CSE 411: Artificial Intelligence (Elective Course #6)

400 Level, Mechatronics Engineering
2nd Term 2016/2017, Lecture #8

Hazem Shehata

Dept. of Computer & Systems Engineering
Zagazig University

April 24^{th}, 2017

Credits to Dr. Mohamed El Abd for the slides

Adminstrivia

Notes

- Midterm:
  - Date: Wednesday, Apr. 26, 2017.
  - Time: 1:00pm - 2:00pm.
  - Scope: lectures $1 \rightarrow 6$.

Course Info:

- Website: http://hshehata.github.io/courses/zu/cse411/
- Office hours: Sunday 11:30am - 12:30pm

Outline

Outline

1. **Logical Agents**
   - Knowledge-based agents
   - Wumpus world
   - Logic
   - Propositional logic

2. **Requirements & Reading Material**

Knowledge-based agents

Introduction

- ***Knowledge-based agents*** are agents that can store knowledge about the environment and use this knowledge to deduce new facts.

Knowledge-based agents

Introduction

- ***Knowledge-based agents*** are agents that can store knowledge about the environment and use this knowledge to deduce new facts.
- They are specially useful since they:
  - Infer hidden information, needed in partially-observable environments.

Knowledge-based agents

Introduction

- ***Knowledge-based agents*** are agents that can store knowledge about the environment and use this knowledge to deduce new facts.
- They are specially useful since they:
  - Infer hidden information, needed in partially-observable environments.
  - Flexible:
    - Can learn new knowledge about the environment.
    - Adapt to environmental changes by updating relevant knowledge.

Knowledge-based agents

Introduction

- Two important aspects are:

Knowledge-based agents

Introduction

- Two important aspects are:
  - Storing the knowledge or representing it using a ***knowledge-base***.

Knowledge-based agents

Introduction

- Two important aspects are:
  - Storing the knowledge or representing it using a ***knowledge-base***.
  - Deducing new facts or ***reasoning*** about the possible actions.

Knowledge-based agents

Introduction

- Two important aspects are:
  - Storing the knowledge or representing it using a ***knowledge-base***.
  - Deducing new facts or ***reasoning*** about the possible actions.
- ***Reasoning*** is also known as ***inferencing***.

Knowledge-based agents

Knowledge base

- The knowledge base (KB) is the central component of knowledge-based agents.

Knowledge-based agents

Knowledge base

- The knowledge base (KB) is the central component of knowledge-based agents.
- The KB holds assertions about the environment:
  - Facts.
  - Rules.

Knowledge-based agents

Knowledge base

- The knowledge base (KB) is the central component of knowledge-based agents.
- The KB holds assertions about the environment:
  - Facts.
  - Rules.
- These assertions are stored as a set of ***sentences***.

Knowledge-based agents

Knowledge base

- The knowledge base (KB) is the central component of knowledge-based agents.
- The KB holds assertions about the environment:
  - Facts.
  - Rules.
- These assertions are stored as a set of ***sentences***.
- Sentences are expressed using a ***knowledge representation language***.

Knowledge-based agents

Knowledge base

- The two main operations carried by the knowledge-based agents are:

Knowledge-based agents

Knowledge base

- The two main operations carried by the knowledge-based agents are:
  - Add new sentences to the knowledge base.

Knowledge-based agents

Knowledge base

- The two main operations carried by the knowledge-based agents are:
  - Add new sentences to the knowledge base.
  - Query what is known.

Knowledge-based agents

Knowledge base

- The two main operations carried by the knowledge-based agents are:
  - Add new sentences to the knowledge base.
  - Query what is known.
- These operations are known as ***TELL*** and ***ASK*** operations.

Knowledge-based agents

Knowledge base

- The two main operations carried by the knowledge-based agents are:
  - Add new sentences to the knowledge base.
  - Query what is known.
- These operations are known as ***TELL*** and ***ASK*** operations.
- When the agent ***ASKs*** a question, the answer should follow from what the KB has been ***TOLD*** (or rather ***TELLed***) before.

Knowledge-based agents

Knowledge base

- The two main operations carried by the knowledge-based agents are:
  - Add new sentences to the knowledge base.
  - Query what is known.
- These operations are known as ***TELL*** and ***ASK*** operations.
- When the agent ***ASKs*** a question, the answer should follow from what the KB has been ***TOLD*** (or rather ***TELLed***) before.
- Inference might be used in both ***ASK*** and ***TELL*** operations.

Knowledge-based agents

Algorithm

An algorithm for a generic knowledge-based agent:

funcion 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*

Knowledge-based agents

Algorithm

An algorithm for a generic knowledge-based agent:

```
funcion 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
```

TELLs the KB what it senses

Knowledge-based agents

Algorithm

An algorithm for a generic knowledge-based agent:

```
funcion 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
```

TELLs the KB what it senses

ASKs the KB about which actions to take

Knowledge-based agents

Algorithm

An algorithm for a generic knowledge-based agent:

funcion 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*

TELLs the KB what it senses

ASKs the KB about which actions to take

TELLs the KB the action taken

Knowledge-based agents

Knowledge base

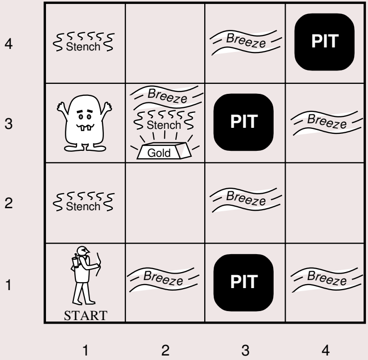
In general, an agent must be able to:

- Represent states, actions, *etc*.
- Incorporate new percepts.
- Update internal representations of the world.
- Deduce hidden properties of the world.
- Deduce appropriate actions.

