

Problem Solving with AI

Dr. Collin F. Lynch

AI Academy: North Carolina State University

Copyright 2021: Dr. Collin F. Lynch



Agenda

Course Overview

Foundations

Problems

Agents

Agent Types

Course Overview	Foundations	Problems	Agents	Agent Types

Course Overview

Course Overview	Foundations	Problems	Agents	Agent Types
<h2>Broad Goals</h2>				

- ▶ Cover foundational *concepts* of AI.
concepts
- ▶ Highlight potential *applications* for AI tools.
potential applications
- ▶ Cover *algorithms*, *techniques* and *design patterns* for AI applications.
algorithms techniques design patterns

Foundations

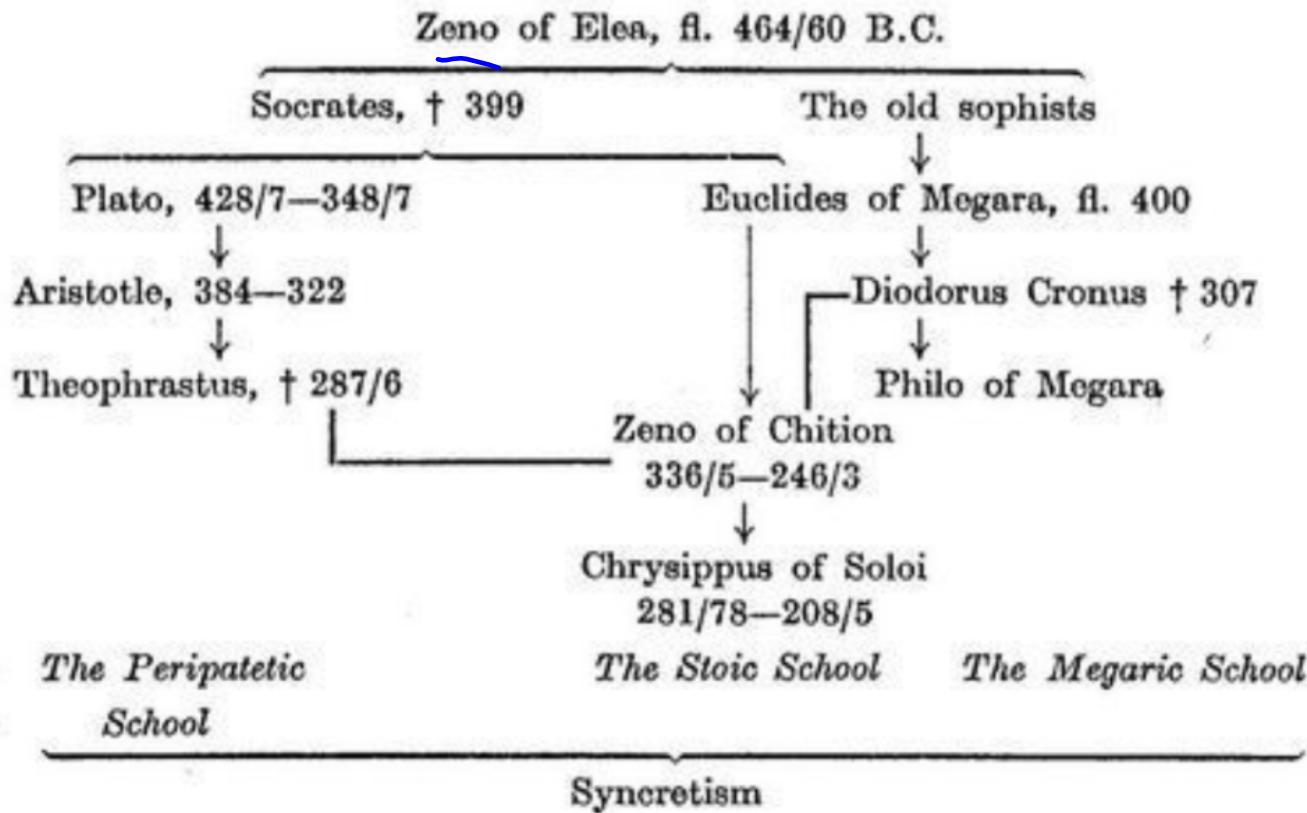
of AI
where does this come from.

Origins



Credit: www.filmeducation.org

Logic



Credit: historyoflogic.com

Applications

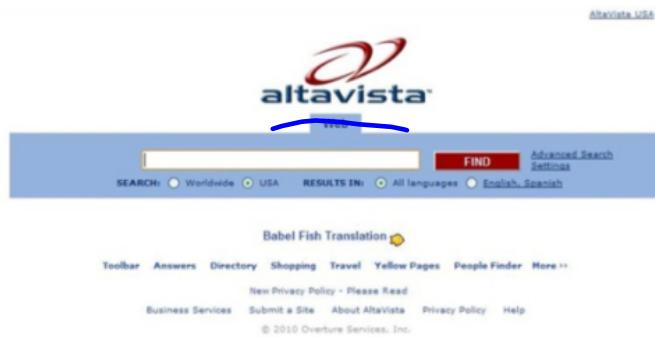


Image Credits: vipinonline.com callcredit.co.uk agweb.com techcrunch.com

The Boxes

Goals Problems.

Predictive Agent.

