK		

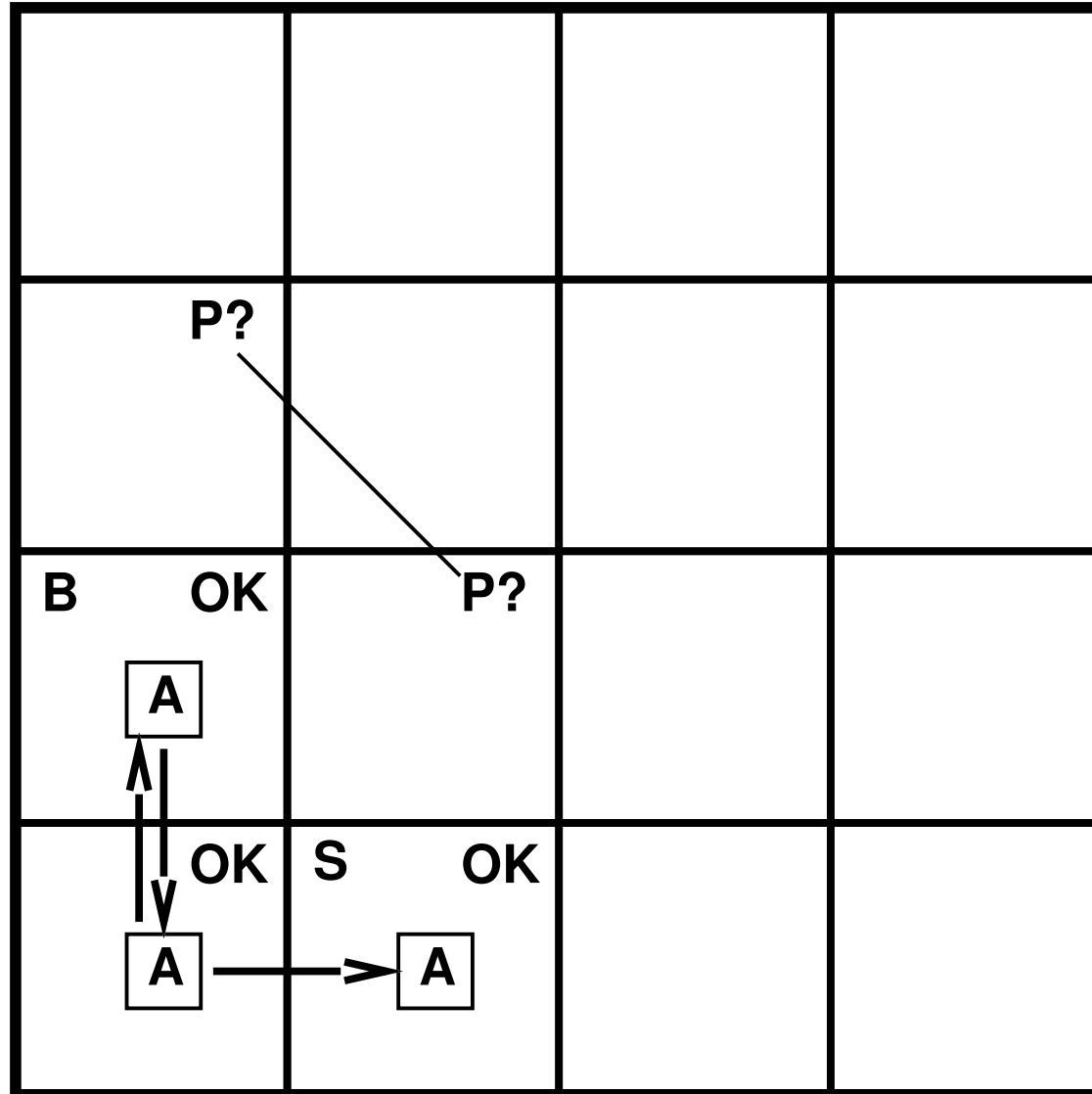
Exploring a Wumpus World



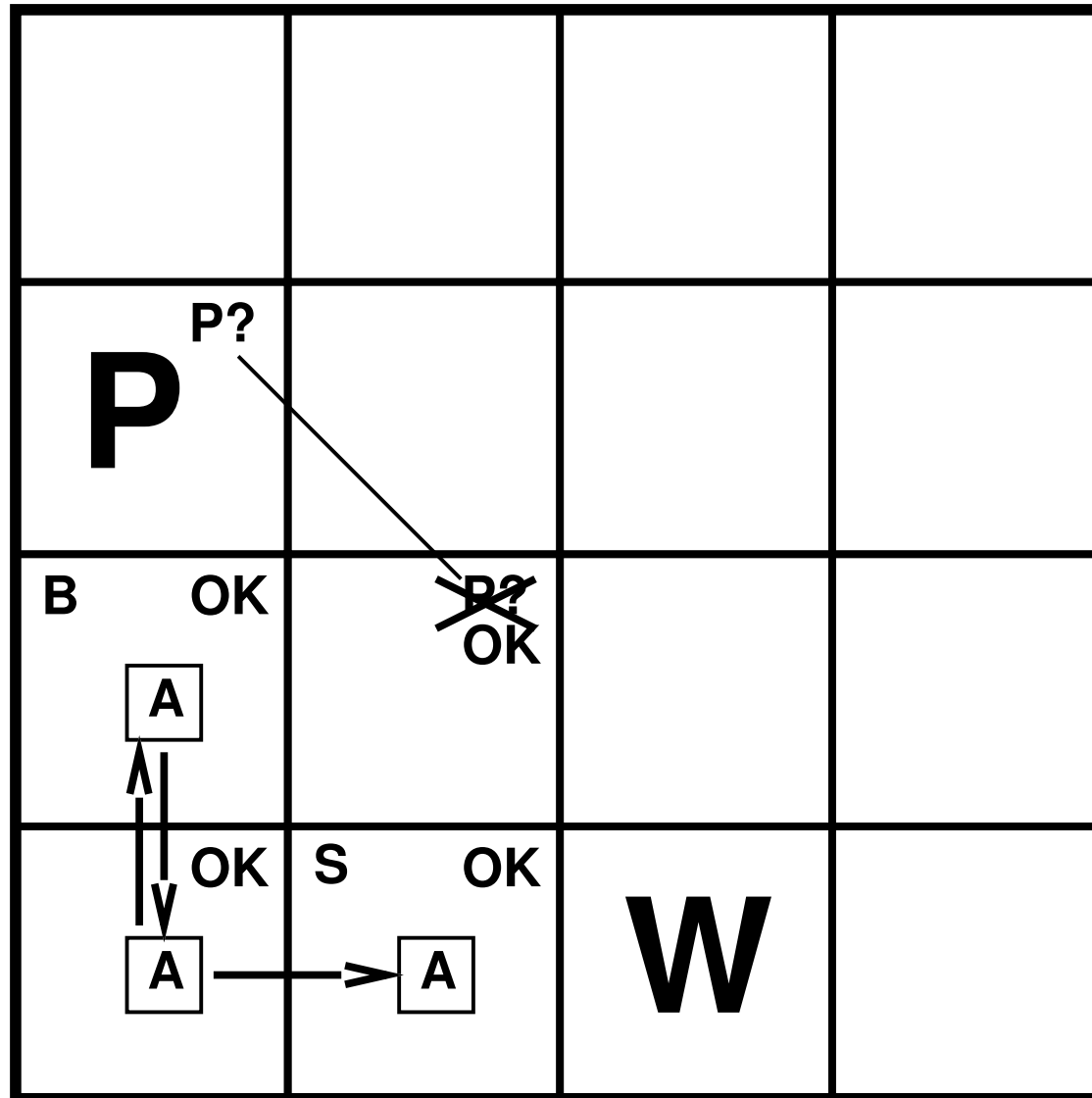
Exploring a Wumpus World



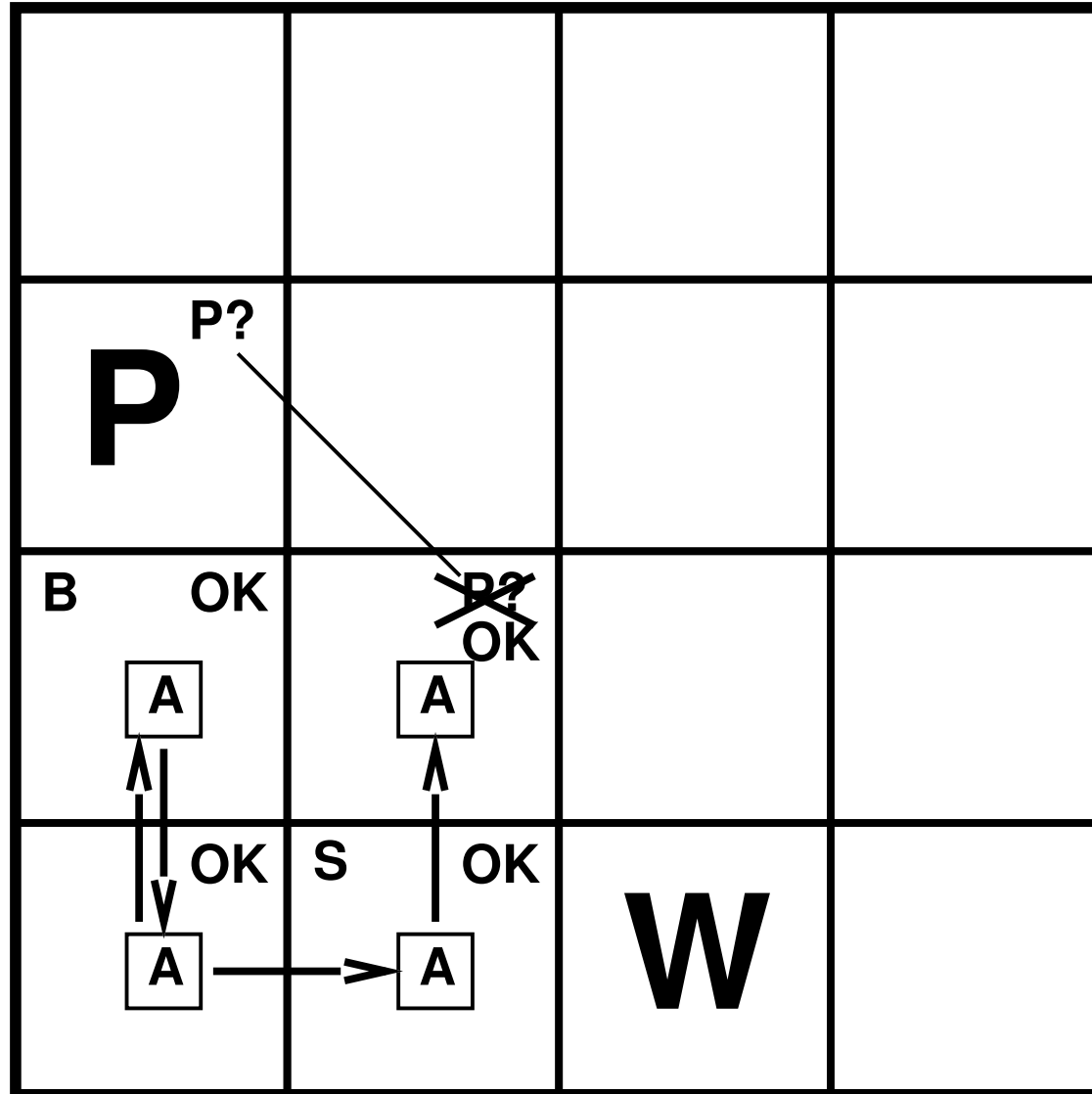
Exploring a Wumpus World



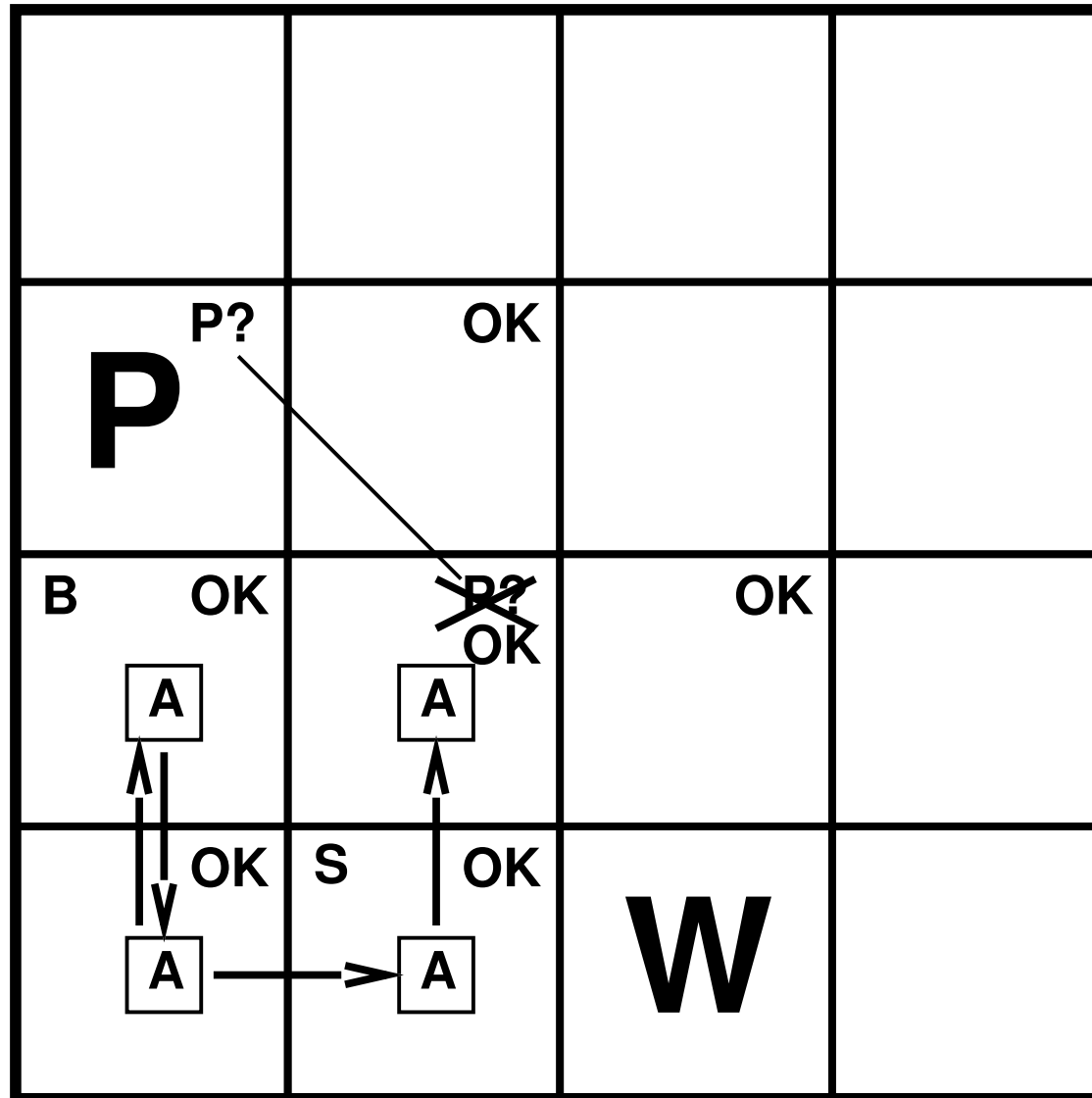
Exploring a Wumpus World



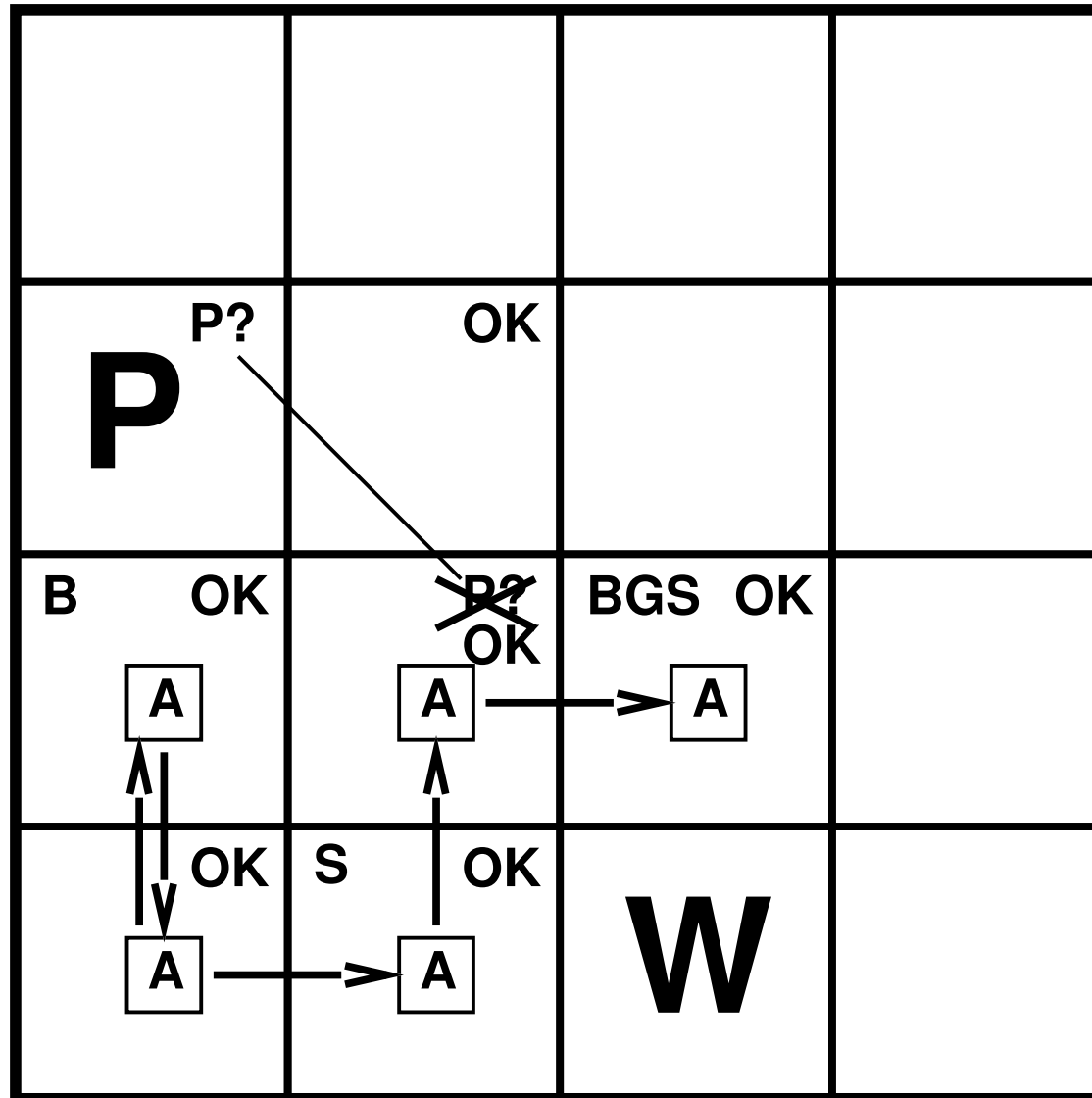
Exploring a Wumpus World



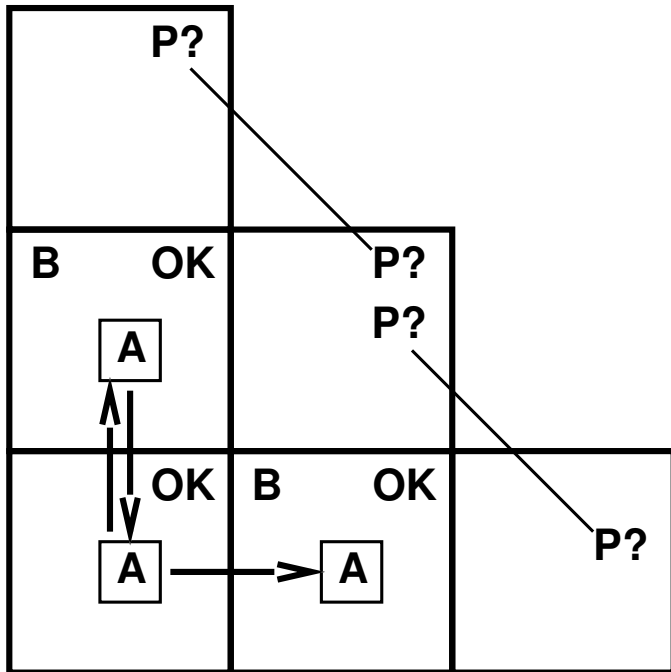
Exploring a Wumpus World



Exploring a Wumpus World



Problematic Situations

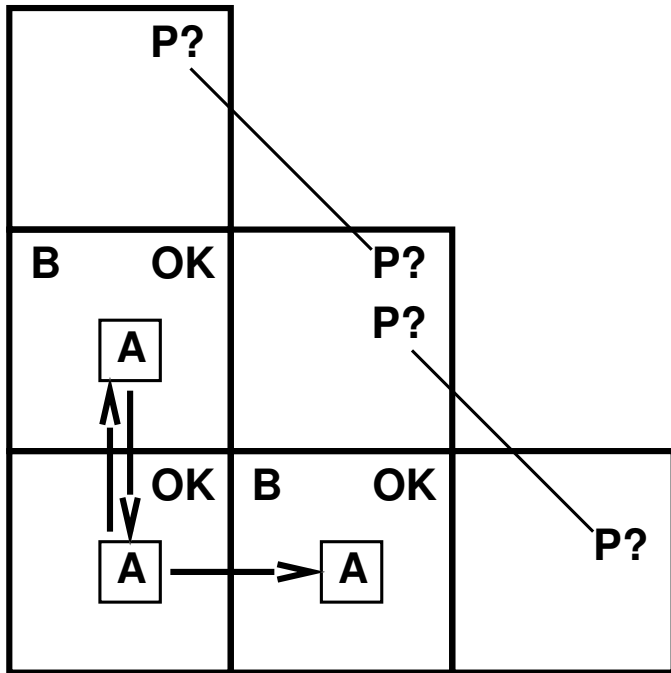


Problem

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

⇒ no safe actions

Problematic Situations



Problem

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

⇒ no safe actions

