# IS-ZC444: Artificial Intelligence

## Lecture-10: Logical Agents

**Dr. Kamlesh Tiwari**
Assistant Professor
Department of Computer Science and Information Systems,
BITS Pilani, Pilani, Jhunjhunu-333031, Rajasthan, INDIA

October 10, 2018 (WILP @ BITS-Pilani Jul-Nov 2018)

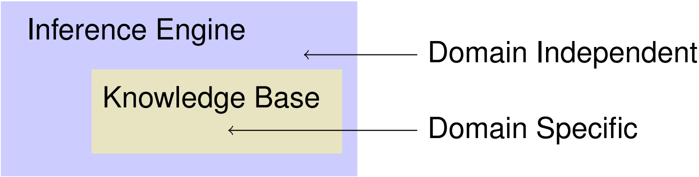
# Logical Agents

- Humans, it seems, know things
- And what they know, helps them to think/reason
- Knowledge base is set of **sentences**[1]

[1] Sentences are derived from knowledge representation language. An underived sentence is **axiom**

# Logical Agents

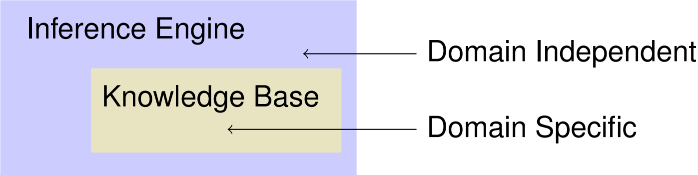
- Humans, it seems, know things
- And what they know, helps them to think/reason
- Knowledge base is set of **sentences**[1]
- Knowledge-based agents




[1] Sentences are derived from knowledge representation language. An underived sentence is **axiom**

# Logical Agents

- Humans, it seems, know things
- And what they know, helps them to think/reason
- Knowledge base is set of **sentences**[1]
- Knowledge-based agents




- Agent program does three operation
  1. **TELL** the knowledge base what it have perceived
  2. **ASK** what to do
  3. **TELL** which action is chosen and the agent execute that action

[1] Sentences are derived from knowledge representation language. An underived sentence is **axiom**

