LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Praktikum Autonome Systeme

An Introduction to Autonomous Systems

Prof. Dr. Claudia Linnhoff-Popien
Thomy Phan, Andreas Sedlmeier, Fabian Ritz
http://www.mobile.ifi.lmu.de

SoSe 2019

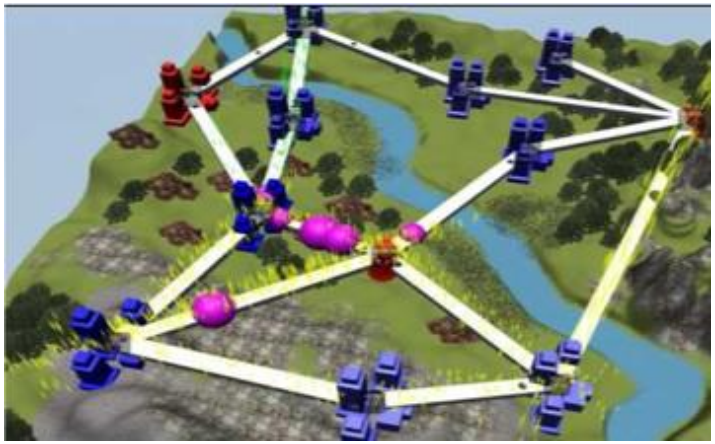
→ **Autonomous Systems**

What is an Autonomous System?

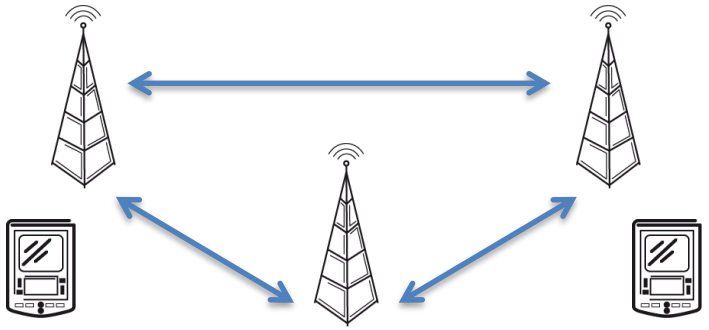
Definition: A system, which can operate without human intervention.