Possible solution

Assuming pits uniformly distributed:

(2,2) has pit with probability 0.86

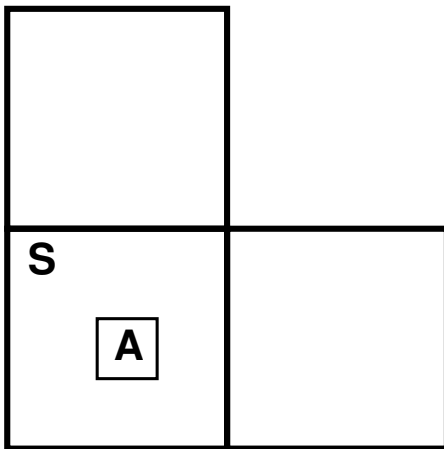
(1,3) and (3,1) have pit with probab. 0.31

Problematic Situations

Problem

Smell in (1,1)

⇒ no safe actions

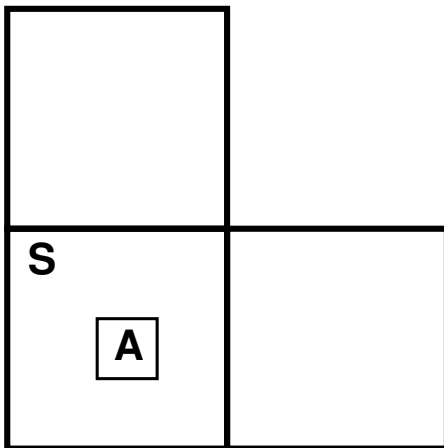


Problematic Situations

Problem

Smell in (1,1)

⇒ no safe actions



Possible solution

Strategy of coercion:

shoot straight ahead

wumpus was there ⇒ dead ⇒ safe

wumpus wasn't there ⇒ safe

Logic in General

Logics

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., defines **truth** of a sentence in a world

Example: Language of Arithmetic

Syntax

$x + 2 \geq y$ **is a sentence**

$x^2 + y >$ **is not a sentence**

Example: Language of Arithmetic

Syntax

$x + 2 \geq y$ **is a sentence**

$x^2 + y >$ **is not a sentence**

Semantics

$x + 2 \geq y$ **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 $x = 0, y = 6$**

Entailment

Definition

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

Notation

$$KB \models \alpha$$

Entailment

Definition

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

Notation

$$KB \models \alpha$$

Note

Entailment is a relationship between sentences (i.e., **syntax**)
that is based on **semantics**

Entailment

Example

The KB containing “the shirt is green” and “the shirt is striped” entails “the shirt is green or the shirt is striped”