# A Generic Knowledge Based Agent

**Algorithm 1:** KB-Agent(percept)

**Input**: *KB* is knowledge base *t* a counter 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
```

# A Generic Knowledge Based Agent

**Algorithm 2:** KB-Agent(percept)

**Input**: *KB* is knowledge base *t* a counter indicating time

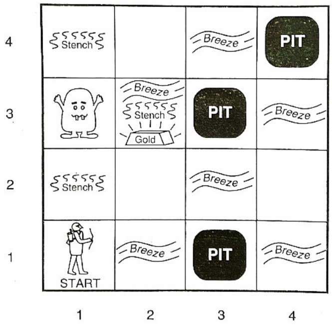
```
TELL(KB, Make-Percept-Sentence(percept,t))
action ← ASK(KB, Make-Action-Query(t))
TELL(KB, Make-Action-Sentence(action,t))
t ← t +1
return action
```

## Knowledge base could be built through

- **Declarative** procedure that tell everything
- **Procedural** approach that writes a program code

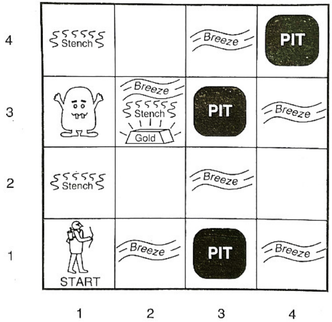
# Example: Wumpus World

- **Performance** gold +100, death -100, step -1, arrow -10
- **Environment** smell around wumpus, breeze around pit
- **Actuator** turn left/right, forward, grab, release, shoot
- **Sensor** breeze, glitter, smell, bump, scream



# Example: Wumpus World

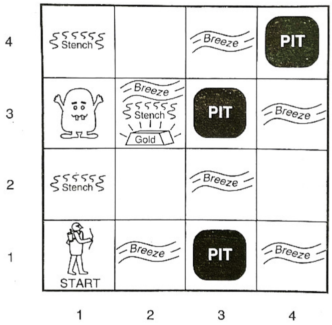
- **Performance** gold +100, death -100, step -1, arrow -10
- **Environment** smell around wumpus, breeze around pit
- **Actuator** turn left/right, forward, grab, release, shoot
- **Sensor** breeze, glitter, smell, bump, scream




Observable? No

# Example: Wumpus World

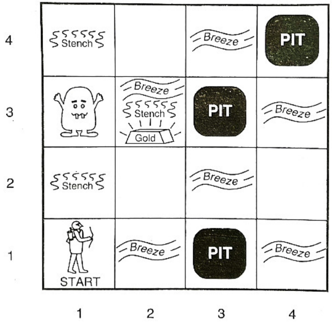
- **Performance** gold +100, death -100, step -1, arrow -10
- **Environment** smell around wumpus, breeze around pit
- **Actuator** turn left/right, forward, grab, release, shoot
- **Sensor** breeze, glitter, smell, bump, scream




| | |
|---|---|
| Observable? | No |
| Deterministic? | Yes |

# Example: Wumpus World

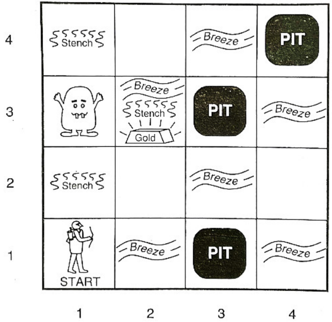
- **Performance** gold +100, death -100, step -1, arrow -10
- **Environment** smell around wumpus, breeze around pit
- **Actuator** turn left/right, forward, grab, release, shoot
- **Sensor** breeze, glitter, smell, bump, scream




| | |
|---|---|
| Observable? | No |
| Deterministic? | Yes |
| Episodic? | No |

# Example: Wumpus World

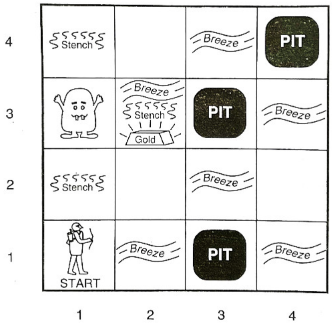
- **Performance** gold +100, death -100, step -1, arrow -10
- **Environment** smell around wumpus, breeze around pit
- **Actuator** turn left/right, forward, grab, release, shoot
- **Sensor** breeze, glitter, smell, bump, scream




| | |
|---|---|
| Observable? | No |
| Deterministic? | Yes |
| Episodic? | No |
| Static? | Yes |

# Example: Wumpus World

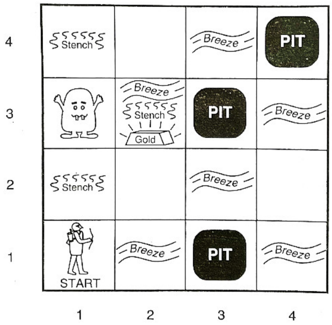
- **Performance** gold +100, death -100, step -1, arrow -10
- **Environment** smell around wumpus, breeze around pit
- **Actuator** turn left/right, forward, grab, release, shoot
- **Sensor** breeze, glitter, smell, bump, scream




| | |
|---|---|
| Observable? | No |
| Deterministic? | Yes |
| Episodic? | No |
| Static? | Yes |
| Discrete? | Yes |

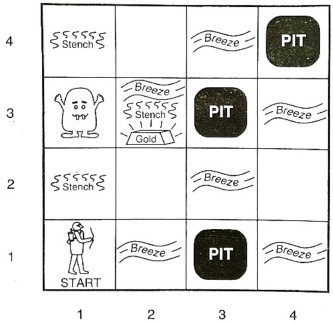