Wumpus world

Example: wumpus world - PEAS

Performance measure:

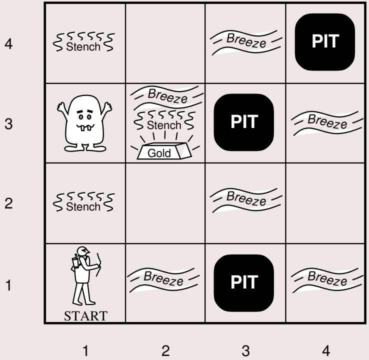


Wumpus world

Example: wumpus world - PEAS

Performance measure:

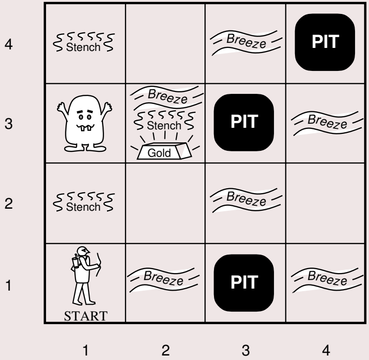
- gold +1000.
- death -1000
- -1 per step.
- -10 for using the arrow.



Wumpus world

Example: wumpus world - PEAS

Environment:

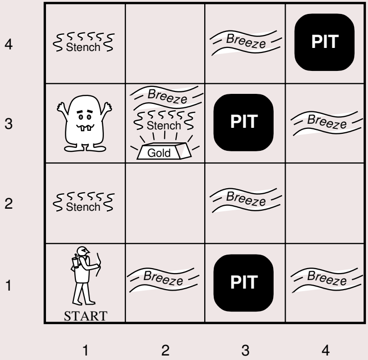


Wumpus world

Example: wumpus world - PEAS

Environment:

- Squares adjacent to wumpus are smelly.

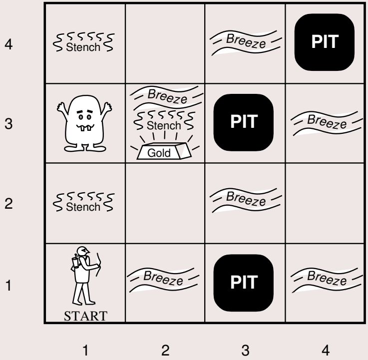


Wumpus world

Example: wumpus world - PEAS

Environment:

- Squares adjacent to wumpus are smelly.
- Squares adjacent to pit are breezy.

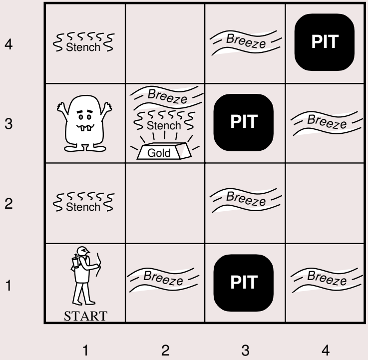


Wumpus world

Example: wumpus world - PEAS

Environment:

- Squares adjacent to wumpus are smelly.
- Squares adjacent to pit are breezy.
- Glitter iff gold is in the same square.

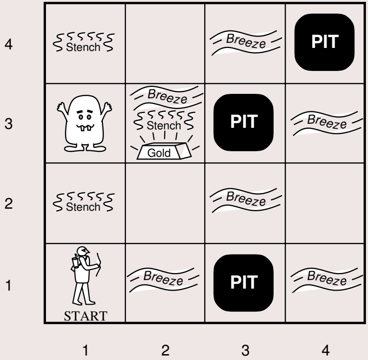


Wumpus world

Example: wumpus world - PEAS

Environment:

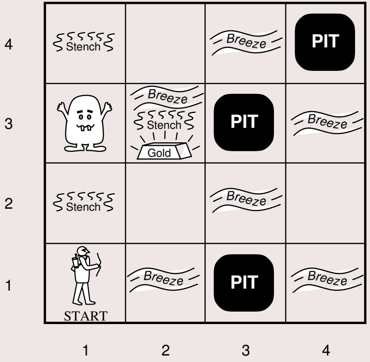
- Squares adjacent to wumpus are smelly.
- Squares adjacent to pit are breezy.
- Glitter iff gold is in the same square.
- Shooting kills wumpus if you are facing it.



Wumpus world

Example: wumpus world - PEAS

Environment (Cont.):

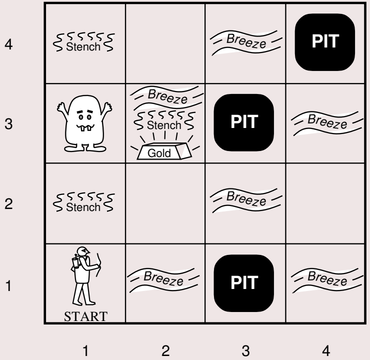


Wumpus world

Example: wumpus world - PEAS

Environment (Cont.):

- Shooting uses up the only arrow.

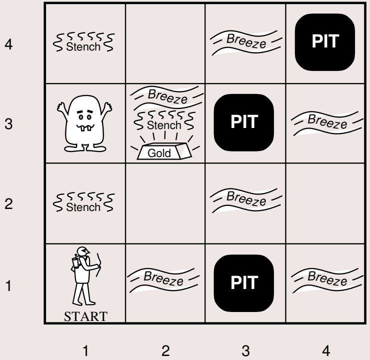


Wumpus world

Example: wumpus world - PEAS

Environment (Cont.):

- Shooting uses up the only arrow.
- Grabbing picks up gold if in same square.