Example

$x + y = 4$ entails $4 = x + y$

Models

Intuition

**Models are formally structured worlds,
with respect to which truth can be evaluated**

Models

Intuition

**Models are formally structured worlds,
with respect to which truth can be evaluated**

Definition

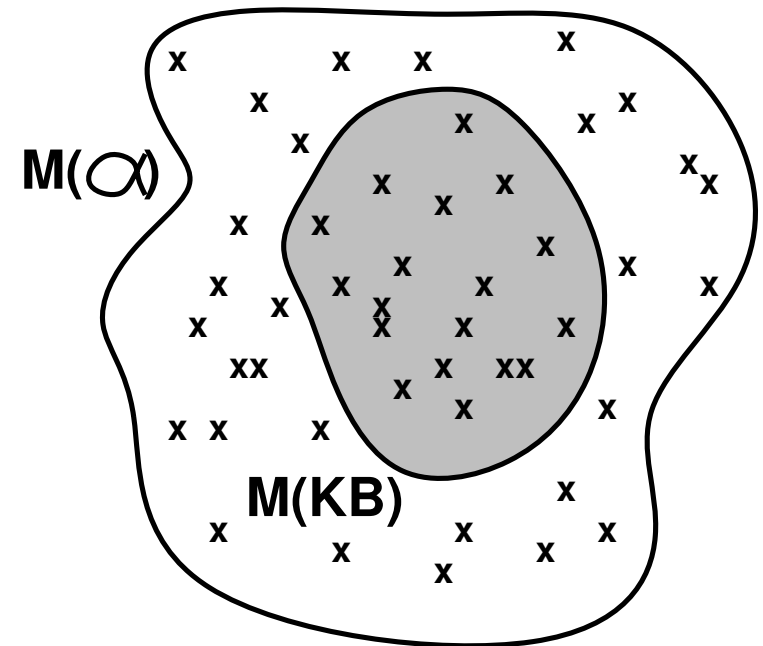
m **is a model of** a sentence α **if** α **is true in** *m*

$M(\alpha)$ **is the set of all models of** α

Note

$KB \models \alpha$ **if and only if** $M(KB) \subseteq M(\alpha)$

Models: Example



KB = The shirt is green and striped

α = The shirt is green

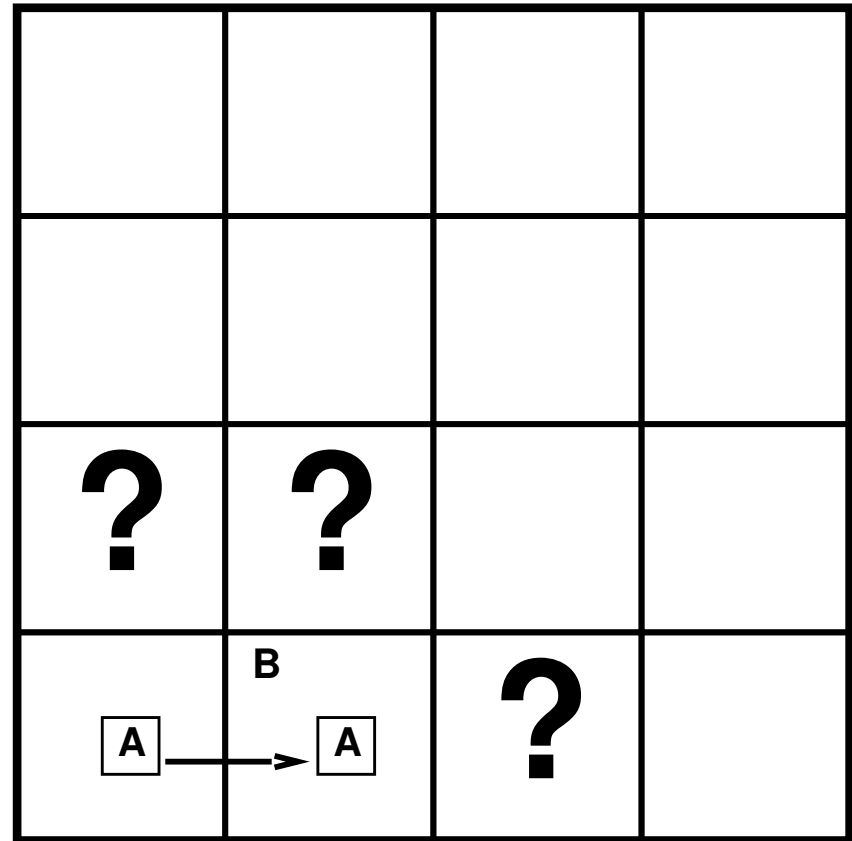
Entailment in the Wumpus World

Situation after

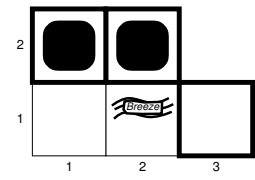
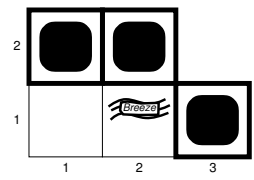
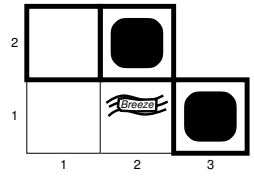
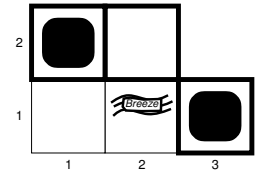
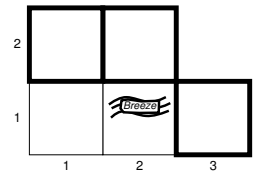
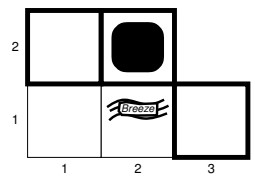
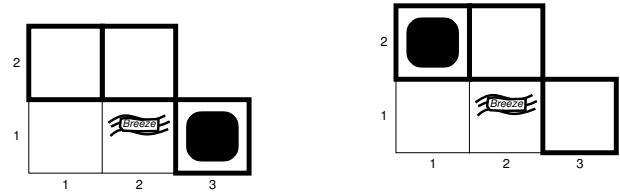
detecting nothing in [1,1],
moving right,
breeze in [2,1]

Consider possible models for “?”s
(considering only pits)