<p>“Machines with minds”</p> <p>Think like humans.</p>	<p>“Computations that perceive reason and act.”</p> <p>Think rationally.</p>
<p>“Machines that perform functions that require intelligence when done by people.”</p> <p>Act like humans</p>	<p>“Intelligent behavior in artifacts.”</p> <p>Act rationally</p>

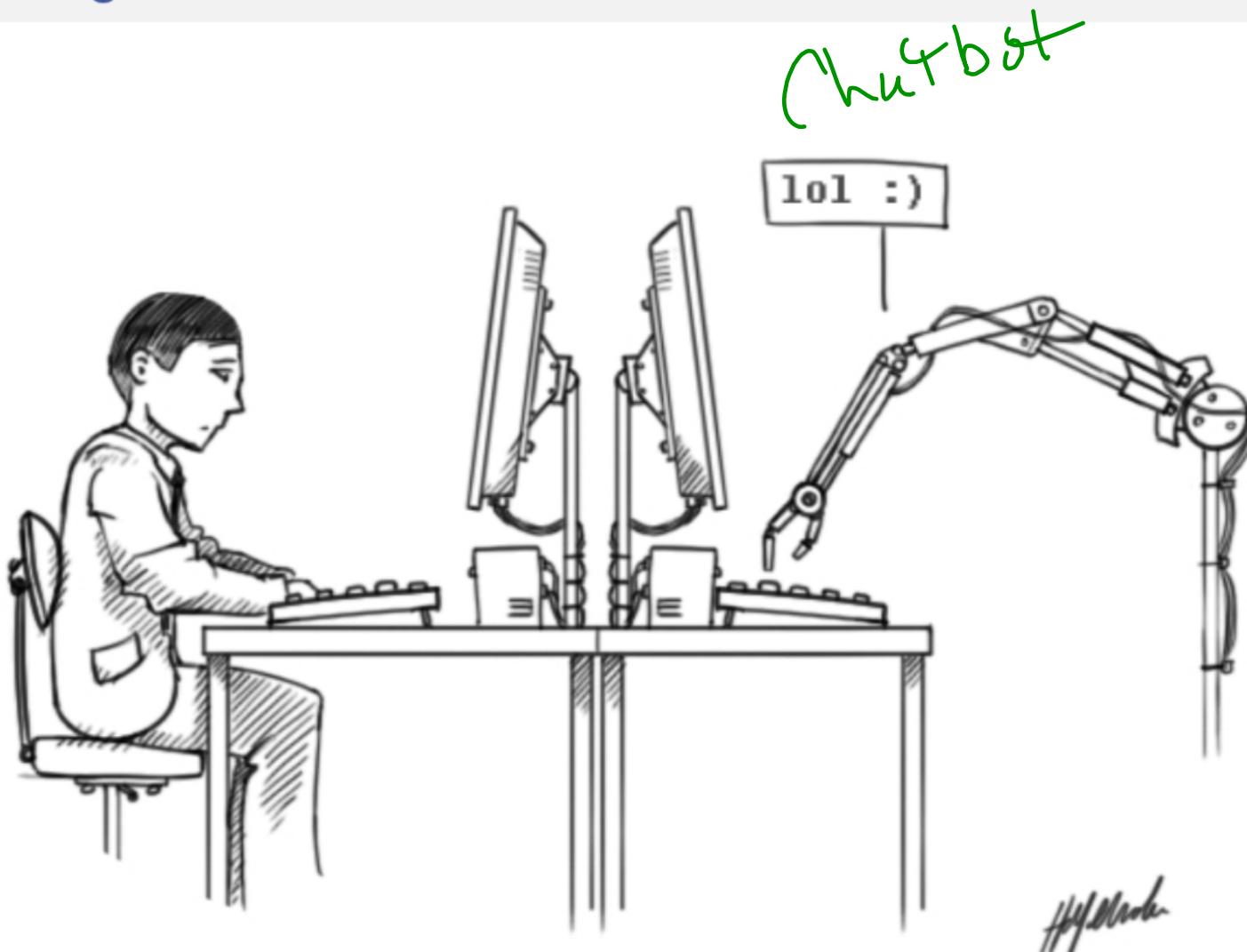
Cybernetics
Robotology

Pragmatism



Image: mathworks.com

The Turing Test



Course Overview	Foundations	Problems	Agents	Agent Types

Student: Can machines think?

Course Overview	Foundations	Problems	Agents	Agent Types

Student: Can machines think?

Master: Can submarines swim?

Course Overview	Foundations	Problems	Agents	Agent Types

Problems

What defines a problem?

Applications
Self-Driving Car
Automotive Soln
Chassis

Problem Types

- Well-Structured (Well-Defined) (Turing Recognizable)

- Ill-Defined

- Wicked Problems

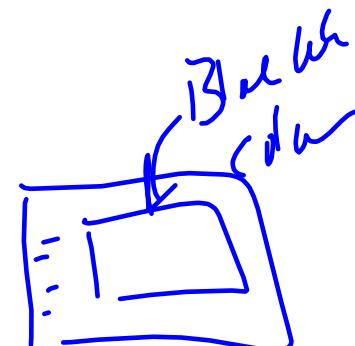
- Toy or Puzzle:

- Atomic;
- Observable;
- Deterministic;
- Completely Known.