(Possible) Real-World Applications

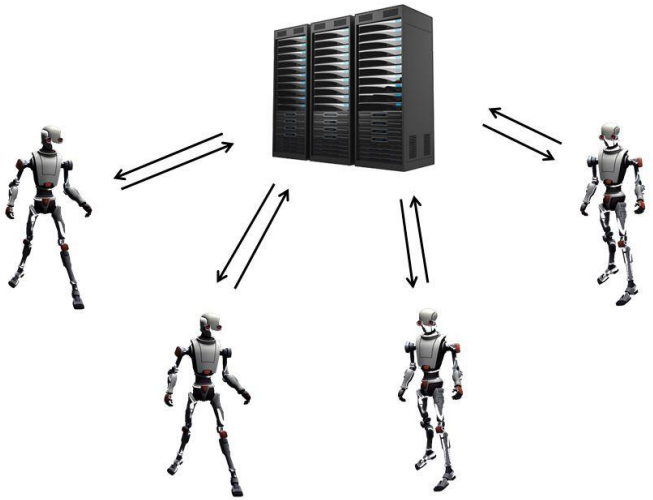
Smart Grids / Cities



Intelligent / Mobile Networks



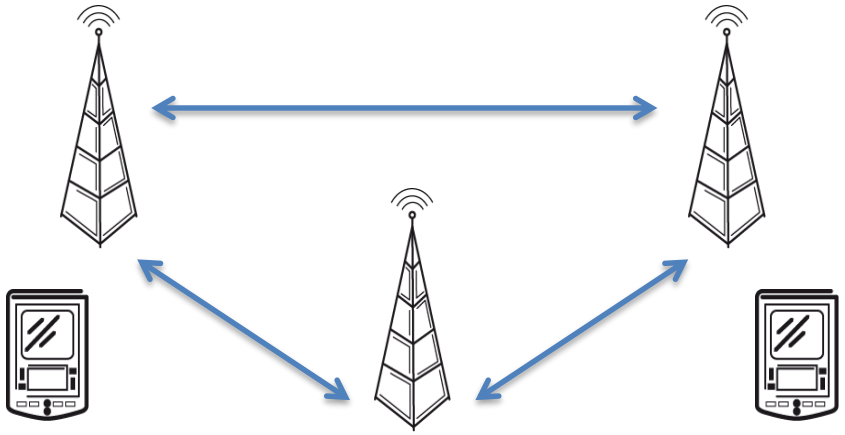
Robotics



Industry 4.0

Properties of Autonomous Systems

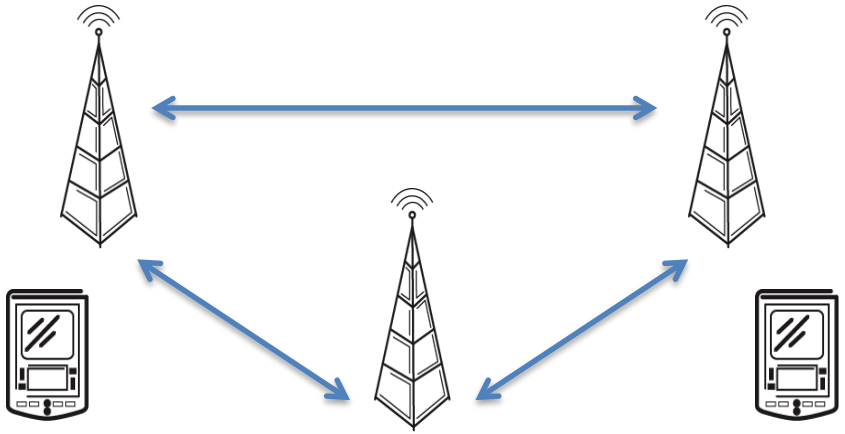
- Self-CHOP
  - Self-Configuration
  - Self-Healing
  - Self-Optimization
  - Self-Protection
- More Self-Properties
  - Self-Learning
  - Self-Organization
  - Self-Regulation
  - …



M. Salehie and L. Tahvildari, Autonomic Computing: Emergent Trends and Open Problems, ACM SIGSOFT Software Engineering Notes, 2005

Challenges of Autonomous Systems

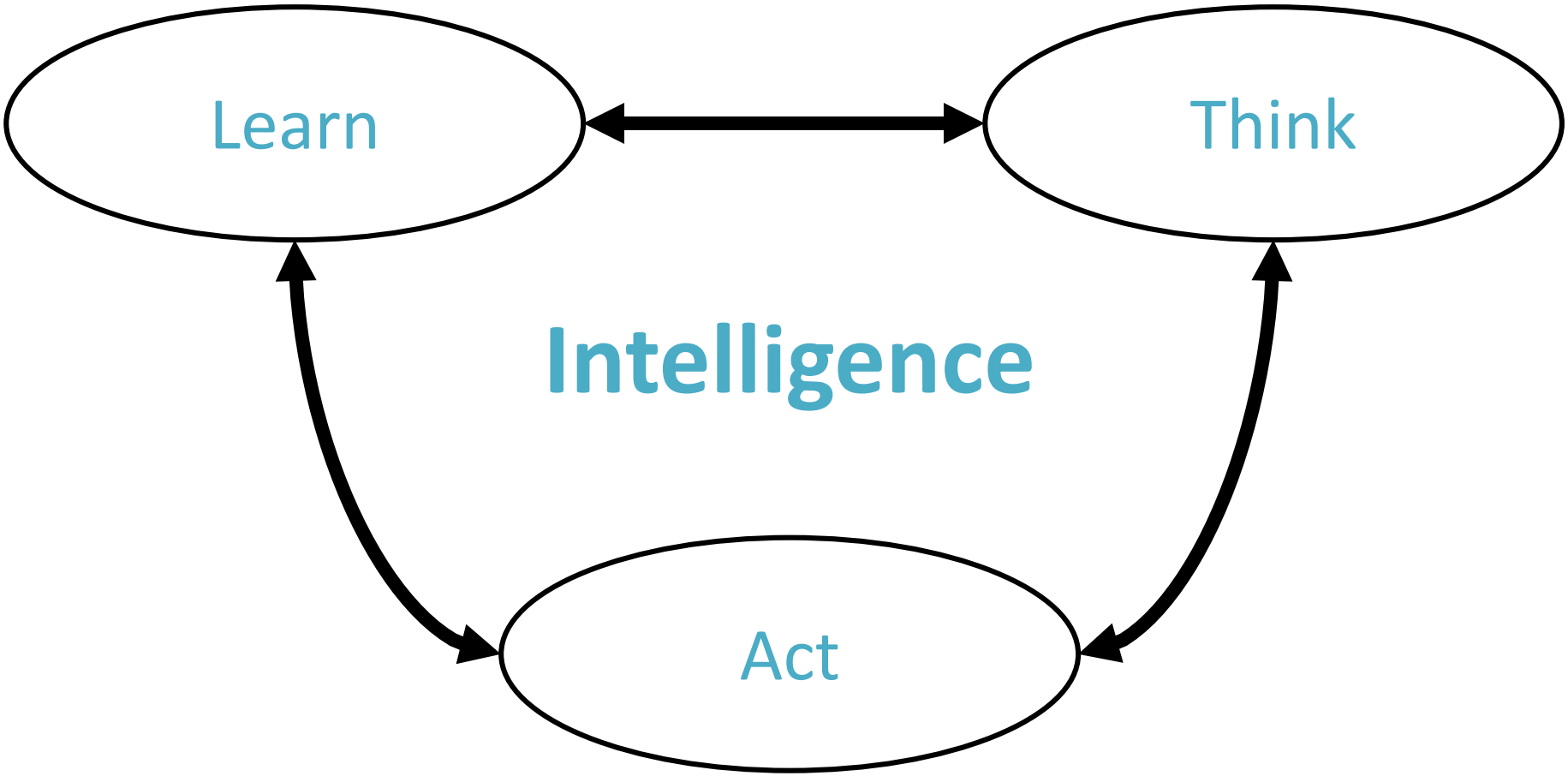
- Dynamic Environments
- High Complexity
- Many Constraints:
  - Perception
  - Computational and Memory Resources
  - Energy Consumption
  - Communication
- Safety and Risk
- Security
- Quality Management