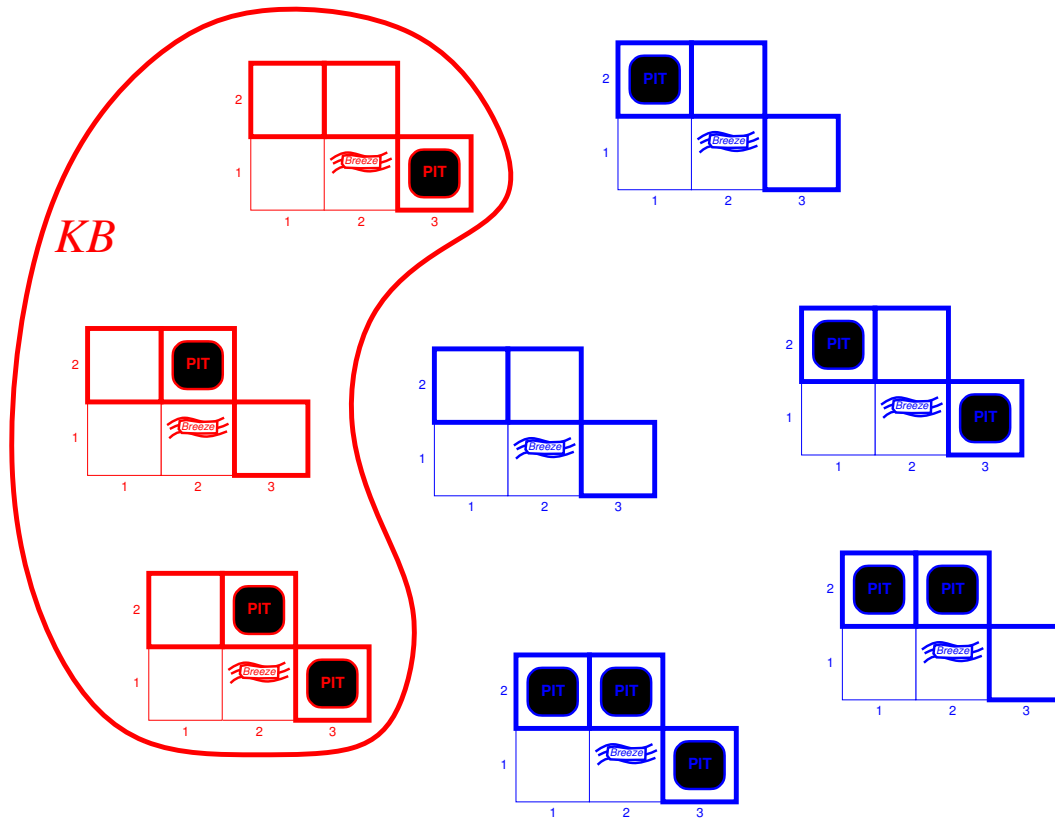
3 Boolean choices
8 possible models



Wumpus Models



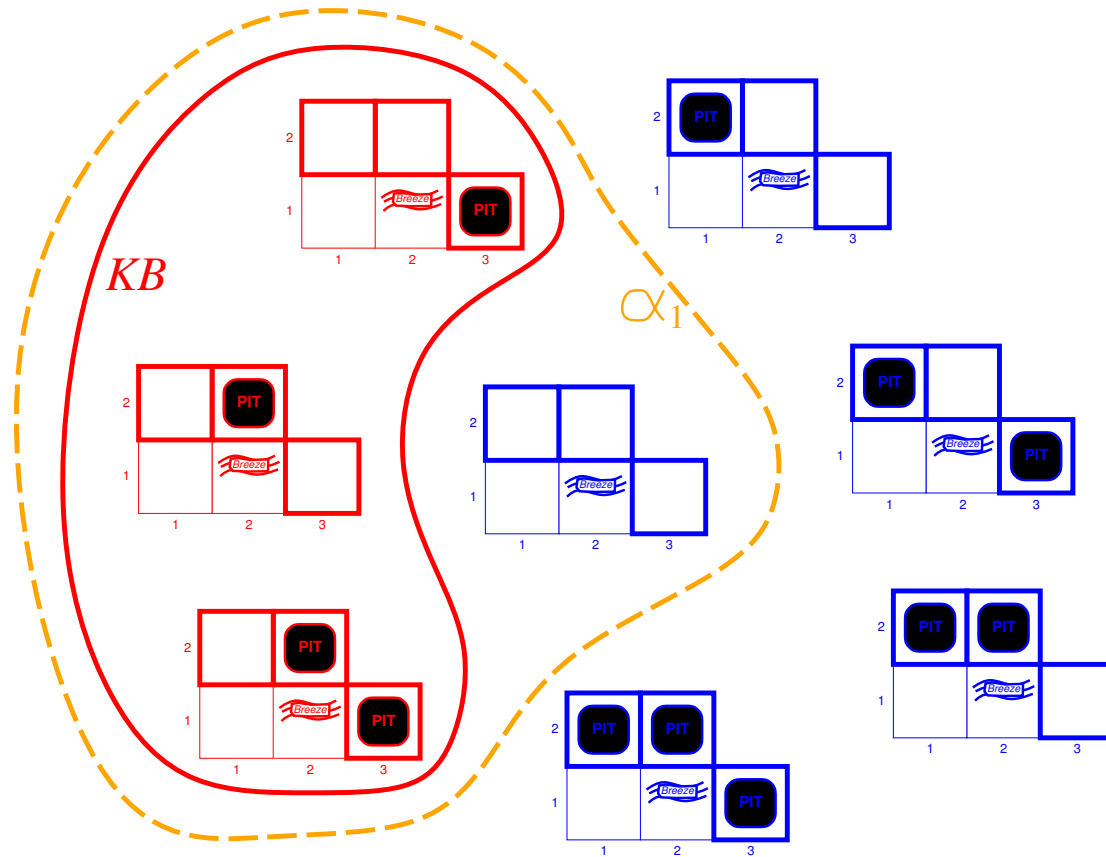
Wumpus Models



KB = wumpus-world rules + observations

Wumpus Models

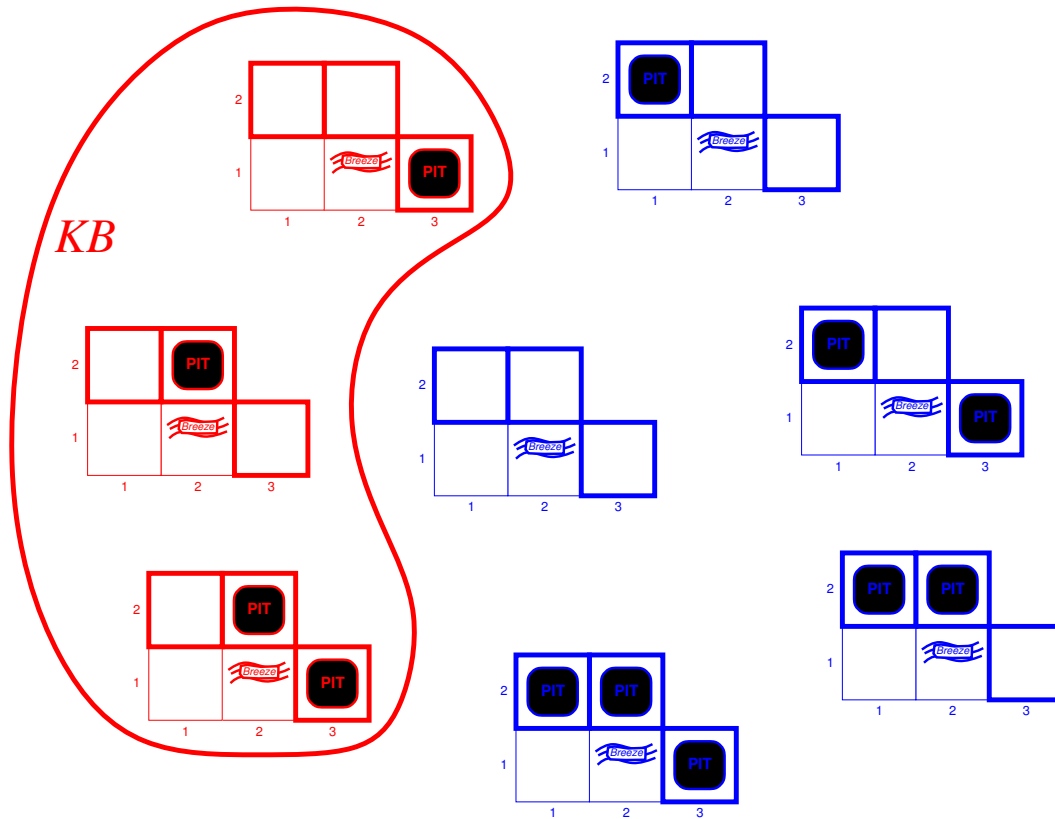
$$KB \models \alpha_1$$



KB = wumpus-world rules + observations

α_1 = “[1,2] is safe”

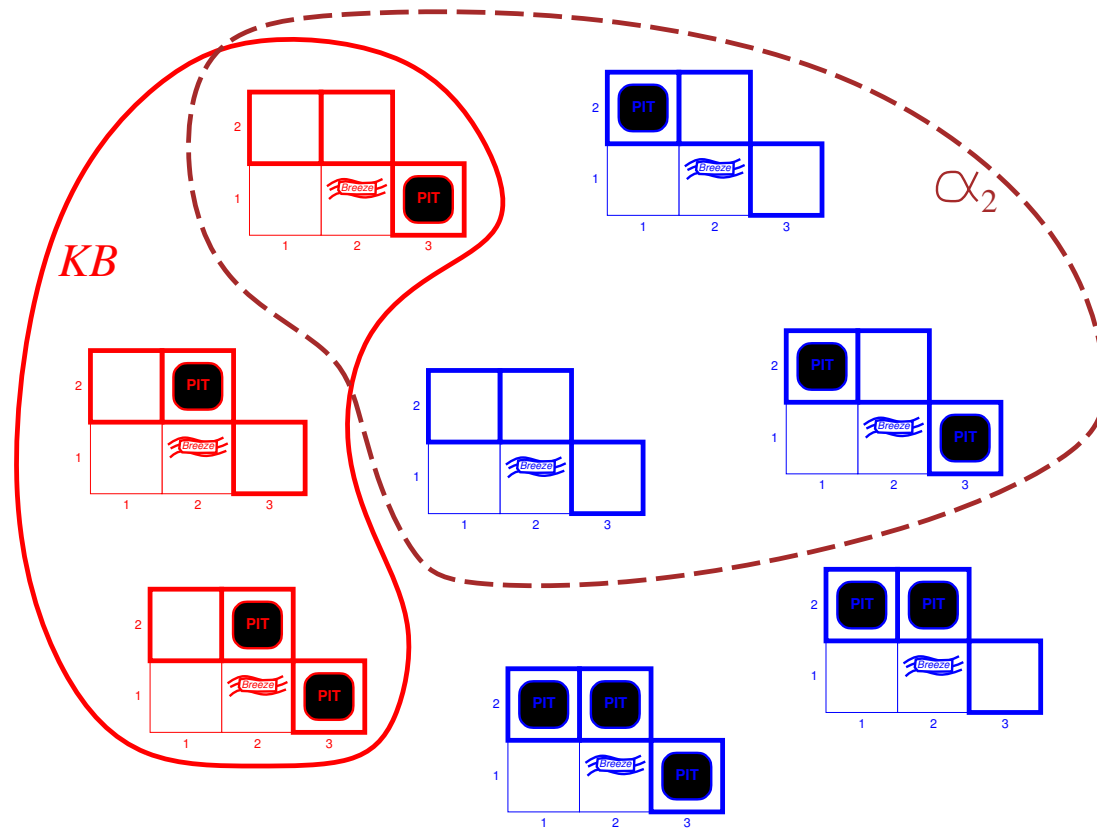
Wumpus Models



$KB = \text{wumpus-world rules} + \text{observations}$

Wumpus Models

$$KB \not\models \alpha_2$$



KB = wumpus-world rules + observations

α_2 = “[2,2] is safe”