# Example: Wumpus World

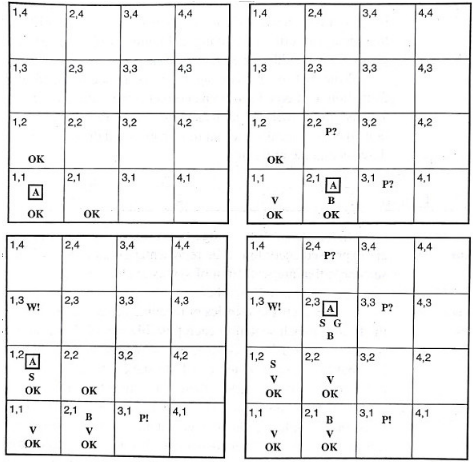
- **Performance** gold +100, death -100, step -1, arrow -10
- **Environment** smell around wumpus, breeze around pit
- **Actuator** turn left/right, forward, grab, release, shoot
- **Sensor** breeze, glitter, smell, bump, scream




| | |
|---|---|
| Observable? | No |
| Deterministic? | Yes |
| Episodic? | No |
| Static? | Yes |
| Discrete? | Yes |
| Single Agent? | Yes |

# Wumpus World



# Logic

- Knowledge base contains sentences that are written according to some syntax that specified whether they are well formed or not.

# Logic

- Knowledge base contains sentences that are written according to some syntax that specified whether they are well formed or not.
- Sementics defines truth of each sentence. $x + y = 4$ is
  - **true** if $x = 1$ and $y = 3$ but
  - **not true** when $x = 2$ and $y = 3$

# Logic

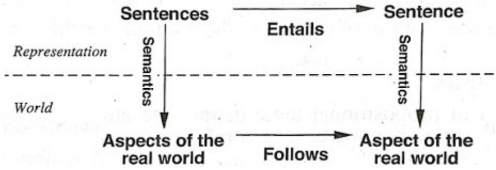
- Knowledge base contains sentences that are written according to some syntax that specified whether they are well formed or not.
- Sementics defines truth of each sentence. $x + y = 4$ is
  - **true** if $x = 1$ and $y = 3$ but
  - **not true** when $x = 2$ and $y = 3$
- Here $x + y = 4$ is a model that is satisfied by some sentence $\alpha$
- We use $M(\alpha)$ to represent a set of all models of $\alpha$

# Logic

- Knowledge base contains sentences that are written according to some syntax that specified whether they are well formed or not.
- Sementics defines truth of each sentence. $x + y = 4$ is
  - **true** if $x = 1$ and $y = 3$ but
  - **not true** when $x = 2$ and $y = 3$
- Here $x + y = 4$ is a model that is satisfied by some sentence $\alpha$
- We use $M(\alpha)$ to represent a set of all models of $\alpha$
- Logical entailment between sentences means that a sentence logically follows from another one $\alpha \models \beta$ it means $M(\alpha) \subseteq M(\beta)$

# Logic

- Knowledge base contains sentences that are written according to some syntax that specified whether they are well formed or not.
- Sementics defines truth of each sentence. $x + y = 4$ is
  - **true** if $x = 1$ and $y = 3$ but
  - **not true** when $x = 2$ and $y = 3$
- Here $x + y = 4$ is a model that is satisfied by some sentence $\alpha$
- We use $M(\alpha)$ to represent a set of all models of $\alpha$
- Logical entailment between sentences means that a sentence logically follows from another one $\alpha \models \beta$ it means $M(\alpha) \subseteq M(\beta)$
- Relationship between representation and real world



# Propositional Logic

- **Propositions** or **declarative sentences** can be true or false
  - Sum of 5 and 4 is 9
  - Could you give me your pen
  - Every even natural number grater then two can be written as sum of two primes
  - Best of luck