get board

thin Expertise Frame

Policy Political



Beginning

Problem Solving

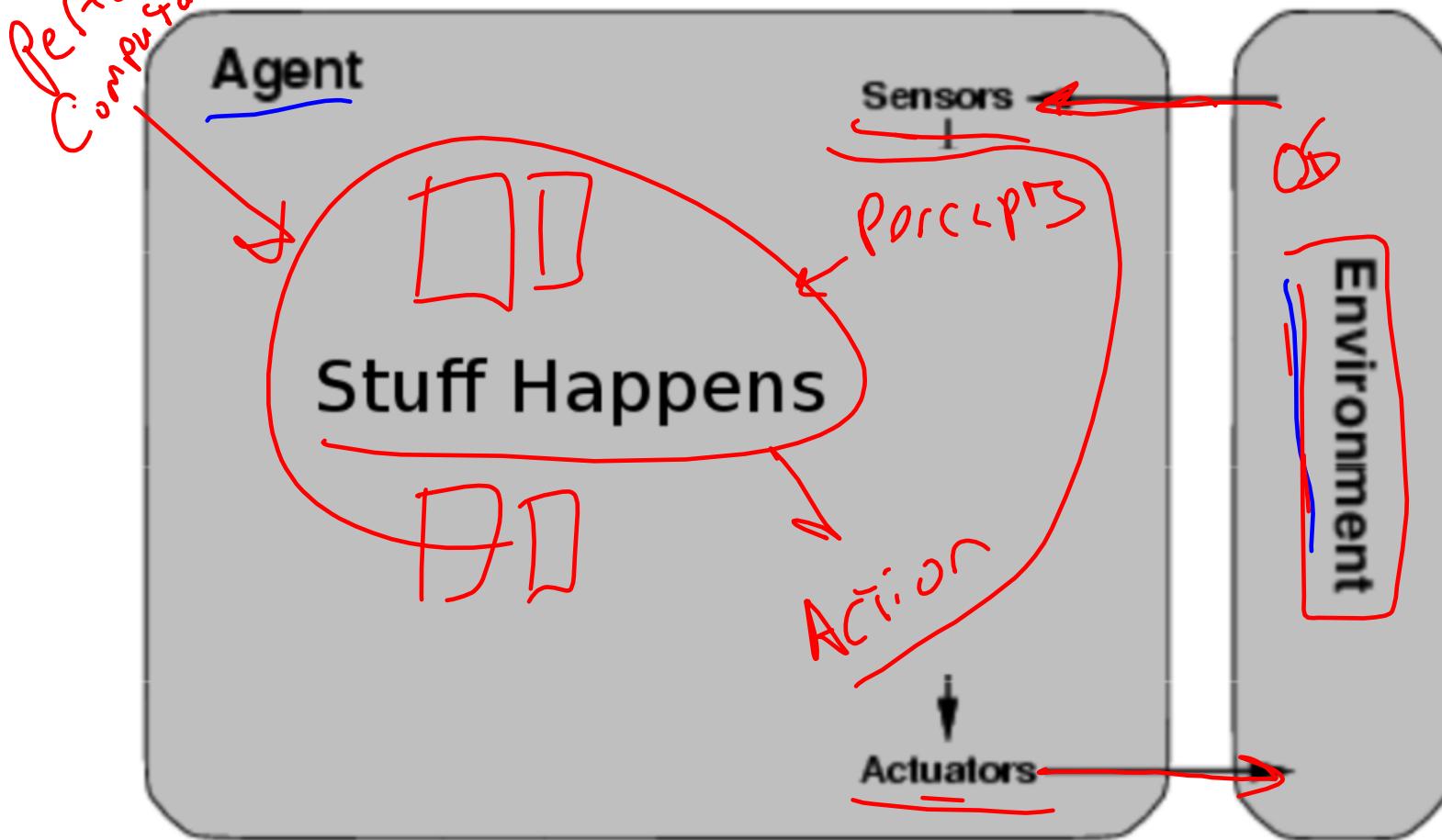
1. Define a representation of the problem (abstraction).
2. Solve the problem by searching for a solution.
3. Execute the planned solution.
4. Evaluate or defend your results and recompute.