Inference

Definition

$$KB \vdash_i \alpha$$

means

sentence α can be derived from KB by inference procedure i

Inference

Definition

$$KB \vdash_i \alpha$$

means

sentence α can be derived from KB by inference procedure i

Soundness (of i)

Whenever $KB \vdash_i \alpha$, it is also true that $KB \models \alpha$

Completeness (of i)

Whenever $KB \models \alpha$, it is also true that $KB \vdash_i \alpha$

Preview

First-order Logic

We will define a logic (first-order logic) that

- is expressive enough to say almost anything of interest, and
- for which there exists a sound and complete inference procedure.

That is, the procedure will answer any question whose answer follows from what is known by the *KB*.

Propositional Logic: Syntax

Definition

- Propositional symbols

$A, B, P_1, P_2, \textit{ShirtIsGreen}, \textbf{etc.}$

are (atomic) sentences

- If S, S_1, S_2 are sentences, then

$\neg S$	(negation)
$S_1 \wedge S_2$	(conjunction)
$S_1 \vee S_2$	(disjunction)
$S_1 \Rightarrow S_2$	(implication)
$S_1 \Leftrightarrow S_2$	(equivalence)

are sentences

Propositional logic: Semantics

Propositional Models

Each model specifies true/false for each proposition symbol

Propositional logic: Semantics

Propositional Models

Each model specifies true/false for each proposition symbol

Example

<i>A</i>	<i>B</i>	<i>C</i>
<i>true</i>	<i>true</i>	<i>false</i>

(For three symbols, there are
8 possible models)

Propositional logic: Semantics

Propositional Models

Each model specifies true/false for each proposition symbol

Example

A B C
true *true* *false*

(For three symbols, there are
8 possible models)

Rules for evaluating truth with respect to a model

$\neg S$ 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 and S_2 have the same truth value

Truth Tables for Connectives

A	B	$\neg A$	$A \wedge B$	$A \vee B$	$A \Rightarrow B$	$A \Leftrightarrow B$
<i>false</i>	<i>false</i>	<i>true</i>	<i>false</i>	<i>false</i>	<i>true</i>	<i>true</i>
<i>false</i>	<i>true</i>	<i>true</i>	<i>false</i>	<i>true</i>	<i>true</i>	<i>false</i>
<i>true</i>	<i>false</i>	<i>false</i>	<i>false</i>	<i>true</i>	<i>false</i>	<i>false</i>
<i>true</i>	<i>true</i>	<i>false</i>	<i>true</i>	<i>true</i>	<i>true</i>	<i>true</i>

Wumpus World Sentences

Propositional symbols

$P_{i,j}$ means: “there is a pit in $[i, j]$ ”

$B_{i,j}$ means: “there is a breeze in $[i, j]$ ”

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

Wumpus World Sentences

Propositional symbols

$P_{i,j}$ means: “there is a pit in $[i, j]$ ”

$B_{i,j}$ means: “there is a breeze in $[i, j]$ ”

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

Sentences

“Pits cause breezes in adjacent squares”

$$P_{1,2} \Rightarrow (B_{1,1} \wedge B_{1,3} \wedge B_{2,2})$$

“A square is breezy **if and only if** there is an adjacent pit”

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

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

Propositional Inference: Enumeration Method

Example

$$\alpha = A \vee B \quad KB = (A \vee C) \wedge (B \vee \neg C)$$

Checking that $KB \models \alpha$

<i>A</i>	<i>B</i>	<i>C</i>	$A \vee C$	$B \vee \neg C$	<i>KB</i>	α
<i>false</i>	<i>false</i>	<i>false</i>				
<i>false</i>	<i>false</i>	<i>true</i>				
<i>false</i>	<i>true</i>	<i>false</i>				
<i>false</i>	<i>true</i>	<i>true</i>				
<i>true</i>	<i>false</i>	<i>false</i>				
<i>true</i>	<i>false</i>	<i>true</i>				
<i>true</i>	<i>true</i>	<i>false</i>				
<i>true</i>	<i>true</i>	<i>true</i>				

Propositional Inference: Enumeration Method

Example

$$\alpha = A \vee B \quad KB = (A \vee C) \wedge (B \vee \neg C)$$

Checking that $KB \models \alpha$

<i>A</i>	<i>B</i>	<i>C</i>	$A \vee C$	$B \vee \neg C$	<i>KB</i>	α
<i>false</i>	<i>false</i>	<i>false</i>	<i>false</i>			
<i>false</i>	<i>false</i>	<i>true</i>	<i>true</i>			
<i>false</i>	<i>true</i>	<i>false</i>	<i>false</i>			
<i>false</i>	<i>true</i>	<i>true</i>	<i>true</i>			
<i>true</i>	<i>false</i>	<i>false</i>	<i>true</i>			
<i>true</i>	<i>false</i>	<i>true</i>	<i>true</i>			
<i>true</i>	<i>true</i>	<i>false</i>	<i>true</i>			
<i>true</i>	<i>true</i>	<i>true</i>	<i>true</i>			

Propositional Inference: Enumeration Method

Example

$$\alpha = A \vee B \quad KB = (A \vee C) \wedge (B \vee \neg C)$$

Checking that $KB \models \alpha$

<i>A</i>	<i>B</i>	<i>C</i>	$A \vee C$	$B \vee \neg C$	<i>KB</i>	α
<i>false</i>	<i>false</i>	<i>false</i>	<i>false</i>	<i>true</i>		
<i>false</i>	<i>false</i>	<i>true</i>	<i>true</i>	<i>false</i>		
<i>false</i>	<i>true</i>	<i>false</i>	<i>false</i>	<i>true</i>		
<i>false</i>	<i>true</i>	<i>true</i>	<i>true</i>	<i>true</i>		
<i>true</i>	<i>false</i>	<i>false</i>	<i>true</i>	<i>true</i>		
<i>true</i>	<i>false</i>	<i>true</i>	<i>true</i>	<i>false</i>		
<i>true</i>	<i>true</i>	<i>false</i>	<i>true</i>	<i>true</i>		
<i>true</i>	<i>true</i>	<i>true</i>	<i>true</i>	<i>true</i>		

Propositional Inference: Enumeration Method

Example

$$\alpha = A \vee B \quad KB = (A \vee C) \wedge (B \vee \neg C)$$

Checking that $KB \models \alpha$

<i>A</i>	<i>B</i>	<i>C</i>	$A \vee C$	$B \vee \neg C$	<i>KB</i>	α
<i>false</i>	<i>false</i>	<i>false</i>	<i>false</i>	<i>true</i>	<i>false</i>	
<i>false</i>	<i>false</i>	<i>true</i>	<i>true</i>	<i>false</i>	<i>false</i>	
<i>false</i>	<i>true</i>	<i>false</i>	<i>false</i>	<i>true</i>	<i>false</i>	
<i>false</i>	<i>true</i>	<i>true</i>	<i>true</i>	<i>true</i>	<i>true</i>	
<i>true</i>	<i>false</i>	<i>false</i>	<i>true</i>	<i>true</i>	<i>true</i>	
<i>true</i>	<i>false</i>	<i>true</i>	<i>true</i>	<i>false</i>	<i>false</i>	
<i>true</i>	<i>true</i>	<i>false</i>	<i>true</i>	<i>true</i>	<i>true</i>	
<i>true</i>	<i>true</i>	<i>true</i>	<i>true</i>	<i>true</i>	<i>true</i>	