# Propositional Logic

- **Propositions** or **declarative sentences** can be true or false
  - Sum of 5 and 4 is 9
  - Could you give me your pen
  - Every even natural number grater then two can be written as sum of two primes
  - Best of luck
- Sometime it is better to assign symbols to a atomic sentences
  $p$: "I won a lottery last week"
  $q$: "I have purchased a lottery ticket last week"

# Propositional Logic

- **Propositions** or **declarative sentences** can be true or false
  - Sum of 5 and 4 is 9
  - Could you give me your pen
  - Every even natural number grater then two can be written as sum of two primes
  - Best of luck
- Sometime it is better to assign symbols to a atomic sentences
  $p$: "I won a lottery last week"
  $q$: "I have purchased a lottery ticket last week"
- Complex sentences could be formed by using

| | | |
|---|---|---|
| $\neg$ | : | negative |
| $\vee$ | : | disjunction, at least one is true |
| $\wedge$ | : | conjunction, both should be true |
| $\rightarrow$ | : | implication |

# Truth Table and Natural Deduction

| $P$ | $Q$ | $\neg P$ | $P \wedge Q$ | $P \vee Q$ | $P \rightarrow Q$ | $P \Leftrightarrow Q$ |
|---|---|---|---|---|---|---|
| T | T | F | T | T | T | T |
| T | F | F | F | T | F | F |
| F | T | T | F | T | T | F |
| F | F | T | F | F | T | T |

# Truth Table and Natural Deduction

| $P$ | $Q$ | $\neg P$ | $P \wedge Q$ | $P \vee Q$ | $P \rightarrow Q$ | $P \Leftrightarrow Q$ |
|---|---|---|---|---|---|---|
| T | T | F | T | T | T | T |
| T | F | F | F | T | F | F |
| F | T | T | F | T | T | F |
| F | F | T | F | F | T | T |

## Natural Deduction

Suppose we have formulas $\phi_1$, $\phi_2$, ..., $\phi_n$ and we have applied some proof rules to get another formula $\psi$ then we denote

$$\phi_1, \phi_2, ..., \phi_n \vdash \psi$$

This equation is called sequent; and is valid if a proof can be found

# Rules for Conjunction

$$\frac{\phi \qquad \psi}{\phi \wedge \psi} \wedge \mathrm{i} \qquad \frac{\phi \wedge \psi}{\phi} \wedge e_1 \qquad \frac{\phi \wedge \psi}{\psi} \wedge e_2$$

# Rules for Conjunction

$$\frac{\phi \quad \psi}{\phi \land \psi} \land \mathrm{i} \qquad \frac{\phi \land \psi}{\phi} \land e_1 \qquad \frac{\phi \land \psi}{\psi} \land e_2$$

**Show $p \land q, r \vdash q \land r$**

# Rules for Conjunction

$$\frac{\phi \quad \psi}{\phi \wedge \psi} \wedge \mathrm{i} \qquad \frac{\phi \wedge \psi}{\phi} \wedge e_1 \qquad \frac{\phi \wedge \psi}{\psi} \wedge e_2$$

## Show $p \wedge q, r \vdash q \wedge r$

| | | |
|---|---|---|
| 1 | $p \wedge q$ | premise |
| 2 | $r$ | premise |
| 3 | $q$ | $\wedge e_2$ 1 |
| 4 | $q \wedge r$ | $\wedge i$ 3,2 |

# Rules for Conjunction

$$\frac{\phi \quad \psi}{\phi \wedge \psi} \wedge i \qquad \frac{\phi \wedge \psi}{\phi} \wedge e_1 \qquad \frac{\phi \wedge \psi}{\psi} \wedge e_2$$

**Show $p \wedge q, r \vdash q \wedge r$**

| | | |