Course Overview	Foundations	Problems	Agents	Agent Types

Agents

Agent
Performance
Configurable

API → causes
Operational
Conditional
Platform

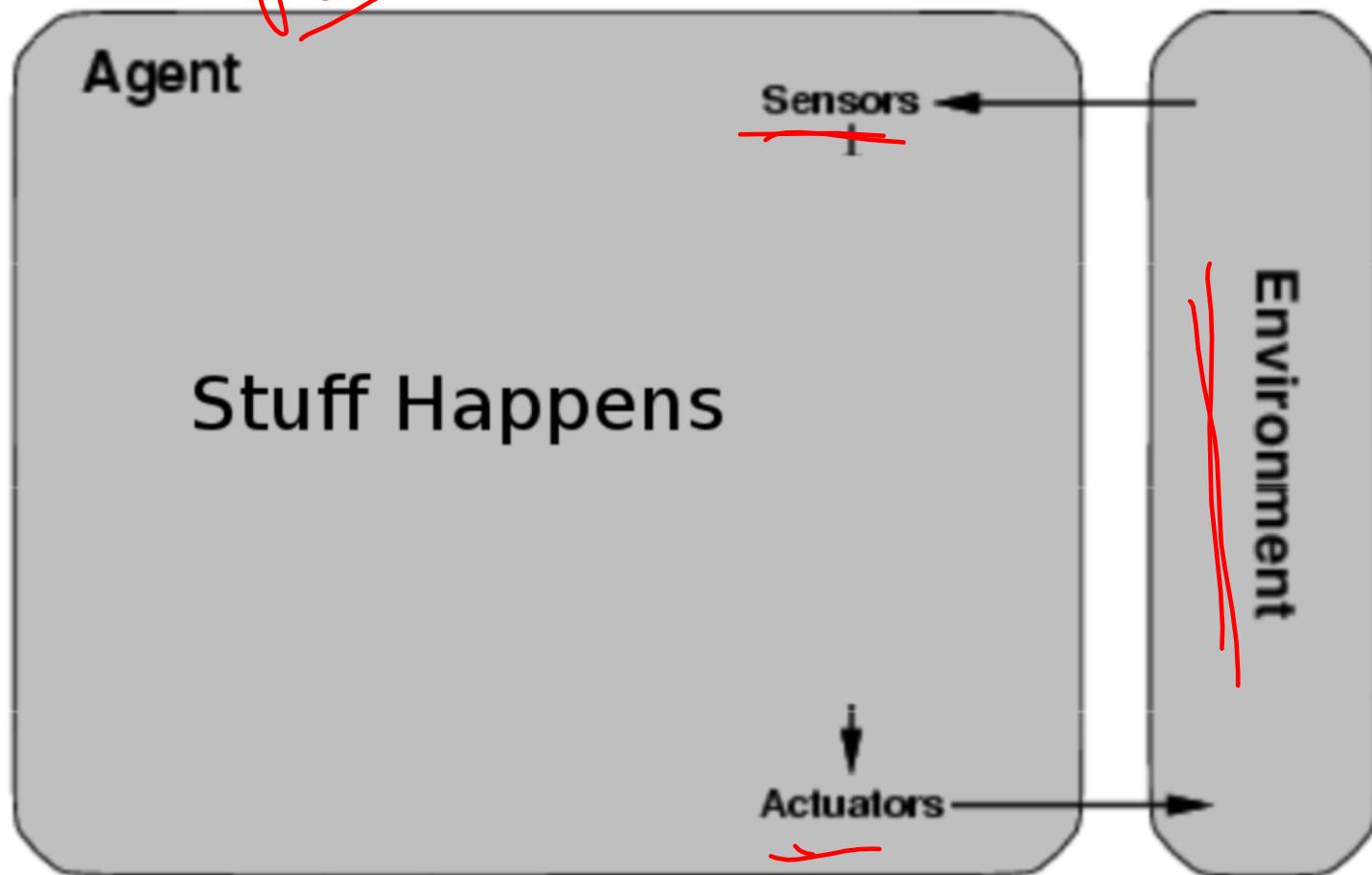


Aspects

- ▶ Separation:
 - ▶ The agent is distinct from the environment.
 - ▶ The sensors and actuators provide the interface.
 - ▶ These are distinct from the percepts and actions which are internal.
- ▶ Architecture agent hardware.
- ▶ Function $a : p_0, \dots, p_n \rightarrow a_i$
- ▶ Program (implementation)

Context

- ▶ Performance measure for success.
- ▶ Agent's prior knowledge.
- ▶ The available actions.
- ▶ Percept sequence to date.

PEAS*Specifying
Pest*

Course Overview	Foundations	Problems	Agents	Agent Types

Pragmatism

Big Idea: Satisficing

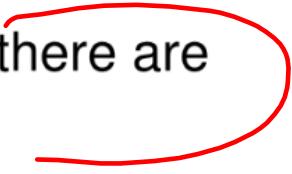
Customer Happiness

Efficiency

Rationality

- ▶ Strong rationality rests on a basic assumption:
 - ▶ “Reasonable people all think the same,... if they think”

 - ▶ This idea rests at the foundation of economics.

 - ▶ And much other discussion, the basic idea that there are good processes of reasoning and clear values.


Rationality

- ▶ Strong rationality rests on a basic assumption:
 - ▶ “Reasonable people all think the same,... if they think”
 - ▶ This idea rests at the foundation of economics.
 - ▶ And much other discussion, the basic idea that there are good processes of reasoning and clear values.
 - ▶ Needless to say humans don't do this.
 - ▶ and AI is more limited.

Rational Agents

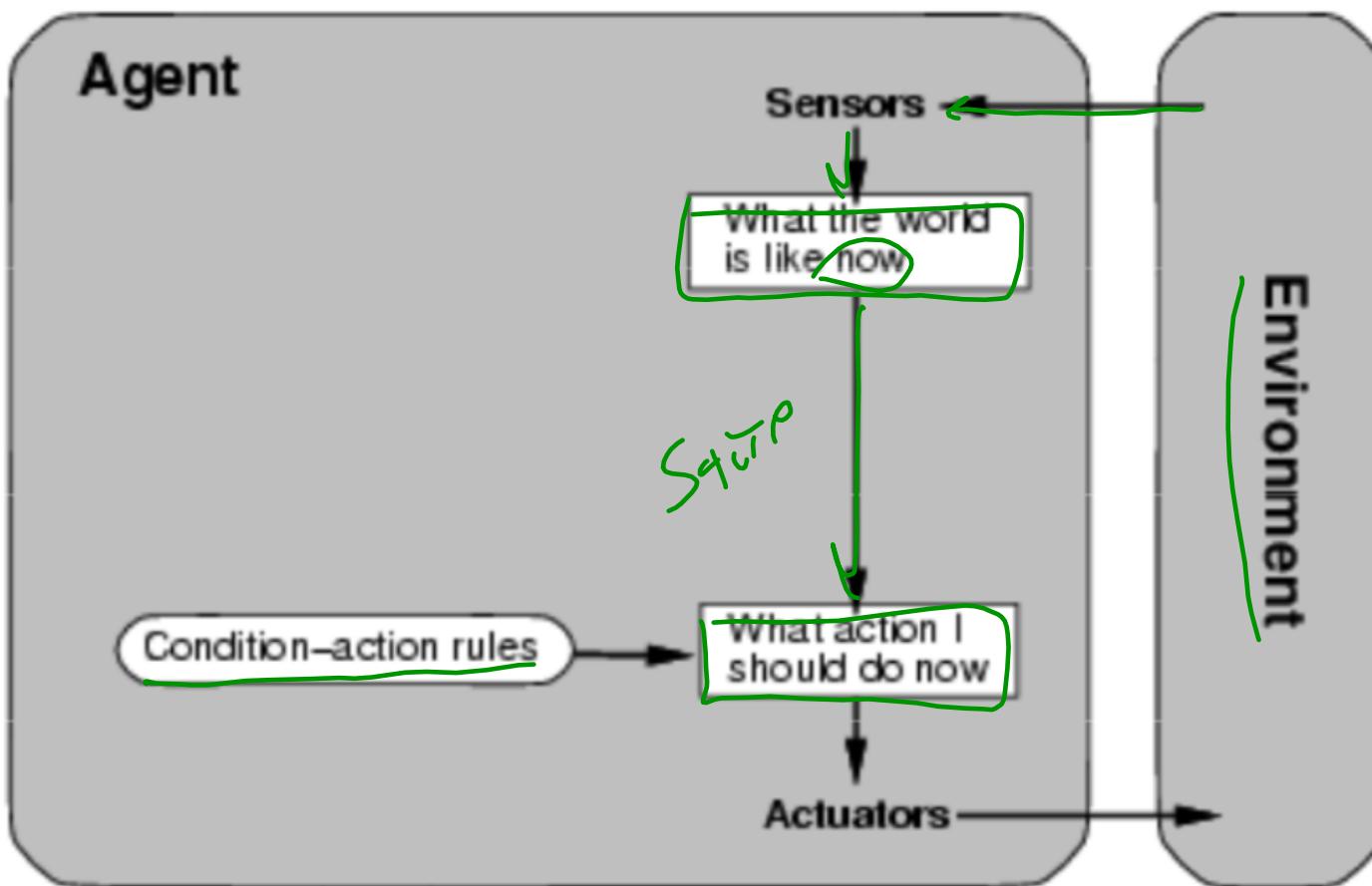
A rational agent seeks to maximize it's
performance given it's current context.
(Rationality ≠ Omniscience)