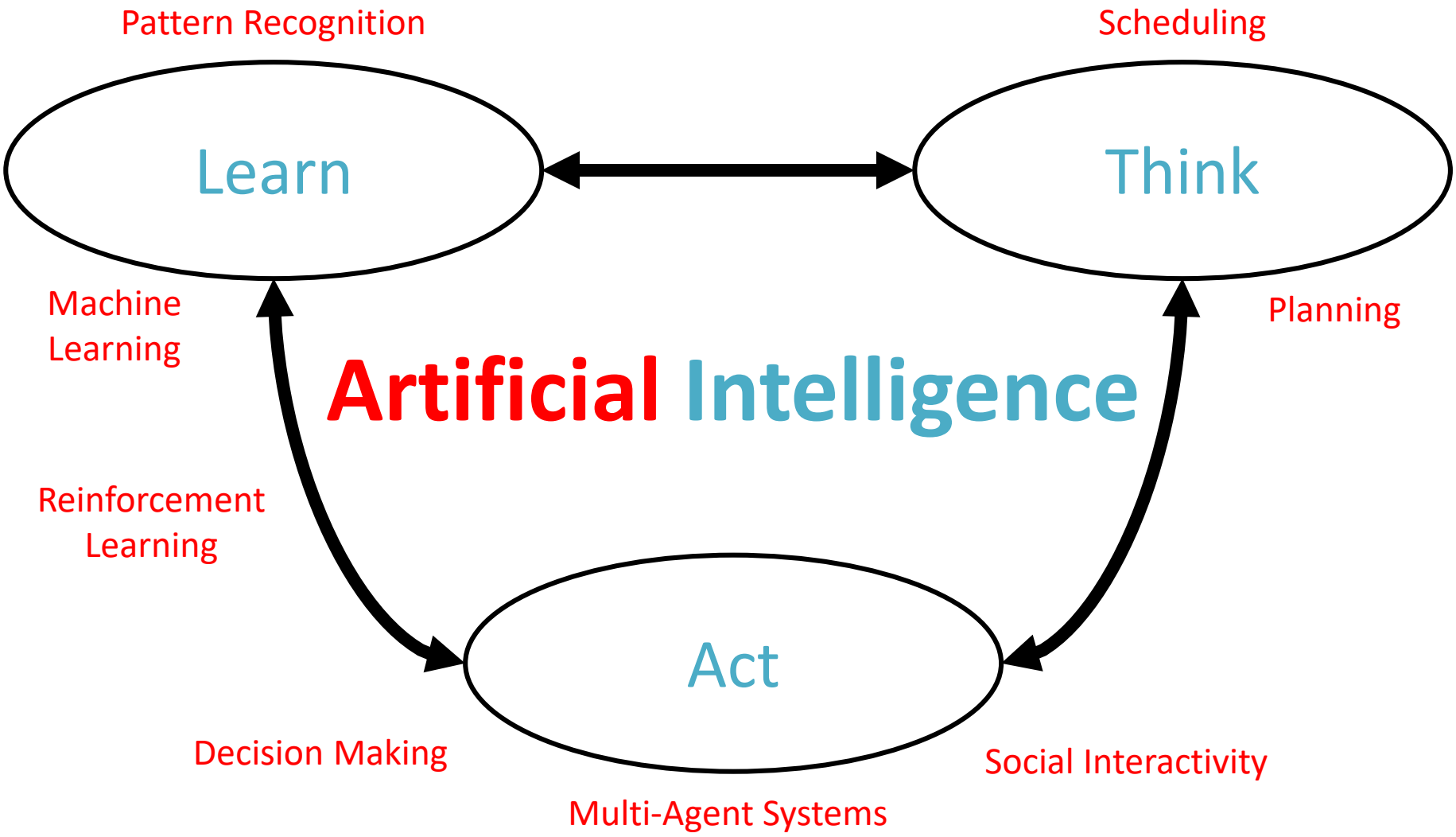


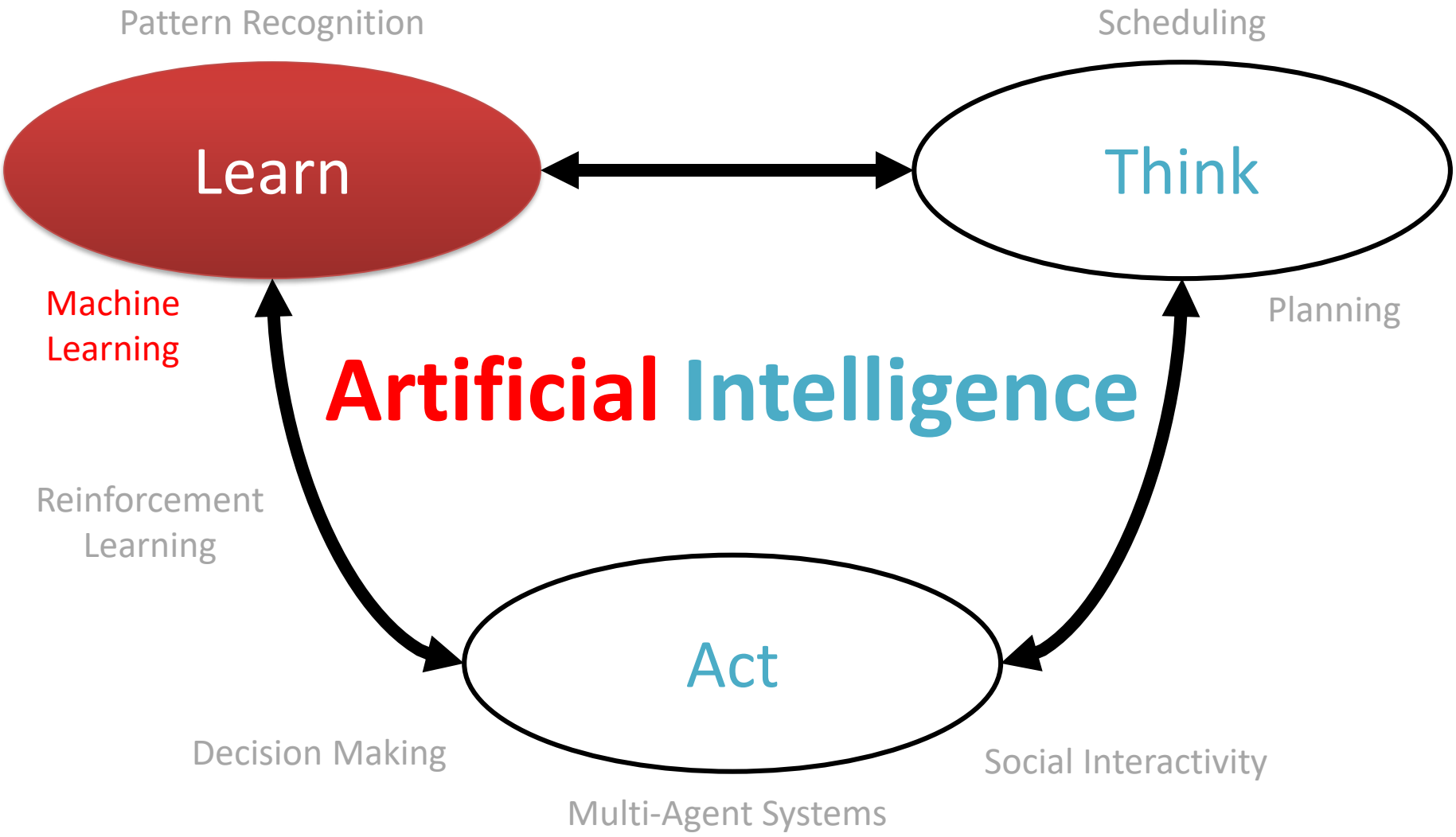
Challenges of Autonomous Systems

- Dynamic Environments
- High Complexity
- Many Constraints:
  - Perception
  - Computational and Memory Resources
  - Energy Consumption
  - Communication
- Safety and Risk
- Security
- Quality Management

→ **Artificial Intelligence**

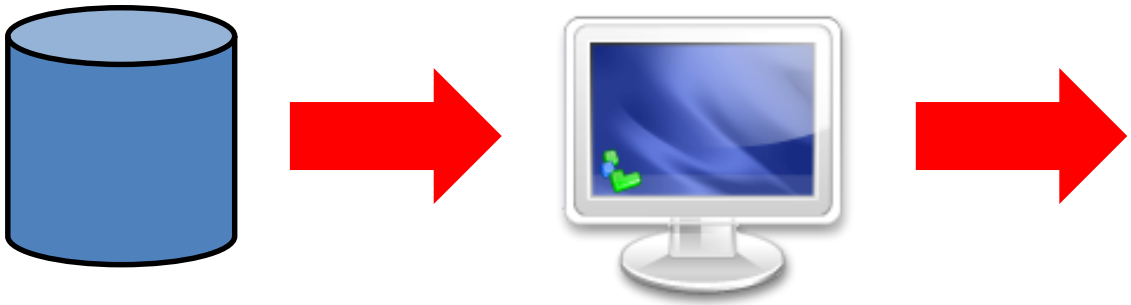

Learn
Think
Intelligence
Act


Pattern Recognition
Scheduling
Learn
Think
Machine Learning
Planning
Artificial Intelligence
Reinforcement Learning
Act
Decision Making
Social Interactivity
Multi-Agent Systems


Pattern Recognition
Scheduling
Learn
Think
Machine Learning
Planning
Artificial Intelligence
Reinforcement Learning
Act
Decision Making
Social Interactivity
Multi-Agent Systems

Machine Learning

- **Goal:** Create programs that learn how to solve complex problems
- Learn statistical models from **experience / data**