Propositional Inference: Enumeration Method

Example

$$\alpha = A \vee B \quad KB = (A \vee C) \wedge (B \vee \neg C)$$

Checking that $KB \models \alpha$

<i>A</i>	<i>B</i>	<i>C</i>	$A \vee C$	$B \vee \neg C$	<i>KB</i>	α
<i>false</i>	<i>false</i>	<i>false</i>	<i>false</i>	<i>true</i>	<i>false</i>	<i>false</i>
<i>false</i>	<i>false</i>	<i>true</i>	<i>true</i>	<i>false</i>	<i>false</i>	<i>false</i>
<i>false</i>	<i>true</i>	<i>false</i>	<i>false</i>	<i>true</i>	<i>false</i>	<i>true</i>
<i>false</i>	<i>true</i>	<i>true</i>	<i>true</i>	<i>true</i>	<i>true</i>	<i>true</i>
<i>true</i>	<i>false</i>	<i>false</i>	<i>true</i>	<i>true</i>	<i>true</i>	<i>true</i>
<i>true</i>	<i>false</i>	<i>true</i>	<i>true</i>	<i>false</i>	<i>false</i>	<i>true</i>
<i>true</i>	<i>true</i>	<i>false</i>	<i>true</i>	<i>true</i>	<i>true</i>	<i>true</i>
<i>true</i>	<i>true</i>	<i>true</i>	<i>true</i>	<i>true</i>	<i>true</i>	<i>true</i>

Propositional Inference: Enumeration Method

Example

$$\alpha = A \vee B \quad KB = (A \vee C) \wedge (B \vee \neg C)$$

Checking that $KB \models \alpha$

<i>A</i>	<i>B</i>	<i>C</i>	$A \vee C$	$B \vee \neg C$	<i>KB</i>	α
<i>false</i>	<i>false</i>	<i>false</i>	<i>false</i>	<i>true</i>	<i>false</i>	<i>false</i>
<i>false</i>	<i>false</i>	<i>true</i>	<i>true</i>	<i>false</i>	<i>false</i>	<i>false</i>
<i>false</i>	<i>true</i>	<i>false</i>	<i>false</i>	<i>true</i>	<i>false</i>	<i>true</i>
<i>false</i>	<i>true</i>	<i>true</i>	<i>true</i>	<i>true</i>	<i>true</i>	<i>true</i>
<i>true</i>	<i>false</i>	<i>false</i>	<i>true</i>	<i>true</i>	<i>true</i>	<i>true</i>
<i>true</i>	<i>false</i>	<i>true</i>	<i>true</i>	<i>false</i>	<i>false</i>	<i>true</i>
<i>true</i>	<i>true</i>	<i>false</i>	<i>true</i>	<i>true</i>	<i>true</i>	<i>true</i>
<i>true</i>	<i>true</i>	<i>true</i>	<i>true</i>	<i>true</i>	<i>true</i>	<i>true</i>

Propositional Inference: Enumeration Method

Example

$$\alpha = A \vee B \quad KB = (A \vee C) \wedge (B \vee \neg C)$$

Checking that $KB \models \alpha$

<i>A</i>	<i>B</i>	<i>C</i>	$A \vee C$	$B \vee \neg C$	<i>KB</i>	α
<i>false</i>	<i>false</i>	<i>false</i>	<i>false</i>	<i>true</i>	<i>false</i>	<i>false</i>
<i>false</i>	<i>false</i>	<i>true</i>	<i>true</i>	<i>false</i>	<i>false</i>	<i>false</i>
<i>false</i>	<i>true</i>	<i>false</i>	<i>false</i>	<i>true</i>	<i>false</i>	<i>true</i>
<i>false</i>	<i>true</i>	<i>true</i>	<i>true</i>	<i>true</i>	<i>true</i>	<i>true</i>
<i>true</i>	<i>false</i>	<i>false</i>	<i>true</i>	<i>true</i>	<i>true</i>	<i>true</i>
<i>true</i>	<i>false</i>	<i>true</i>	<i>true</i>	<i>false</i>	<i>false</i>	<i>true</i>
<i>true</i>	<i>true</i>	<i>false</i>	<i>true</i>	<i>true</i>	<i>true</i>	<i>true</i>
<i>true</i>	<i>true</i>	<i>true</i>	<i>true</i>	<i>true</i>	<i>true</i>	<i>true</i>

Note

Table has 2^n rows for n symbols

Logical Equivalence

Definition

Two sentences are **logically equivalent**, denoted by

$$\alpha \equiv \beta$$

iff they are true in the same models, i.e., iff:

$$\alpha \models \beta \quad \text{and} \quad \beta \models \alpha$$

Logical Equivalence

Definition

Two sentences are **logically equivalent**, denoted by

$$\alpha \equiv \beta$$

iff they are true in the same models, i.e., iff:

$$\alpha \models \beta \quad \text{and} \quad \beta \models \alpha$$

Example

$$(A \Rightarrow B) \equiv (\neg B \Rightarrow \neg A) \quad \text{(contraposition)}$$

Logical Equivalence

Theorem

If

- $\alpha \equiv \beta$
- γ is the result of replacing a subformula α of δ by β ,

then $\gamma \equiv \delta$

Logical Equivalence

Theorem

If

- $\alpha \equiv \beta$
- γ is the result of replacing a subformula α of δ by β ,

then $\gamma \equiv \delta$

Example

$$A \vee B \equiv B \vee A$$

implies

$$(C \wedge (A \vee B)) \Rightarrow D \equiv (C \wedge (B \vee A)) \Rightarrow D$$

Important Equivalences

$(\alpha \wedge \beta) \equiv (\beta \wedge \alpha)$	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
$(\alpha \wedge \alpha) \equiv \alpha$	idempotence for \wedge
$(\alpha \vee \alpha) \equiv \alpha$	idempotence for \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
$(\alpha \Leftrightarrow \beta) \equiv ((\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha))$	equivalence elimination
$\neg(\alpha \wedge \beta) \equiv (\neg\alpha \vee \neg\beta)$	de Morgan's rules
$\neg(\alpha \vee \beta) \equiv (\neg\alpha \wedge \neg\beta)$	de Morgan's rules
$(\alpha \wedge (\beta \vee \gamma)) \equiv ((\alpha \wedge \beta) \vee (\alpha \wedge \gamma))$	distributivity of \wedge over \vee
$(\alpha \vee (\beta \wedge \gamma)) \equiv ((\alpha \vee \beta) \wedge (\alpha \vee \gamma))$	distributivity of \vee over \wedge

The Logical Constants *true* and *false*

Semantics

true evaluates to *true* in all models

false evaluates to *false* in all models

The Logical Constants *true* and *false*

Semantics

true evaluates to *true* in all models

false evaluates to *false* in all models

Important equivalences with *true* and *false*

$$(\alpha \wedge \neg\alpha) \equiv \textit{false}$$

$$(\alpha \vee \neg\alpha) \equiv \textit{true} \quad \text{tertium non datur}$$

$$(\alpha \wedge \textit{true}) \equiv \alpha$$

$$(\alpha \wedge \textit{false}) \equiv \textit{false}$$

$$(\alpha \vee \textit{true}) \equiv \textit{true}$$

$$(\alpha \vee \textit{false}) \equiv \alpha$$

Validity

Definition

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

Examples

$$A \vee \neg A, \quad A \Rightarrow A, \quad (A \wedge (A \Rightarrow B)) \Rightarrow B$$

Validity

Definition

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

Examples

$$A \vee \neg A, \quad A \Rightarrow A, \quad (A \wedge (A \Rightarrow B)) \Rightarrow B$$

Deduction Theorem (connects inference and validity)

$$KB \models \alpha \quad \text{if and only if} \quad KB \Rightarrow \alpha \quad \text{is valid}$$

Satisfiability

Definition

A sentence is **satisfiable** if it is true in **some** model

Examples

$$A \vee B, \quad A, \quad A \wedge (A \Rightarrow B)$$

Satisfiability

Definition

A sentence is **satisfiable** if it is true in **some** model

Examples

$$A \vee B, \quad A, \quad A \wedge (A \Rightarrow B)$$

Definition

A sentence is **unsatisfiable** if it is true in **no** models,
i.e., if it is not satisfiable

Example

$$A \wedge \neg A$$

Satisfiability

Theorem (connects validity and unsatisfiability)

α is valid if and only if $\neg\alpha$ is unsatisfiable

Satisfiability

Theorem (connects validity and unsatisfiability)

α is valid if and only if $\neg\alpha$ is unsatisfiable

Theorem (connects inference and unsatisfiability)

$KB \models \alpha$ if and only if $(KB \wedge \neg\alpha)$ is unsatisfiable

Satisfiability

Theorem (connects validity and unsatisfiability)

α is valid if and only if $\neg\alpha$ is unsatisfiable

Theorem (connects inference and unsatisfiability)

$KB \models \alpha$ if and only if $(KB \wedge \neg\alpha)$ is unsatisfiable

Note

Validity and inference can be proved by **reductio ad absurdum**

Two Kinds of Proof Methods

1. Application of inference rules

Legitimate (sound) generation of new sentences from old

Construction of / search for a proof

(proof = sequence of inference rule applications)

Two Kinds of Proof Methods

1. Application of inference rules

Legitimate (sound) generation of new sentences from old

Construction of / search for a proof

(proof = sequence of inference rule applications)

Properties

Typically requires translation of sentences into a normal form

Two Kinds of Proof Methods

1. Application of inference rules

Legitimate (sound) generation of new sentences from old

Construction of / search for a proof

(proof = sequence of inference rule applications)

Properties

Typically requires translation of sentences into a normal form

Different kinds

Tableau calculus, resolution, forward/backward chaining, ...