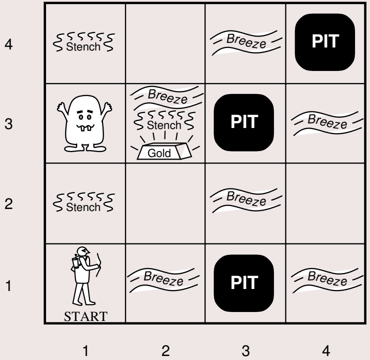


Wumpus world

Example: wumpus world - PEAS

Environment (Cont.):

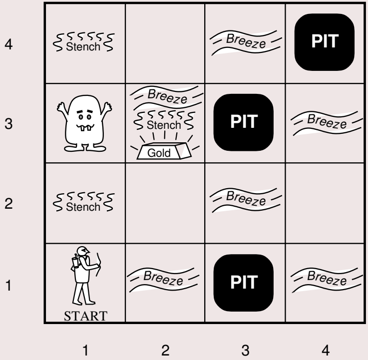
- Shooting uses up the only arrow.
- Grabbing picks up gold if in same square.
- Game ends:
  - Entering square w/t live wumpus or pit.
  - Climbing out of square (1,1).



Wumpus world

Example: wumpus world - PEAS

Sensors:

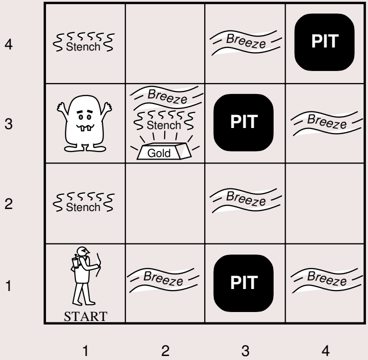


Wumpus world

Example: wumpus world - PEAS

Sensors:

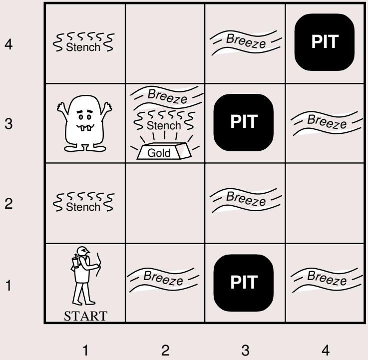
- Stench.
- Breeze.
- Glitter.
- Bump.
- Scream.



Wumpus world

Example: wumpus world - PEAS

Actuators:

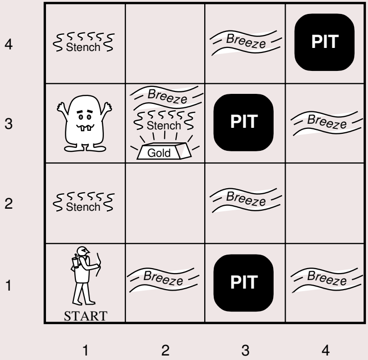


Wumpus world

Example: wumpus world - PEAS

Actuators:

- Left turn.
- Right turn.
- Forward.
- Grab.
- Shoot.
- Climb.



Wumpus world

Example: wumpus world - properties of environment

- Observable?

Wumpus world

Example: wumpus world - properties of environment

- Observable? Partially

Wumpus world

Example: wumpus world - properties of environment

- Observable? Partially
- Deterministic?

Wumpus world

Example: wumpus world - properties of environment

- Observable? Partially
- Deterministic? Yes

Wumpus world

Example: wumpus world - properties of environment

- Observable? Partially
- Deterministic? Yes
- Episodic?

Wumpus world

Example: wumpus world - properties of environment

- Observable? Partially
- Deterministic? Yes
- Episodic? Sequential

Wumpus world

Example: wumpus world - properties of environment

- Observable? Partially
- Deterministic? Yes
- Episodic? Sequential
- Static?

Wumpus world

Example: wumpus world - properties of environment

- Observable? Partially
- Deterministic? Yes
- Episodic? Sequential
- Static? Yes

Wumpus world

Example: wumpus world - properties of environment

- Observable? Partially
- Deterministic? Yes
- Episodic? Sequential
- Static? Yes
- Discrete?

Wumpus world

Example: wumpus world - properties of environment

- Observable? Partially
- Deterministic? Yes
- Episodic? Sequential
- Static? Yes
- Discrete? Yes

Wumpus world

Example: wumpus world - properties of environment

- Observable? Partially
- Deterministic? Yes
- Episodic? Sequential
- Static? Yes
- Discrete? Yes
- Single agent?

Wumpus world

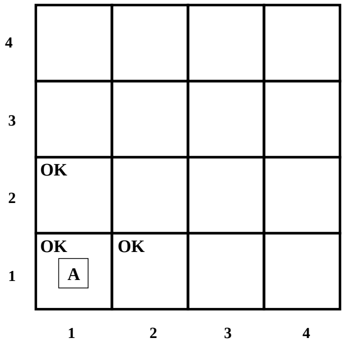
Example: wumpus world - properties of environment

- Observable? Partially
- Deterministic? Yes
- Episodic? Sequential
- Static? Yes
- Discrete? Yes
- Single agent? Yes

Wumpus world

Example: wumpus world - agent in action

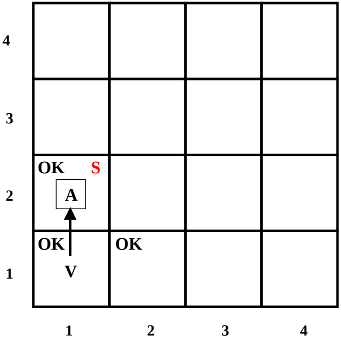
A sequence of actions leading to gold:



Wumpus world

Example: wumpus world - agent in action

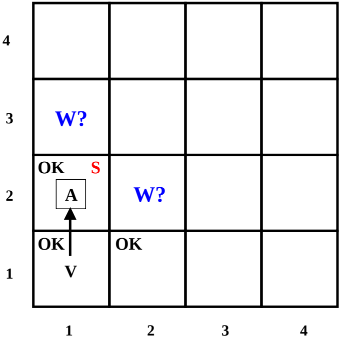
A sequence of actions leading to gold:



Wumpus world

Example: wumpus world - agent in action

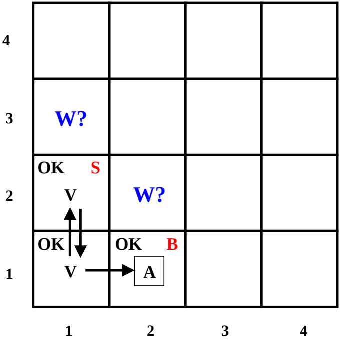
A sequence of actions leading to gold:



Wumpus world

Example: wumpus world - agent in action

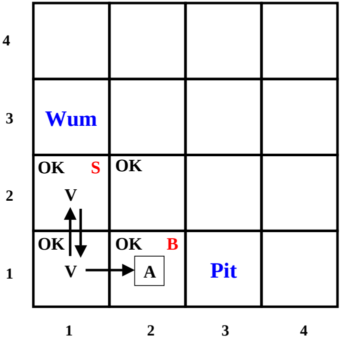
A sequence of actions leading to gold:



Wumpus world

Example: wumpus world - agent in action

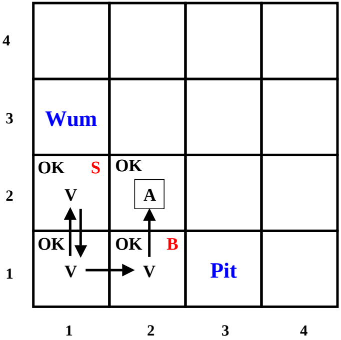
A sequence of actions leading to gold:



Wumpus world

Example: wumpus world - agent in action

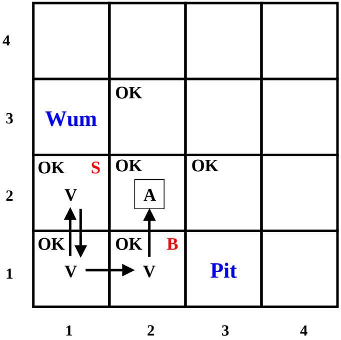
A sequence of actions leading to gold:



Wumpus world

Example: wumpus world - agent in action

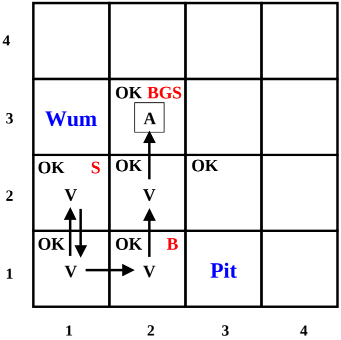
A sequence of actions leading to gold:



Wumpus world

Example: wumpus world - agent in action

A sequence of actions leading to gold:



Wumpus world

Example: wumpus world - KB agent

- In order to implement an agent for traveling the wumpus world, the main difficulty would be the agent's initial ignorance of the environment configuration.

Wumpus world

Example: wumpus world - KB agent

- In order to implement an agent for traveling the wumpus world, the main difficulty would be the agent's initial ignorance of the environment configuration.
- ***Logical reasoning*** is required to overcome this ignorance.

Wumpus world

Example: wumpus world - KB agent

- The agent should:

Wumpus world

Example: wumpus world - KB agent

- The agent should:
  - Use the perceived information (breeze, stench, *etc.*) in order to

Wumpus world

Example: wumpus world - KB agent

- The agent should:
  - Use the perceived information (breeze, stench, *etc.*) in order to
  - Deduce more facts about the environment (is a certain place safe?) and

Wumpus world

Example: wumpus world - KB agent

- The agent should:
  - Use the perceived information (breeze, stench, *etc.*) in order to
  - Deduce more facts about the environment (is a certain place safe?) and
  - Take the correct action (turn left, shoot, *etc.*).

Wumpus world

Example: wumpus world - KB agent

- The agent should:
  - Use the perceived information (breeze, stench, *etc.*) in order to
  - Deduce more facts about the environment (is a certain place safe?) and
  - Take the correct action (turn left, shoot, *etc.*).
- This is a complicated process requiring:

Wumpus world

Example: wumpus world - KB agent

- The agent should:
  - Use the perceived information (breeze, stench, *etc.*) in order to
  - Deduce more facts about the environment (is a certain place safe?) and
  - Take the correct action (turn left, shoot, *etc.*).
- This is a complicated process requiring:
  - Combining knowledge learned at different places in different time steps.

Wumpus world

Example: wumpus world - KB agent

- The agent should:
  - Use the perceived information (breeze, stench, *etc.*) in order to
  - Deduce more facts about the environment (is a certain place safe?) and
  - Take the correct action (turn left, shoot, *etc.*).
- This is a complicated process requiring:
  - Combining knowledge learned at different places in different time steps.
  - Relying on the lack of a percept.

Logic

Logical representation

- Remember that the KB consists of a set of sentences.

Logic

Logical representation

- Remember that the KB consists of a set of sentences.
- The sentences are expressed using a certain **syntax** specifying all well formed sentences.

Logic

Logical representation

- Remember that the KB consists of a set of sentences.
- The sentences are expressed using a certain **syntax** specifying all well formed sentences.
- For example, in a mathematical syntax:
  - "X + Y = 2" is a well formed sentence.
  - "X 2 Y + =" is not.

Logic

Logical representation

- Logic should also define the **semantics** of a language.

Logic

Logical representation

- Logic should also define the **semantics** of a language.
- Semantics are related to the meanings of a sentence.

Logic

Logical representation

- Logic should also define the **semantics** of a language.
- Semantics are related to the meanings of a sentence.
- In logic, the semantics define the **truth** of each sentence in each ***possible world***.