Object Recognition

Prediction

Control

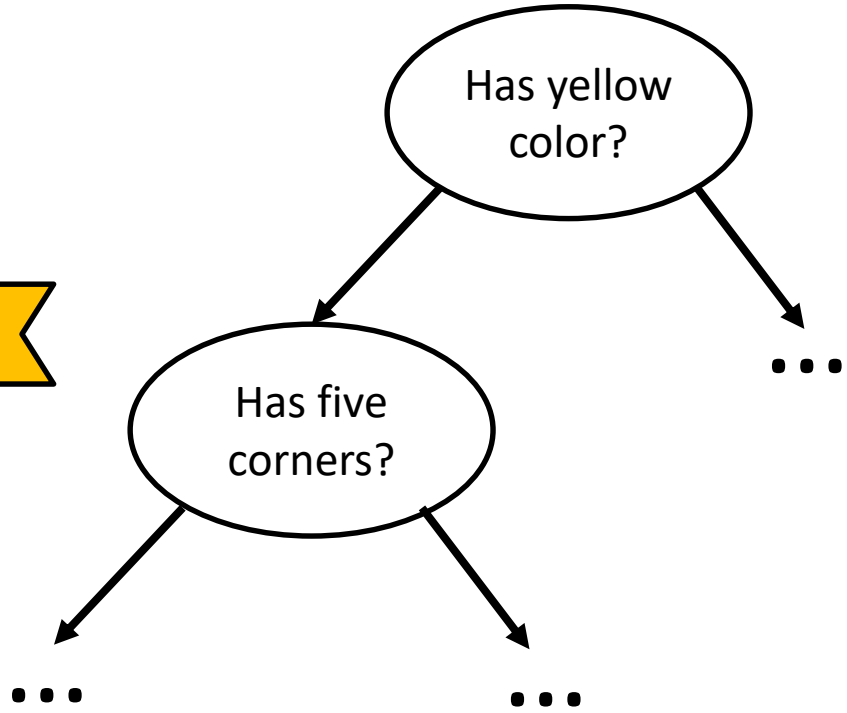
Compression

Data Generation

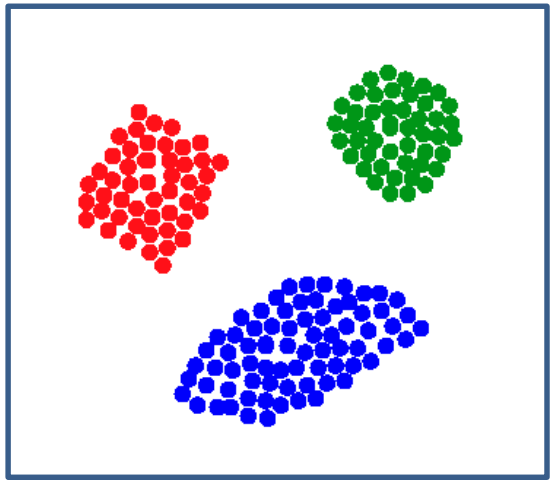
...

Why Machine Learning?

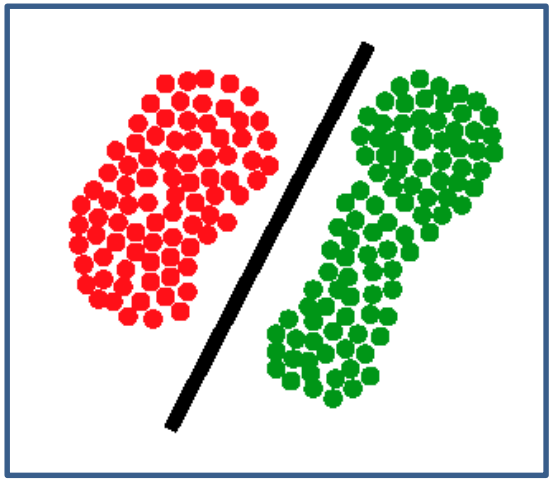
- **Goal:** Create programs that learn how to solve complex problems
- Many problems cannot be solved by engineering handcrafted solutions
  - Too many aspects to consider
  - Too many rules
  - Hard adaption to changes
  - Hard generalization
- **Example:**
  - How to classify a star?