Agent Types

Agent Types

- ▶ Simple Reflex
- ▶ Model-based reflex
- ▶ Goal-based
- ▶ Utility-based

Simple Reflex Agent



Simple Reflex Agent: Schema

```
def agent_func(Percept):
    Rules = {"p0": "a0", "p1": "a3", ...}
    Action = Rules[Percept]
    return Action
```

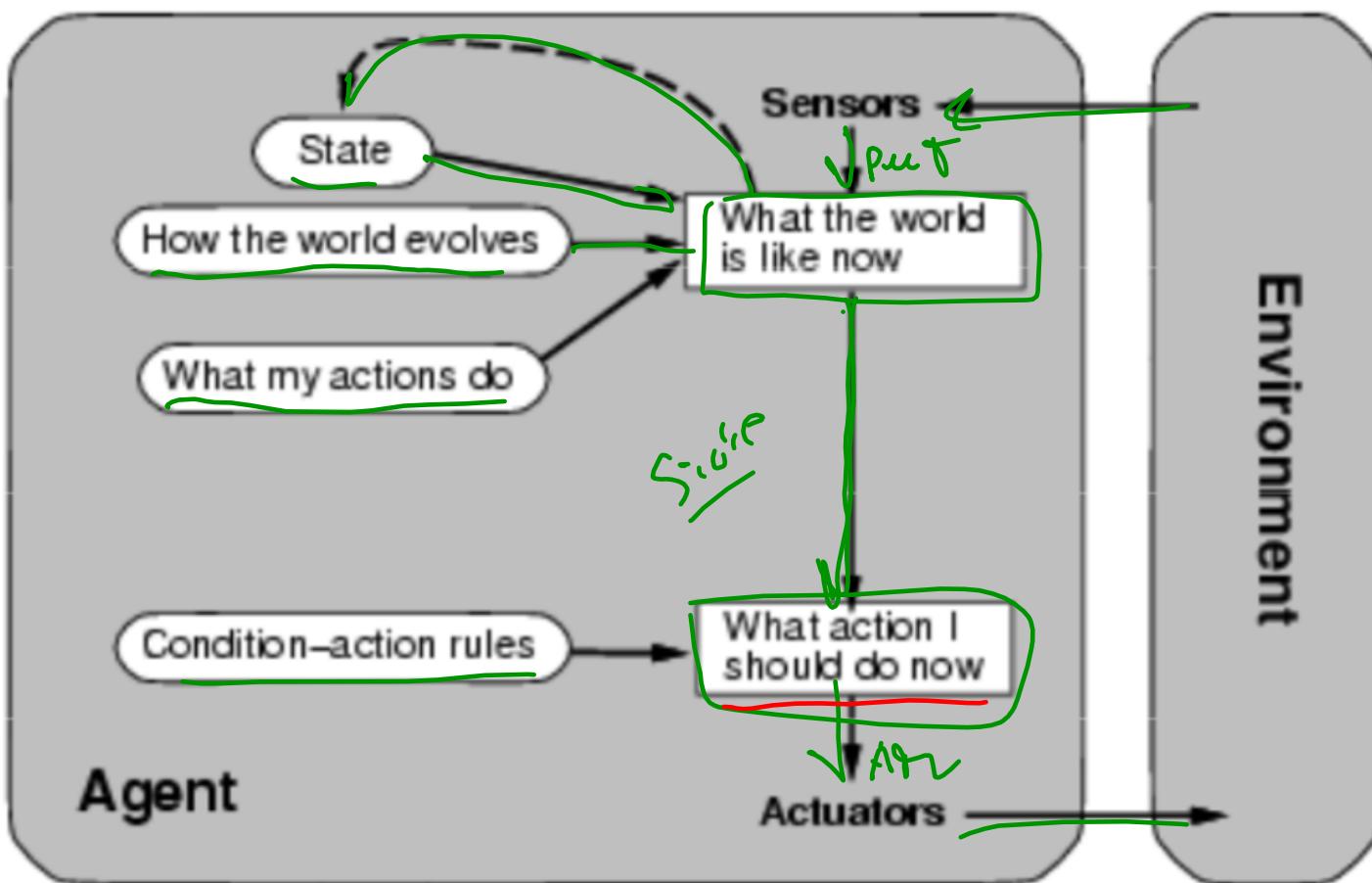
Hash

(Pseudo-Random) Simple Reflex Agent: Schema (2)

No Memory

```
def agent_func(Percept):
    Rules = {"p0" : ["a0", "a4", ...],
    . .
    "p1" : ["a3", ...]}
    Actions = Rules[Percept]
    Choice = random.choice(Actions)
    return Choice
```