|---|---|---|
| 1 | $p \wedge q$ | premise |
| 2 | $r$ | premise |
| 3 | $q$ | $\wedge e_2$ 1 |
| 4 | $q \wedge r$ | $\wedge i$ 3,2 |

- Show $(p \wedge q) \wedge r, s \wedge t \vdash q \wedge s$

# Rules for Double Negation

Introduction $$\frac{\phi}{\neg\neg\phi}\ \neg\neg i$$

Elimination $$\frac{\neg\neg\phi}{\phi}\ \neg\neg e$$

# Rules for Double Negation

Introduction $\dfrac{\phi}{\neg\neg\phi}\ \neg\neg i$

Elimination $\dfrac{\neg\neg\phi}{\phi}\ \neg\neg e$

$p, \neg\neg(q \wedge r) \vdash \neg\neg p \wedge r$

# Rules for Double Negation

Introduction $\dfrac{\phi}{\neg\neg\phi}\ \neg\neg i$  Elimination $\dfrac{\neg\neg\phi}{\phi}\ \neg\neg e$

**$p, \neg\neg(q \wedge r) \vdash \neg\neg p \wedge r$**

| | | |
|---|---|---|
| 1 | $p$ | premise |
| 2 | $\neg\neg(q \wedge r)$ | premise |
| 3 | $\neg\neg p$ | $\neg\neg i$ 1 |
| 4 | $q \wedge r$ | $\neg\neg e$ 2 |
| 5 | $r$ | $\wedge e_2$ 4 |
| 6 | $\neg\neg p \wedge r$ | $\wedge i$ 3,5 |

# Rules for Implication Elimination

Elimination $\dfrac{\phi \quad \phi \rightarrow \psi}{\psi} \rightarrow e$

# Rules for Implication Elimination

Elimination $\dfrac{\phi \quad \phi \to \psi}{\psi} \to e$

**Can $p, p \to q, p \to (q \to r)$ infer $r$**

# Rules for Implication Elimination

Elimination $\dfrac{\phi \quad \phi \to \psi}{\psi} \to e$

## Can $p, p \to q, p \to (q \to r)$ infer $r$

| | | |
|---|---|---|
| 1 | $p \to (q \to r)$ | premise |
| 2 | $p \to q$ | premise |
| 3 | $p$ | premise |
| 4 | $q \to r$ | $\to e$ 1,3 |
| 5 | $q$ | $\to e$ 2,3 |
| 6 | $r$ | $\to e$ 4,5 |

# Rules for Implication Elimination

Elimination $$\frac{\phi \quad \phi \rightarrow \psi}{\psi} \rightarrow e$$

## Can $p, p \rightarrow q, p \rightarrow (q \rightarrow r)$ infer $r$

| | | |
|---|---|---|
| 1 | $p \rightarrow (q \rightarrow r)$ | premise |
| 2 | $p \rightarrow q$ | premise |
| 3 | $p$ | premise |
| 4 | $q \rightarrow r$ | $\rightarrow e$ 1,3 |
| 5 | $q$ | $\rightarrow e$ 2,3 |
| 6 | $r$ | $\rightarrow e$ 4,5 |

## Modus Tollens

$$\frac{\phi \rightarrow \psi \qquad \neg\psi}{\neg\phi} \; MT$$

# Rules for Implication Elimination

Elimination $\dfrac{\phi \quad \phi \to \psi}{\psi} \to e$

## Can $p, p \to q, p \to (q \to r)$ infer $r$

| | | |
|---|---|---|
| 1 | $p \to (q \to r)$ | premise |
| 2 | $p \to q$ | premise |
| 3 | $p$ | premise |
| 4 | $q \to r$ | $\to e$ 1,3 |
| 5 | $q$ | $\to e$ 2,3 |
| 6 | $r$ | $\to e$ 4,5 |

## Modus Tollens

$$\frac{\phi \to \psi \qquad \neg\psi}{\neg\phi} \; MT$$

Show: $p \to (q \to r), p, \neg r \vdash \neg q$

# Rules for Implication Introduction

$$\frac{\boxed{\begin{array}{c}\phi \\ \vdots \\ \psi\end{array}}}{\phi \to \psi} \to \mathrm{i}$$

# Rules for Implication Introduction

$$\frac{\boxed{\begin{array}{c}\phi \\ \vdots \\ \psi\end{array}}}{\phi \rightarrow \psi} \rightarrow \text{i}$$