Logic

Logical representation

- Logic should also define the **semantics** of a language.
- Semantics are related to the meanings of a sentence.
- In logic, the semantics define the **truth** of each sentence in each ***possible world***.
- For example, ”X + Y = 4” is:
  - TRUE in a world where “X=2 and Y=2”.
  - FALSE in a world where “X=1 and Y=1”.

Logic

Logical representation

- Logic should also define the **semantics** of a language.
- Semantics are related to the meanings of a sentence.
- In logic, the semantics define the **truth** of each sentence in each ***possible world***.
- For example, ”X + Y = 4” is:
  - TRUE in a world where “X=2 and Y=2”.
  - FALSE in a world where “X=1 and Y=1”.
- A ***possible world*** is referred to as a ***model***.

Logic

Logical reasoning

- Logic reasoning is related to the notion of **logical entailment**.

Logic

Logical reasoning

- Logic reasoning is related to the notion of **logical entailment**.
- In mathematical notation:
  $\alpha \models \beta$
  which is read as: ”sentence α **entails** sentence β”.

Logic

Logical reasoning

- Logic reasoning is related to the notion of **logical entailment**.
- In mathematical notation:
  $\alpha \models \beta$
  which is read as: ”sentence α **entails** sentence β”.
- This means that in **every model** where α is **TRUE**, β is also **TRUE**.

Logic

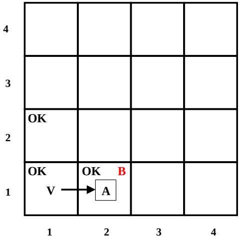
Logical reasoning

- Logic reasoning is related to the notion of **logical entailment**.
- In mathematical notation:
  $\alpha \models \beta$
  which is read as: "sentence α **entails** sentence β".
- This means that in **every model** where α is **TRUE**, β is also **TRUE**.
- If an inference algorithm i can derive β from α:
  $\alpha \vdash_i \beta$
  which is read as: "sentence β is **derived from** sentence α by i".

Logic

Example: wumpus world - logical reasoning

So, consider the shown case, after one step, the KB consists of:

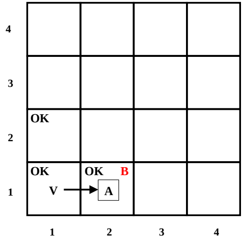


Logic

Example: wumpus world - logical reasoning

So, consider the shown case, after one step, the KB consists of:

- ***RULES:*** The set of rules governing the world.

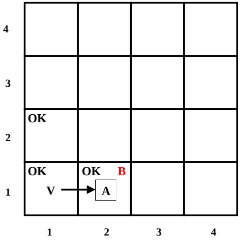


Logic

Example: wumpus world - logical reasoning

So, consider the shown case, after one step, the KB consists of:

- ***RULES:*** The set of rules governing the world.
- ***FACT:*** nothing is detected in [1,1].

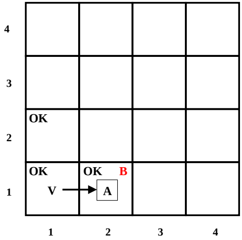


Logic

Example: wumpus world - logical reasoning

So, consider the shown case, after one step, the KB consists of:

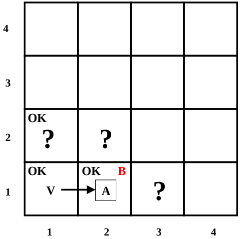
- ***RULES:*** The set of rules governing the world.
- ***FACT:*** nothing is detected in [1,1].
- ***FACT:*** a breeze is detected in [2,1].



Logic

Example: wumpus world - logical reasoning

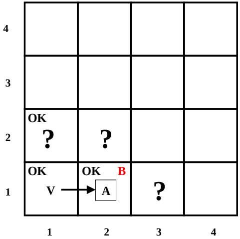
- Let's say the agent wants to know if any of the squares [1,2], [2,2] or [3,1] contain a pit.



Logic

Example: wumpus world - logical reasoning

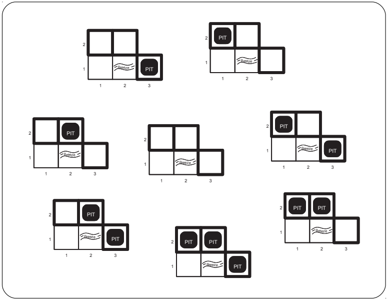
- Let's say the agent wants to know if any of the squares [1,2], [2,2] or [3,1] contain a pit.
- We have $2^3 = 8$ possible models.



Logic

Example: wumpus world - logical reasoning

Logical reasoning by model checking:

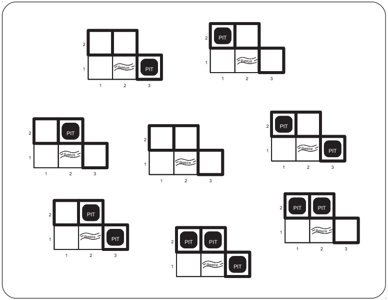


We have 8 different possible worlds, (8 different models)

Logic

Example: wumpus world - logical reasoning

Logical reasoning by model checking:



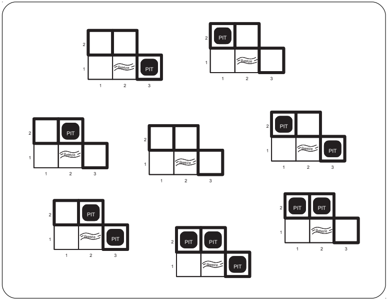
We have 8 different possible worlds, (8 different models)

KB is not TRUE in all of the possible models

Logic

Example: wumpus world - logical reasoning

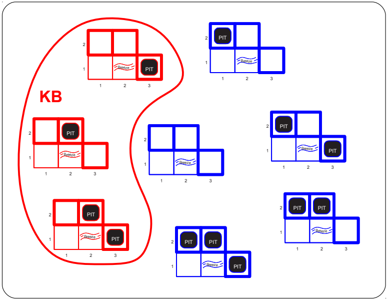
Logical reasoning by model checking:



We have 8 different possible worlds, (8 different models)
KB is not TRUE in all of the possible models
For example, there cannot be a pit in [1,2]

Example: wumpus world - logical reasoning

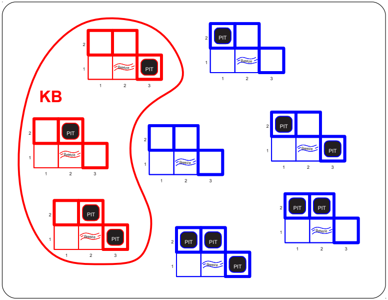
Logical reasoning by model checking:



In fact, KB is only TRUE in three of them

Example: wumpus world - logical reasoning

Logical reasoning by model checking:



Now, let's consider:

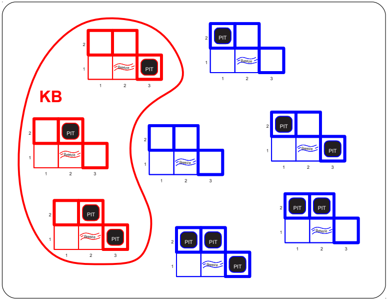
β_1 = "There's no pit in [1,2]"

β_2 = "There's no pit in [2,2]"

Logic

Example: wumpus world - logical reasoning

Logical reasoning by model checking:

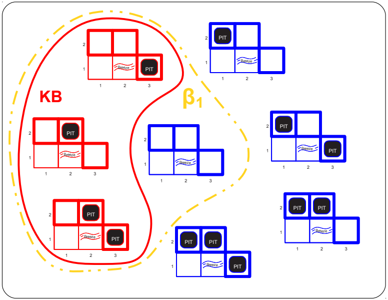


β_1 = "There's no pit in [1,2]"
β_2 = "There's no pit in [2,2]"
Can we prove that: $KB \models \beta_1$ and $KB \models \beta_2$?

Logic

Example: wumpus world - logical reasoning

Logical reasoning by model checking:

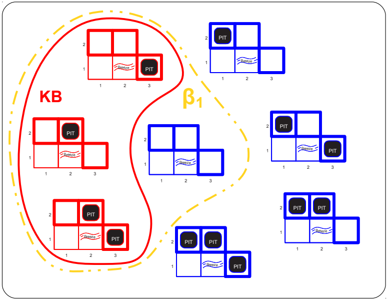


Shown are models in which β_1 is TRUE

Logic

Example: wumpus world - logical reasoning

Logical reasoning by model checking:



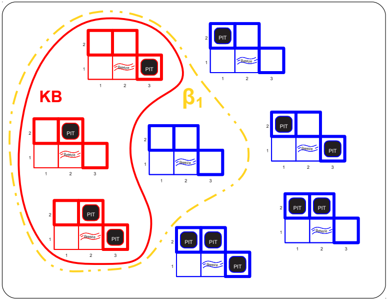
Shown are models in which β_1 is TRUE

In all models in which KB is TRUE, β_1 is also TRUE

Logic

Example: wumpus world - logical reasoning

Logical reasoning by model checking:



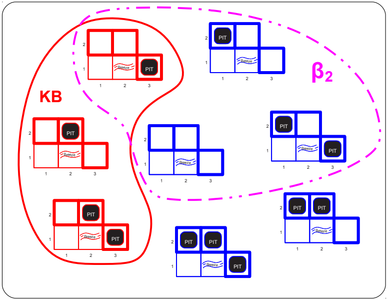
Shown are models in which β_1 is TRUE

In all models in which KB is TRUE, β_1 is also TRUE

Hence, $KB \models \beta_1$, no pit in [1,2]

Example: wumpus world - logical reasoning

Logical reasoning by model checking:

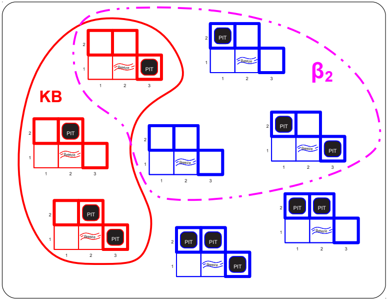


Shown are models in which β_2 is TRUE

Logic

Example: wumpus world - logical reasoning

Logical reasoning by model checking:



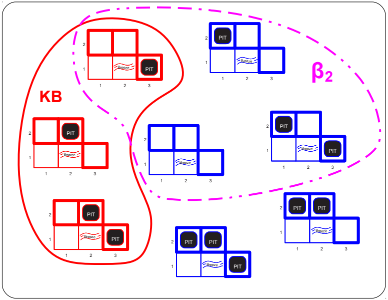
Shown are models in which β_2 is TRUE

In some models in which KB is TRUE, β_2 is FALSE

Logic

Example: wumpus world - logical reasoning

Logical reasoning by model checking:



Shown are models in which β_2 is TRUE
In some models in which KB is TRUE, β_2 is FALSE
Hence, $KB \not\models \beta_2$, cannot conclude there's no pit in [2,2]

Logic

Logical reasoning

- The previous approach is known as ***Model checking***.

Logic

Logical reasoning

- The previous approach is known as ***Model checking***.
- It enumerates all possible models to check whether α is TRUE in all models in which KB is TRUE.

Logic

Logical reasoning

- The previous approach is known as ***Model checking***.
- It enumerates all possible models to check whether α is TRUE in all models in which KB is TRUE.
- This approach is:
  - ***Sound:*** derive entailed sentences only, never makes things up.
  - ***Complete:*** can derive any sentence that is entailed.

Propositional logic

Introduction

- ***Propositional logic*** is a very simple logic approach that could be used by an agent for traveling through the wumpus world.