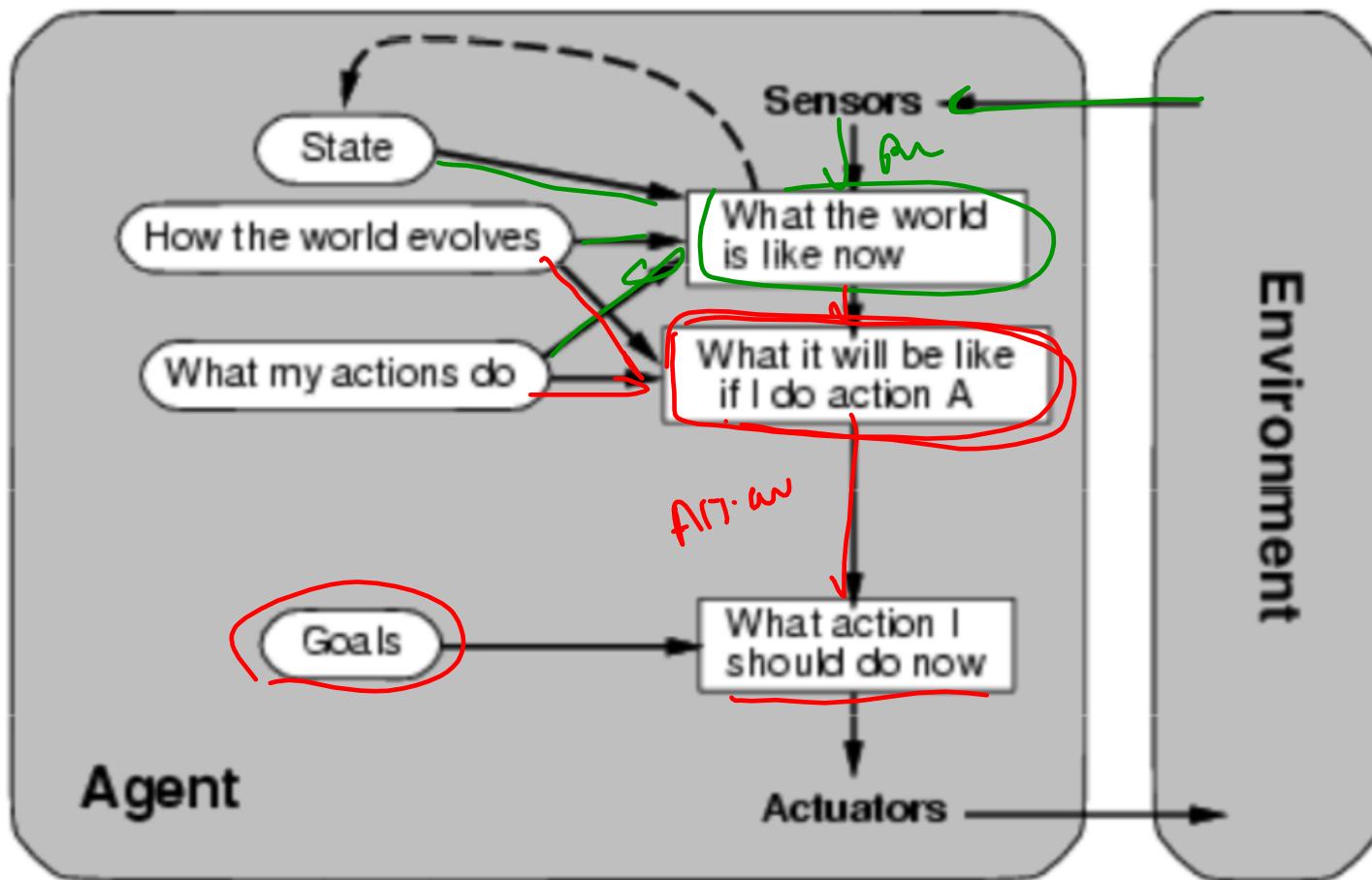
Model-Based Reflex Agent



Model-Based Reflex Agent: Schema

```
def agent_func(Percept, Curr_State, Last_Action):
    Rules = {"s0" : "a0", "s1" : "a3", ...}
    New_State = update_state(Curr_State, Last_Action,
                             Percept, Model)
    Action = Rules[New_State]
    return (Action, New_State)
```