**Show $(q \rightarrow r) \rightarrow ((\neg q \rightarrow \neg p) \rightarrow (p \rightarrow r))$**

# Rules for Implication Introduction

$$\frac{\boxed{\begin{array}{c}\phi\\ \vdots\\ \psi\end{array}}}{\phi \to \psi} \to \mathrm{i}$$

**Show $(q \to r) \to ((\neg q \to \neg p) \to (p \to r))$**

| | | |
|---|---|---|
| 1. | $q \to r$ | Assumption |
| 9. | $(\neg q \to \neg r) \to (p \to r)$ | |
| 10. | $(q \to r) \to ((\neg q \to \neg p) \to (p \to r))$ | $\to$ i 1-9 |

# Rules for Implication Introduction

$$\frac{\boxed{\begin{array}{c}\phi \\ \vdots \\ \psi\end{array}}}{\phi \to \psi} \to \text{i}$$

**Show $(q \to r) \to ((\neg q \to \neg p) \to (p \to r))$**

| | | |
|---|---|---|
| 1. | $q \to r$ | Assumption |
| 2. | $\neg q \to \neg p$ | Assumption |
| 8. | $p \to r$ | |
| 9. | $(\neg q \to \neg r) \to (p \to r)$ | $\to$ i 2-8 |
| 10. | $(q \to r) \to ((\neg q \to \neg p) \to (p \to r))$ | $\to$ i 1-9 |

# Rules for Implication Introduction

$$\frac{\boxed{\begin{matrix}\phi \\ \vdots \\ \psi\end{matrix}}}{\phi \rightarrow \psi} \rightarrow \text{i}$$

## Show $(q \rightarrow r) \rightarrow ((\neg q \rightarrow \neg p) \rightarrow (p \rightarrow r))$

| | | |
|---|---|---|
| 1. | $q \rightarrow r$ | Assumption |
| 2. | $\neg q \rightarrow \neg p$ | Assumption |
| 3. | $p$ | |
| | $r$ | |
| 8. | $p \rightarrow r$ | |
| 9. | $(\neg q \rightarrow \neg r) \rightarrow (p \rightarrow r)$ | $\rightarrow$ i 2-8 |
| 10. | $(q \rightarrow r) \rightarrow ((\neg q \rightarrow \neg p) \rightarrow (p \rightarrow r))$ | $\rightarrow$ i 1-9 |

# Rules for Implication Introduction

$$\frac{\boxed{\begin{array}{c}\phi \\ \vdots \\ \psi\end{array}}}{\phi \to \psi} \to \mathrm{i}$$

**Show $(q \to r) \to ((\neg q \to \neg p) \to (p \to r))$**

| | | | |
|---|---|---|---|
| 1. | $q \to r$ | | Assumption |
| 2. | $\neg q \to \neg p$ | | Assumption |
| 3. | $p$ | | Assumption |
| 4. 5. | $\neg\neg p, \neg\neg q$ | $\neg\neg i$ 3, $MT$ 2,4 | |
| 6. | $q$ | $\neg\neg e$ 5 | |
| 7. | $r$ | | $\to$ e 1-6 |
| 8. | $p \to r$ | | $\to$ i 3-7 |
| 9. | $(\neg q \to \neg r) \to (p \to r)$ | | $\to$ i 2-8 |
| 10. | $(q \to r) \to ((\neg q \to \neg p) \to (p \to r))$ | | $\to$ i 1-9 |

# Examples

Show following

1. $p \wedge q \rightarrow r \vdash p \rightarrow (q \rightarrow r)$

# Examples

Show following

1. $p \wedge q \to r \vdash p \to (q \to r)$
2. $(p \to (q \to r)) \to ((p \wedge q) \to r)$

# Examples

Show following

1. $p \wedge q \rightarrow r \vdash p \rightarrow (q \rightarrow r)$
2. $(p \rightarrow (q \rightarrow r)) \rightarrow ((p \wedge q) \rightarrow r)$
3. $p \rightarrow q \vdash (p \wedge r) \rightarrow (q \wedge r)$

# Thank You!

**Thank you very much for your attention!**

**Queries ?**

(Reference[2])

[2] 1) Book - *AIMA*, ch-07, Russell and Norvig. 2) Book - *Logic in CS*, ch-01, Mitchel Huth and Mark Ryan.