Propositional logic

Introduction

- ***Propositional logic*** is a very simple logic approach that could be used by an agent for traveling through the wumpus world.
- Also known as ***Boolean logic***.

Propositional logic

Introduction

- ***Propositional logic*** is a very simple logic approach that could be used by an agent for traveling through the wumpus world.
- Also known as ***Boolean logic***.
- We will take a look at:
  - Syntax.
  - Semantics.
  - Entailment.

Propositional logic

Introduction

- ***Propositional logic*** is a very simple logic approach that could be used by an agent for traveling through the wumpus world.
- Also known as ***Boolean logic***.
- We will take a look at:
  - Syntax.
  - Semantics.
  - Entailment.
- This will lead for a very simple algorithm for logical inference.

Propositional logic

Syntax

- Two types of sentences occur:

Propositional logic

Syntax

- Two types of sentences occur:
  - ***Atomic sentences:*** a single ***proposition symbol*** which are basically any variable name, TRUE or FALSE.

Propositional logic

Syntax

- Two types of sentences occur:
  - ***Atomic sentences:*** a single ***proposition symbol*** which are basically any variable name, TRUE or FALSE.
  - ***Complex sentences:*** constructed from simpler sentences using ***logical connectives***.

Propositional logic

Syntax

- Two types of sentences occur:
  - ***Atomic sentences:*** a single ***proposition symbol*** which are basically any variable name, TRUE or FALSE.
  - ***Complex sentences:*** constructed from simpler sentences using ***logical connectives***.
- Logical connectives:

$\neg$	negation.
$\wedge$	conjunction (and).
$\vee$	disjunction (or).
$\Rightarrow$	implication (if-then).
$\Leftrightarrow$	biconditional (if and only if).

Propositional logic

Syntax

- Sentences formed in this way can be called ***Well-Formed Formula*** (***WFF***).

Propositional logic

Syntax

- Sentences formed in this way can be called ***Well-Formed Formula*** (***WFF***).
- Parentheses can be used to indicate precedence.

Propositional logic

Syntax

- Sentences formed in this way can be called ***Well-Formed Formula*** (***WFF***).
- Parentheses can be used to indicate precedence.
- Complex sentences evaluate to TRUE or FALSE.

Propositional logic

Syntax

- Sentences formed in this way can be called ***Well-Formed Formula*** (***WFF***).
- Parentheses can be used to indicate precedence.
- Complex sentences evaluate to TRUE or FALSE.
- The symbols and the connectives together define the syntax of the language.

Propositional logic

Semantics

- Semantics define rules for determining the truth of a sentence with respect to a particular model.

Propositional logic

Semantics

- Semantics define rules for determining the truth of a sentence with respect to a particular model.
- Need to be able to evaluate sentences to TRUE or FALSE:

Propositional logic

Semantics

- Semantics define rules for determining the truth of a sentence with respect to a particular model.
- Need to be able to evaluate sentences to TRUE or FALSE:
  - Atomic sentences must have truth assigned (*i.e.*, be assigned TRUE of FALSE).

Propositional logic

Semantics

- Semantics define rules for determining the truth of a sentence with respect to a particular model.
- Need to be able to evaluate sentences to TRUE or FALSE:
  - Atomic sentences must have truth assigned (*i.e.*, be assigned TRUE of FALSE).
  - The evaluation of complex sentences is done using ***truth tables*** for the connectives.

Propositional logic

Semantics

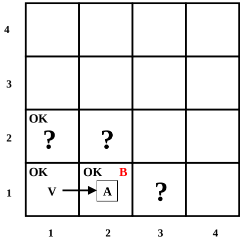
- Semantics define rules for determining the truth of a sentence with respect to a particular model.
- Need to be able to evaluate sentences to TRUE or FALSE:
  - Atomic sentences must have truth assigned (*i.e.*, be assigned TRUE of FALSE).
  - The evaluation of complex sentences is done using ***truth tables*** for the 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*

Propositional logic

Example: wumpus world - a simple KB in PL

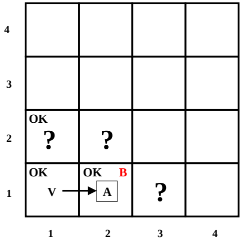
- We want to represent what we know about the wumpus world in a simple KB using propositional logic.



Propositional logic

Example: wumpus world - a simple KB in PL

- We want to represent what we know about the wumpus world in a simple KB using propositional logic.
- Proposition symbols:
  - Let P_{ij} be TRUE is there's a pit in location [i,j].
  - Let B_{ij} be TRUE is there's a breeze in location [i,j].



Propositional logic

Example: wumpus world - a simple KB in PL

- KB consists of:

Propositional logic

Example: wumpus world - a simple KB in PL

- KB consists of:
  - ***FACT:*** there's no pit in [1,1]:

Propositional logic

Example: wumpus world - a simple KB in PL

- KB consists of:
  - ***FACT:*** there's no pit in [1,1]:

    R1: $\neg P_{11}$.

Propositional logic

Example: wumpus world - a simple KB in PL

- KB consists of:
  - ***FACT:*** there's no pit in [1,1]:

    R1: $\neg P_{11}$.
  - ***RULES:*** A square is breezy if and only if there's a pit in a neighboring square:

Propositional logic

Example: wumpus world - a simple KB in PL

- KB consists of:
  - ***FACT:*** there's no pit in [1,1]:

    R1: $\neg P_{11}$.
  - ***RULES:*** A square is breezy if and only if there's a pit in a neighboring square:

    R2: $B_{11} \Leftrightarrow (P_{12} \lor P_{21})$.
    R3: $B_{21} \Leftrightarrow (P_{11} \lor P_{22} \lor P_{31})$.