Goal-Based Agent

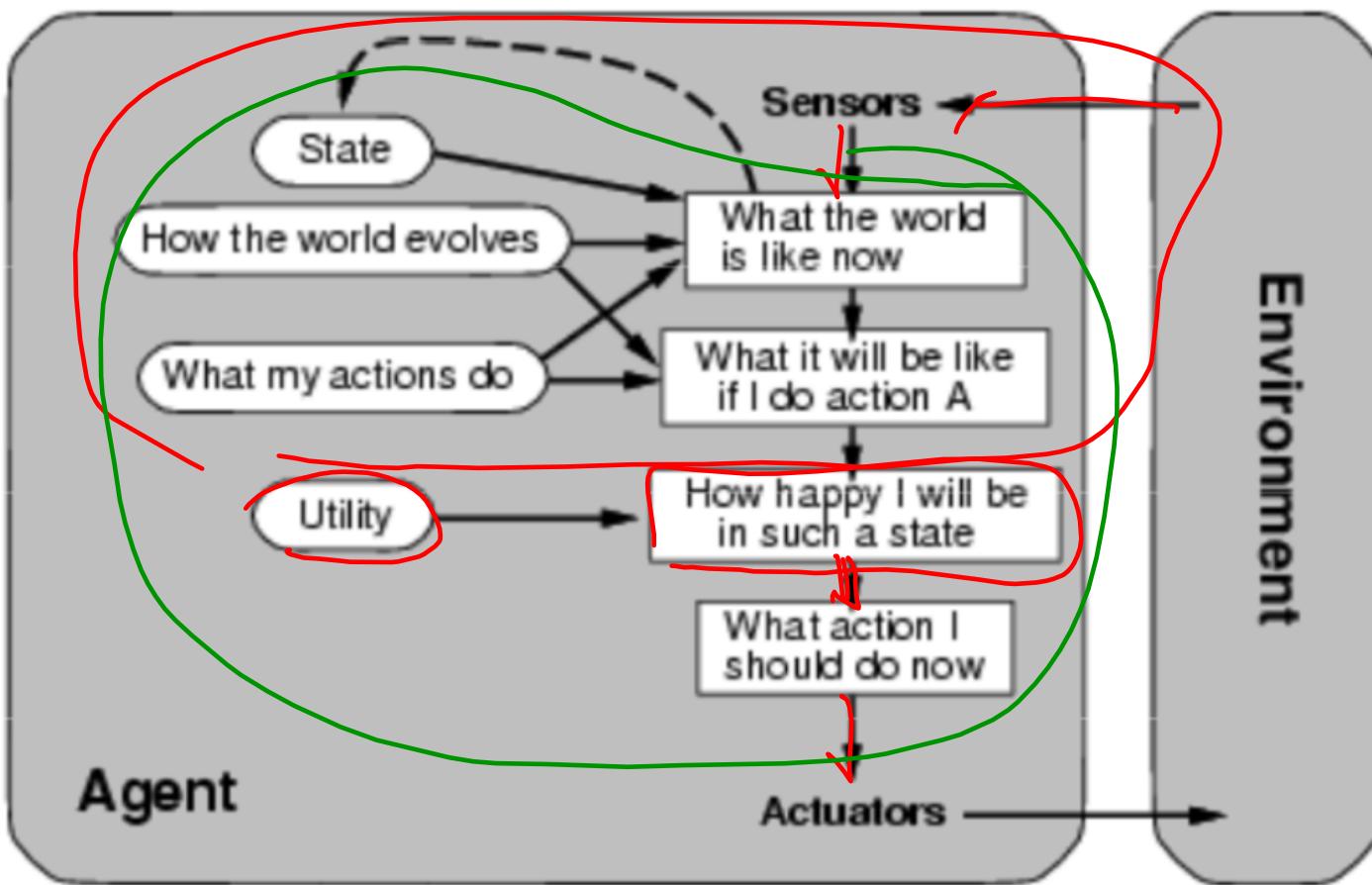


Goal-Based Agent: Schema

```
def agent_func(Percept, Curr_State, Last_Action):
    New_State = update_state(Curr_State, Last_Action,
                             Percept, Model)
    for Act in Possible Actions:
        Possible_State = check_act(New_State, Act)
        if (Possible_State == Goal_State):
            return (Act, New_State)
```

curr?
Goals

Utility-Based Agent



Utility-Based Agent: Schema

```
def agent_func(Percept, Curr_State, Last_Action):
    New_State = update_state(Curr_State, Last_Action,
                             Percept, Model)
    Potentials = [check_act(New_State, A)
                  for A in Possible Actions]
    Potentials.sort()
    return Potentials[0]
```

Course Overview

Foundations

Problems

Agents

Agent Types

Why Utility?

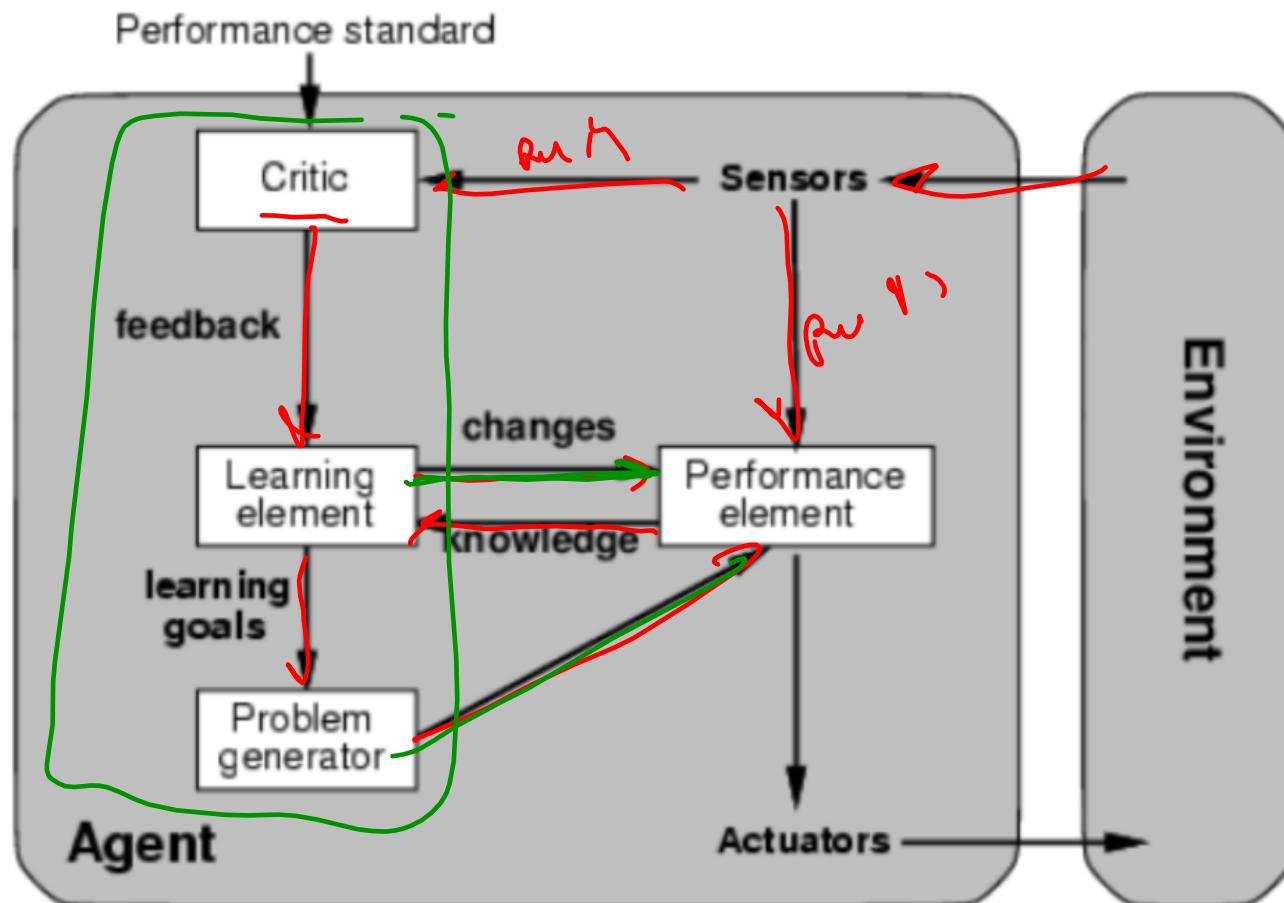
Why Utility?