Two Kinds of Proof Methods

2. Model checking

Construction of / search for a satisfying model

Two Kinds of Proof Methods

2. Model checking

Construction of / search for a satisfying model

Different kinds

- Truth table enumeration (always exponential number of symbols)
- Improved backtracking search for models
e.g.: Davis-Putnam-Logemann-Loveland
- Heuristic search in model space (sound but incomplete)
e.g.: hill-climbing algorithms

Normal Forms

Literal

A literal is

- **an atomic sentence (propositional symbol), or**
- **the negation of an atomic sentence**

Normal Forms

Literal

A literal is

- an atomic sentence (propositional symbol), or
- the negation of an atomic sentence

Clause

A disjunction of literals

Normal Forms

Literal

A literal is

- an atomic sentence (propositional symbol), or
- the negation of an atomic sentence

Clause

A disjunction of literals

Conjunctive Normal Form (CNF)

A conjunction of disjunctions of literals,
i.e., a conjunction of clauses

Example

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

Resolution

Inference rule

$$\begin{array}{c} P_1 \vee \dots \vee P_{i-1} \vee Q \vee P_{i+1} \vee \dots \vee P_k \qquad R_1 \vee \dots \vee R_{j-1} \vee \neg Q \vee R_{j+1} \vee \dots \vee R_n \\ \hline P_1 \vee \dots \vee P_{i-1} \vee P_{i+1} \vee \dots \vee P_k \vee R_1 \vee \dots \vee R_{j-1} \vee R_{j+1} \vee \dots \vee R_n \end{array}$$

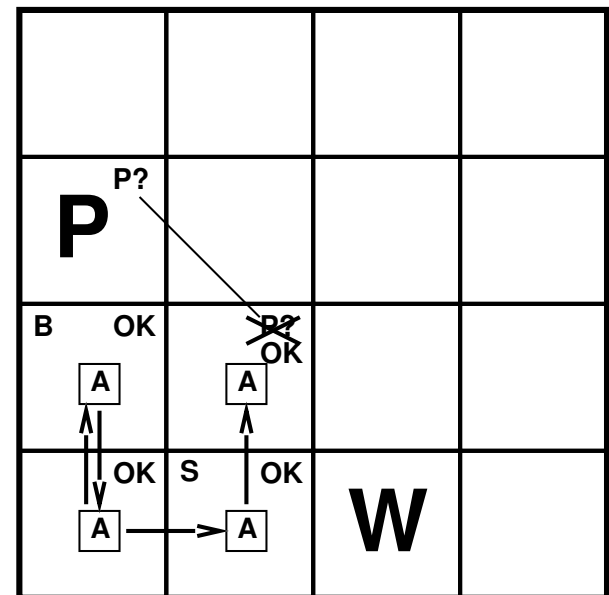
Resolution

Inference rule

$$\frac{P_1 \vee \dots \vee P_{i-1} \vee Q \vee P_{i+1} \vee \dots \vee P_k \quad R_1 \vee \dots \vee R_{j-1} \vee \neg Q \vee R_{j+1} \vee \dots \vee R_n}{P_1 \vee \dots \vee P_{i-1} \vee P_{i+1} \vee \dots \vee P_k \vee R_1 \vee \dots \vee R_{j-1} \vee R_{j+1} \vee \dots \vee R_n}$$

Example

$$\frac{P_{1,3} \vee P_{2,2} \quad \neg P_{2,2}}{P_{1,3}}$$



Resolution

Correctness theorem

Resolution is sound and complete for propositional logic,

i.e., given a formula α in CNF (conjunction of clauses):

α is unsatisfiable

iff

the empty clause can be derived from α with resolution

Conversion to CNF

0. Given

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

Conversion to CNF

0. Given

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

1. Eliminate \Leftrightarrow , replacing $\alpha \equiv \beta$ with $(\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)$

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

Conversion to CNF

0. Given

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

1. Eliminate \Leftrightarrow , replacing $\alpha \equiv \beta$ with $(\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)$

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

2. Eliminate \Rightarrow , replacing $\alpha \Rightarrow \beta$ with $\neg\alpha \vee \beta$

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

Conversion to CNF

0. Given

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

1. Eliminate \Leftrightarrow , replacing $\alpha \equiv \beta$ with $(\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)$

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

2. Eliminate \Rightarrow , replacing $\alpha \Rightarrow \beta$ with $\neg\alpha \vee \beta$

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

3. Move \neg inwards using de Morgan's rules (and double-negation)

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

Conversion to CNF

0. Given

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

1. Eliminate \Leftrightarrow , replacing $\alpha \equiv \beta$ with $(\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)$

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

2. Eliminate \Rightarrow , replacing $\alpha \Rightarrow \beta$ with $\neg\alpha \vee \beta$

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

3. Move \neg inwards using de Morgan's rules (and double-negation)

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

4. Apply distributivity law (\vee over \wedge) and flatten

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

Resolution Example

Given

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

$$\alpha = \neg P_{1,2}$$

Resolution Example

Given

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

$$\alpha = \neg P_{1,2}$$

Resolution proof for $KB \models \alpha$

Derive empty clause \square from $KB \wedge \neg\alpha$ in CNF

Resolution Example

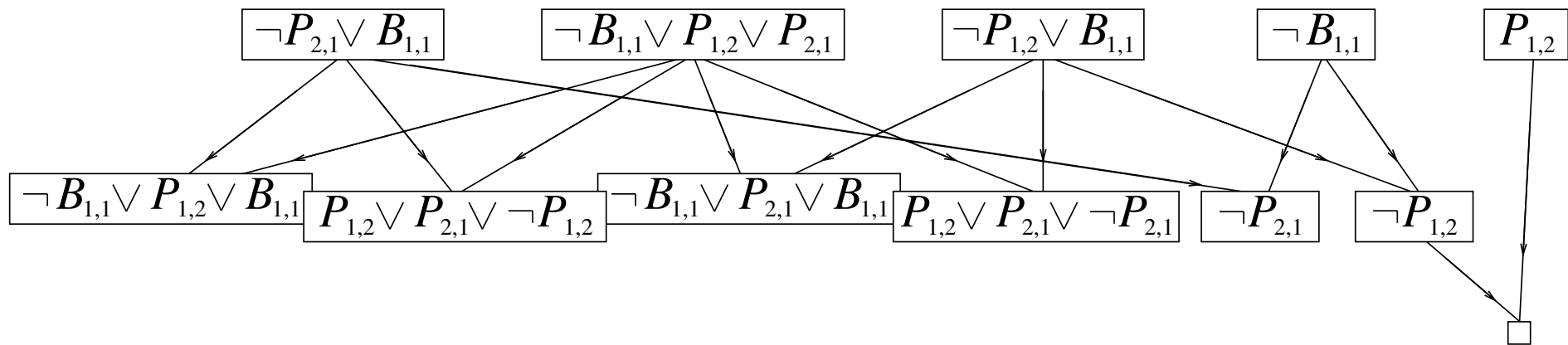
Given

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

$$\alpha = \neg P_{1,2}$$

Resolution proof for $KB \models \alpha$

Derive empty clause \square from $KB \wedge \neg\alpha$ in CNF



Summary

- **Logical agents apply inference to a knowledge base to derive new information and make decisions**

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 w.r.t. 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

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 w.r.t. 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.**

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 w.r.t. 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.**
- **Resolution is sound and complete for propositional logic**

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 w.r.t. 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.**
- **Resolution is sound and complete for propositional logic**
- **Propositional logic lacks expressive power**