Propositional logic

Example: wumpus world - a simple KB in PL

- KB consists of:
  - ***FACT:*** there's no pit in [1,1]:

    R1: $\neg P_{11}$.
  - ***RULES:*** A square is breezy if and only if there's a pit in a neighboring square:

    R2: $B_{11} \Leftrightarrow (P_{12} \vee P_{21})$.
    R3: $B_{21} \Leftrightarrow (P_{11} \vee P_{22} \vee P_{31})$.
  - ***FACTS:*** from sensors:

Propositional logic

Example: wumpus world - a simple KB in PL

- KB consists of:
  - ***FACT:*** there's no pit in [1,1]:

    R1: $\neg P_{11}$.
  - ***RULES:*** A square is breezy if and only if there's a pit in a neighboring square:

    R2: $B_{11} \Leftrightarrow (P_{12} \vee P_{21})$.
    R3: $B_{21} \Leftrightarrow (P_{11} \vee P_{22} \vee P_{31})$.
  - ***FACTS:*** from sensors:

    R4: $\neg B_{11}$.
    R5: B_{21}.

Propositional logic

Example: wumpus world - a simple KB in PL

- KB consists of:
  - ***FACT:*** there's no pit in [1,1]:

    R1: $\neg P_{11}$.
  - ***RULES:*** A square is breezy if and only if there's a pit in a neighboring square:

    R2: $B_{11} \Leftrightarrow (P_{12} \vee P_{21})$.
    R3: $B_{21} \Leftrightarrow (P_{11} \vee P_{22} \vee P_{31})$.
  - ***FACTS:*** from sensors:

    R4: $\neg B_{11}$.
    R5: B_{21}.
- The KB can be represented as the conjunction of all sentences:

  $R_1 \wedge R_2 \wedge R_3 \wedge R_4 \wedge R_5$

Propositional logic

Example: wumpus world - reasoning in PL

Reasoning in propositional logic by model checking:

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

Propositional logic

Example: wumpus world - reasoning in PL

Reasoning in propositional logic by model checking:

$B_{1,1}$	$B_{2,1}$	$P_{1,1}$	$P_{1,2}$	$P_{2,1}$	$P_{2,2}$	$P_{3,1}$	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	<u>true</u>
false	true	false	false	false	true	false	true	true	true	true	true	<u>true</u>
false	true	false	false	false	true	true	true	true	true	true	true	<u>true</u>
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

Now, let's consider:

β_1 = "There's no pit in [1,2]" = $\neg P_{12}$

Can we prove that $KB \models \beta_1$?

Propositional logic

Example: wumpus world - reasoning in PL

Reasoning in propositional logic by model checking:

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

We have seven proposition symbols in the KB:

$B_{11}, B_{21}, P_{11}, P_{12}, P_{21}, P_{22}, P_{31}$

We have $2^7 = 128$ possible models

Propositional logic

Example: wumpus world - reasoning in PL

Reasoning in propositional logic by model checking:

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

There is only 3 models in which the KB is TRUE

Propositional logic

Example: wumpus world - reasoning in PL

Reasoning in propositional logic by model checking:

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

There is only 3 models in which the KB is TRUE

In all three models, $\beta_1 = \neg P_{12}$ is also TRUE

Propositional logic

Example: wumpus world - reasoning in PL

Reasoning in propositional logic by model checking:

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

There is only 3 models in which the KB is TRUE

In all three models, $\beta_1 = \neg P_{12}$ is also TRUE

Hence, $KB \models \beta_1$.

Propositional logic

Example: wumpus world - reasoning in PL

Reasoning in propositional logic by model checking:

$B_{1,1}$	$B_{2,1}$	$P_{1,1}$	$P_{1,2}$	$P_{2,1}$	$P_{2,2}$	$P_{3,1}$	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*	<u>*true*</u>
false	*true*	*false*	*false*	*false*	*true*	*false*	*true*	*true*	*true*	*true*	*true*	<u>*true*</u>
false	*true*	*false*	*false*	*false*	*true*	*true*	*true*	*true*	*true*	*true*	*true*	<u>*true*</u>
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*

On the other hand, if β_2 = “There’s no pit in [2,2]” = $\neg P_{22}$

Propositional logic

Example: wumpus world - reasoning in PL

Reasoning in propositional logic by model checking:

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

On the other hand, if β_2 = "There's no pit in [2,2]" = $\neg P_{22}$

$\beta_2 = \neg P_{22}$ is TRUE in only one of these models

Propositional logic

Example: wumpus world - reasoning in PL

Reasoning in propositional logic by model checking:

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

On the other hand, if β_2 = "There's no pit in [2,2]" = $\neg P_{22}$

$\beta_2 = \neg P_{22}$ is TRUE in only one of these models

Hence, $KB \not\models \beta_2$.

Propositional logic

Logical reasoning

- The previous approach is known as ***Model checking***.

Propositional logic

Logical reasoning

- The previous approach is known as ***Model checking***.
- Also known as ***Inferencing with Truth Tables***,

Propositional logic

Logical reasoning

- The previous approach is known as ***Model checking***.
- Also known as ***Inferencing with Truth Tables***,
- Sound and complete.

Propositional logic

Logical reasoning

- The previous approach is known as ***Model checking***.
- Also known as ***Inferencing with Truth Tables***,
- Sound and complete.
- However, Time complexity = $O(2^n)$.

Outline

1. Logical Agents
   - Knowledge-based agents
   - Wumpus world
   - Logic
   - Propositional logic

2. Requirements & Reading Material

Requirements

What do I need from you

- When given a certain problem you should be able to:
  - Express the problem in terms of propositional logic (i.e. write the KB in form of rules).
  - Inference certain conclusions using:
    - Truth tables.
- Answer descriptive questions.

Reading Material

Which parts of the textbook are covered

- Russell-Norvig, Chapters 7:
  - Pages 234 - 248.