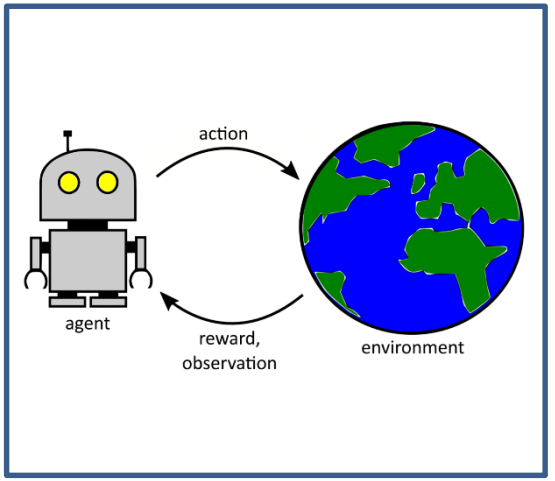
Types of Machine Learning



Unsupervised Learning



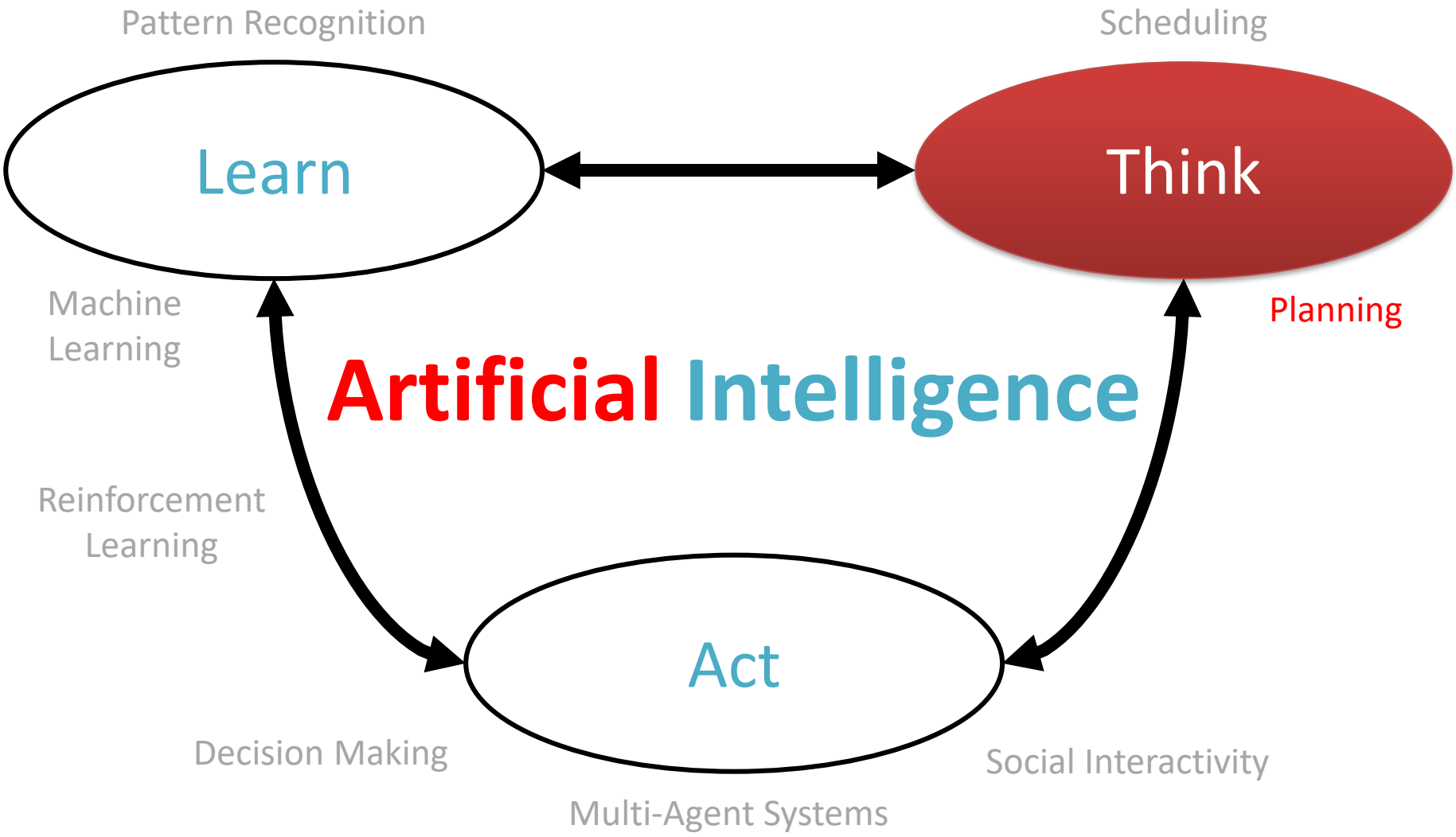
Supervised Learning



Reinforcement Learning

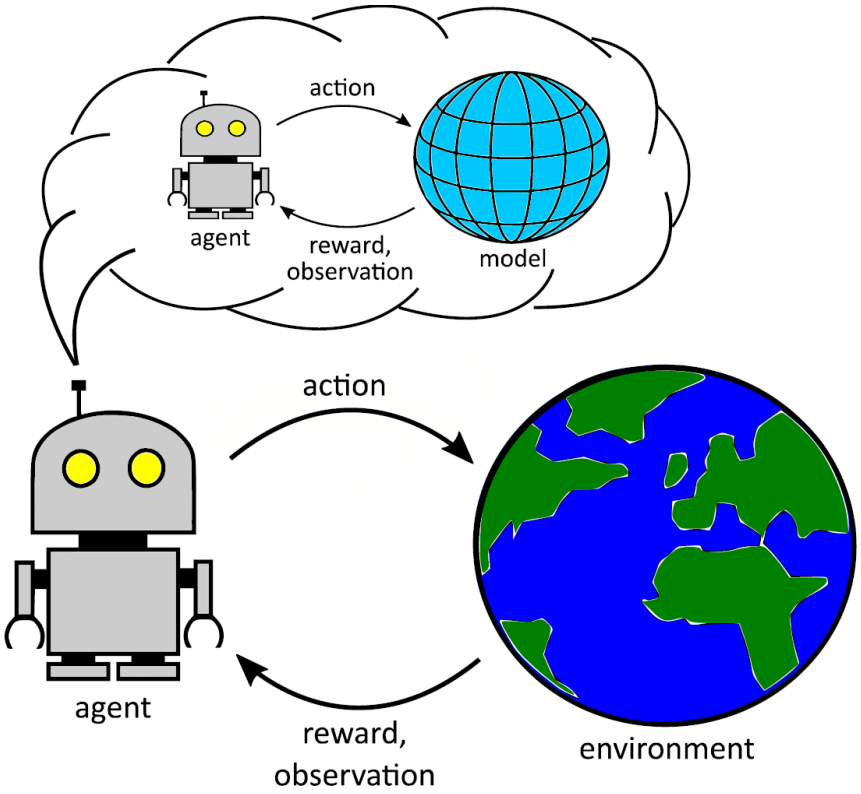
Challenges of Machine Learning

- Data Availability
- Data Complexity
- Efficiency
- Compactness
- Interpretability
- Robustness
- Adaptivity


Pattern Recognition
Scheduling
Learn
Think
Machine Learning
Planning
Artificial Intelligence
Reinforcement Learning
Act
Decision Making
Social Interactivity
Multi-Agent Systems

Automated Planning

- **Goal:** Find (near-)optimal strategies to solve complex problems