- ▶ Sometimes there is no goal per-se.
- ▶ Sometimes the goal changes.
- ▶ Or the environment does.

Uncertainty.

- ▶ Certainty know Known Risk
- ▶ Uncertainty — Known Unknown
- ▶ Risk — Unknown Unknown

Learning Agent



Learning Agent: Schema

```
def update_func(\n    Percept, Curr_State, Last_Action, Perf_Standard):\n    Est = Perf_Standard(\n        Percept, Curr_State, Last_Action)\n    (Goals, Changes) = Learning_Elt.give_feedback(Est)\n    New_Problems = Problem_Generator.gen_prob(Goals)\n    (Knowledge, Next_Action) = agent_Func(\n        Percept, Curr_State, Last_Action, Changes)\n    Learning_Element.add_knowledge(Knowledge)\n    return(Next_Action)
```

Wumpus World

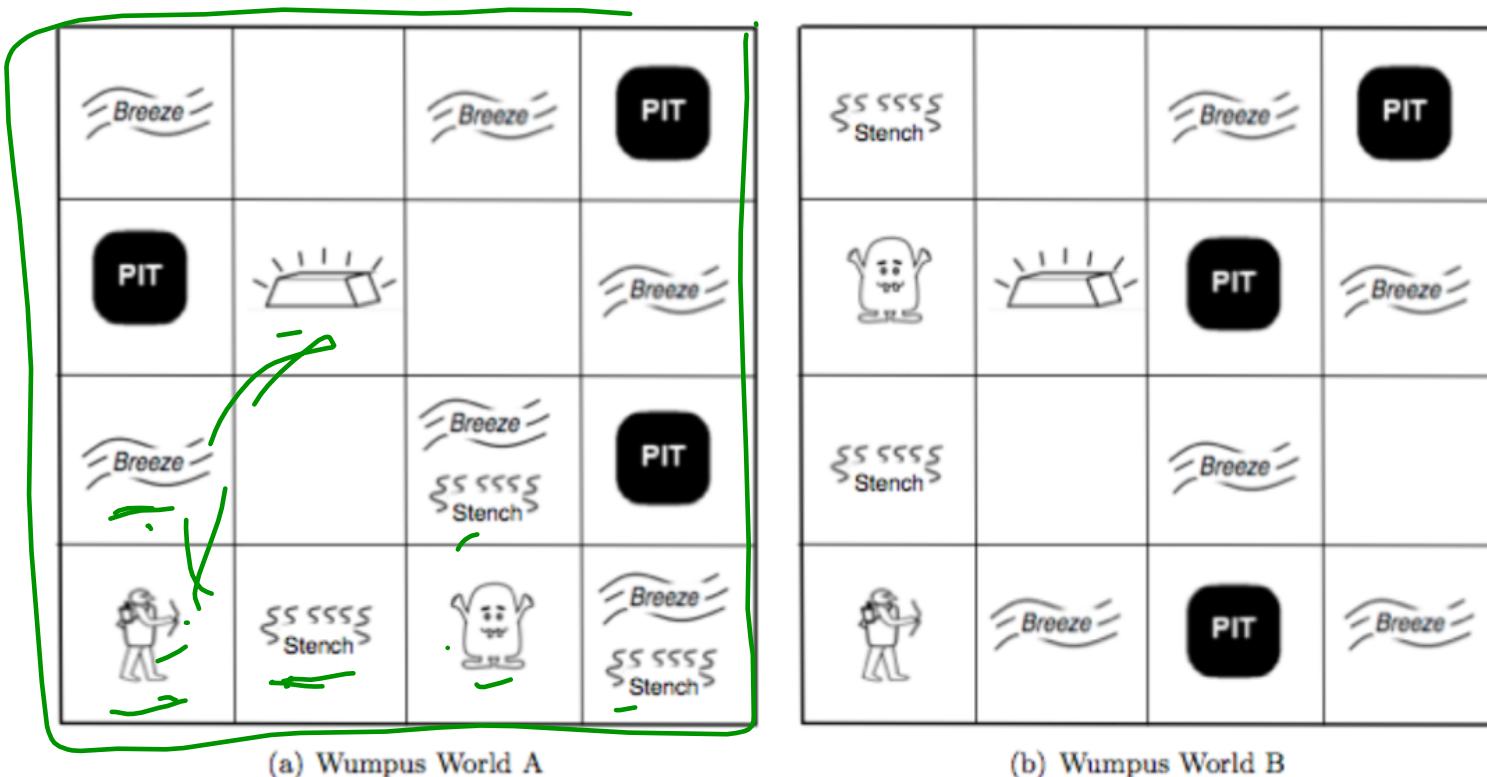


Figure 1: Two Instances of the Wumpus World (AI: A Modern Approach (Russell and Norvig))

Course Overview	Foundations	Problems	Agents	Agent Types
<h2>Ethics</h2>				

- ▶ Ethics has already started.

Ethics

- ▶ Ethics has already started.
- ▶ **Applications:** what should be built?

Ethics

- ▶ Ethics has already started.
 - ▶ **Applications:** what should be built?
 - ▶ **Engineering:** how should we build it?
- 

Course Overview	Foundations	Problems	Agents	Agent Types
<h2>Ethics</h2>				

- ▶ Ethics has already started.
- ▶ **Applications:** what should be built?
- ▶ **Engineering:** how should we build it?
- ▶ **Implementation:** Can agents have ethics?

Course Overview	Foundations	Problems	Agents	Agent Types
<h2>Ethics</h2>				

- ▶ Ethics has already started.
- ▶ **Applications:** what should be built?
- ▶ **Engineering:** how should we build it?
- ▶ **Implementation:** Can agents have ethics?
- ▶ **Enforcement:** How can we enforce these rules?