- Use (heuristic) lookahead search on a **given model** of the problem

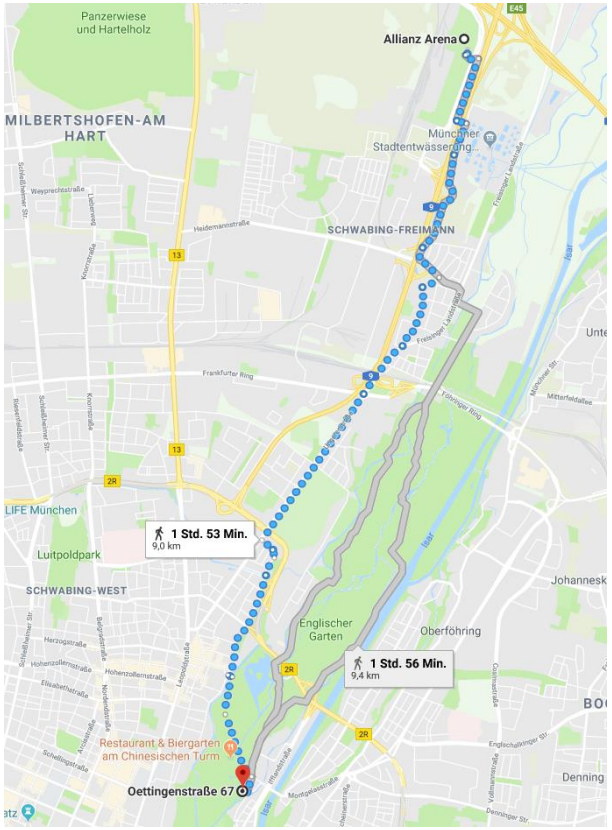


Why Automated Planning?

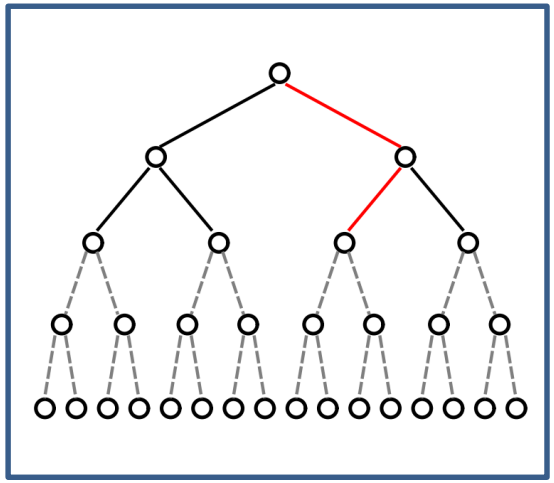
- **Goal:** Find (near-)optimal strategies to solve complex problems

- Planning is necessary, if **explicit reasoning** is required:
  - Consideration of risks and uncertainties
  - Consideration of hard constraints

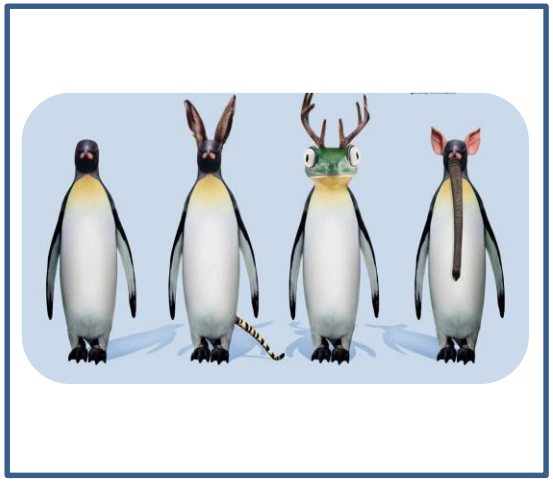
- Planning is **flexible**:
  - Use the same method for different problems by replacing the model
  - Search for multiple alternative strategies



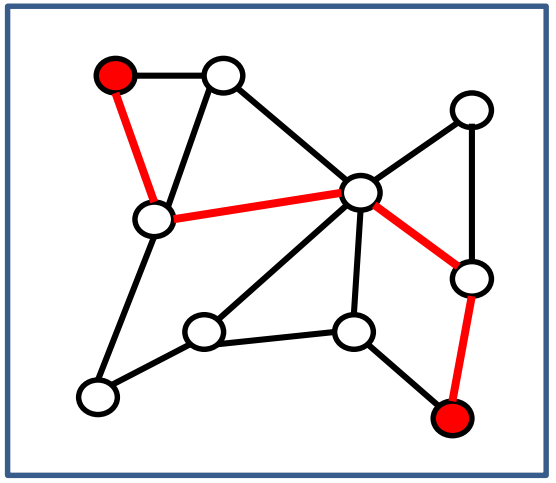
Planning Approaches (Examples)



Tree Search



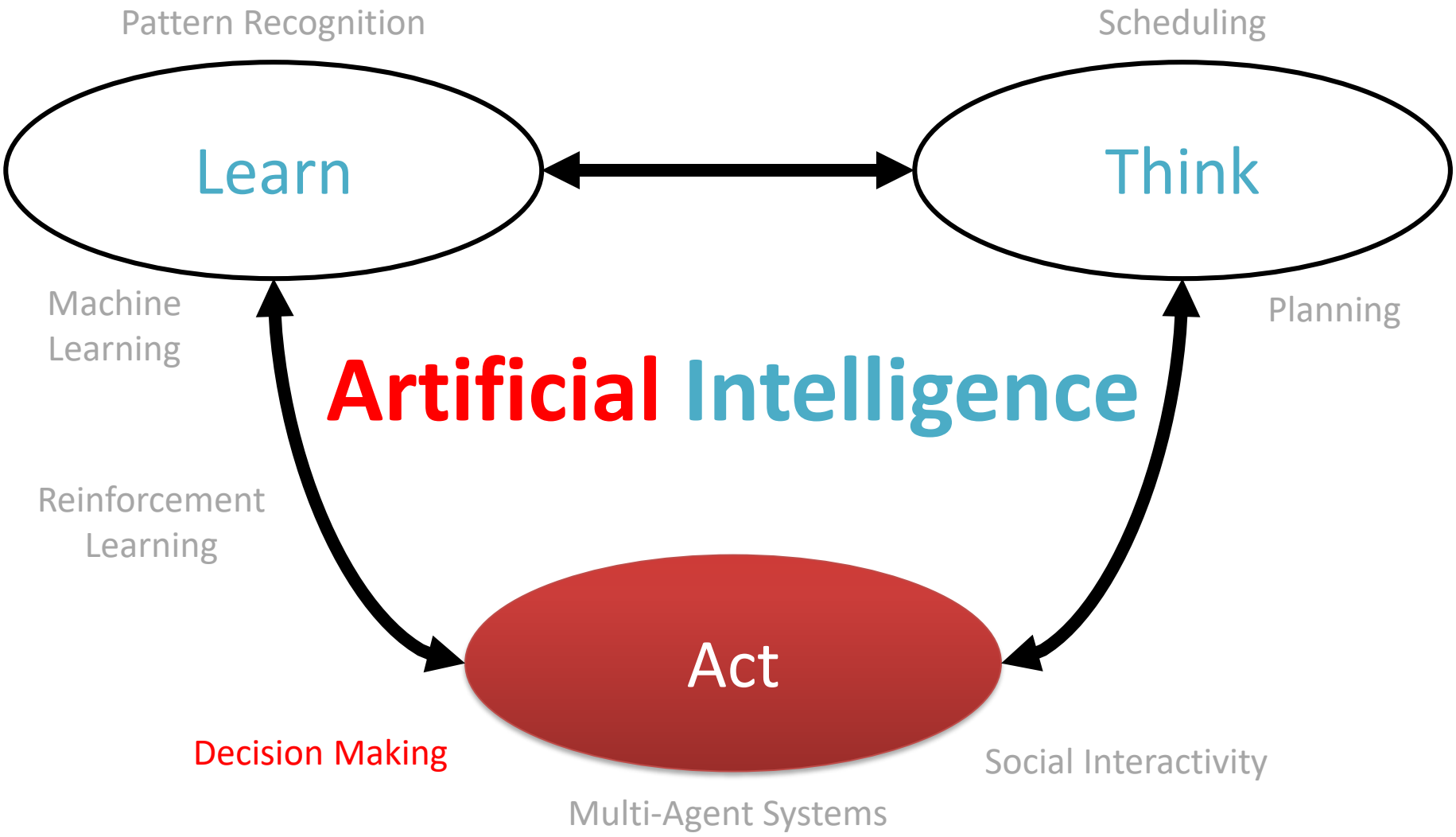
Evolutionary Computation



Dynamic Programming

Challenges of Automated Planning

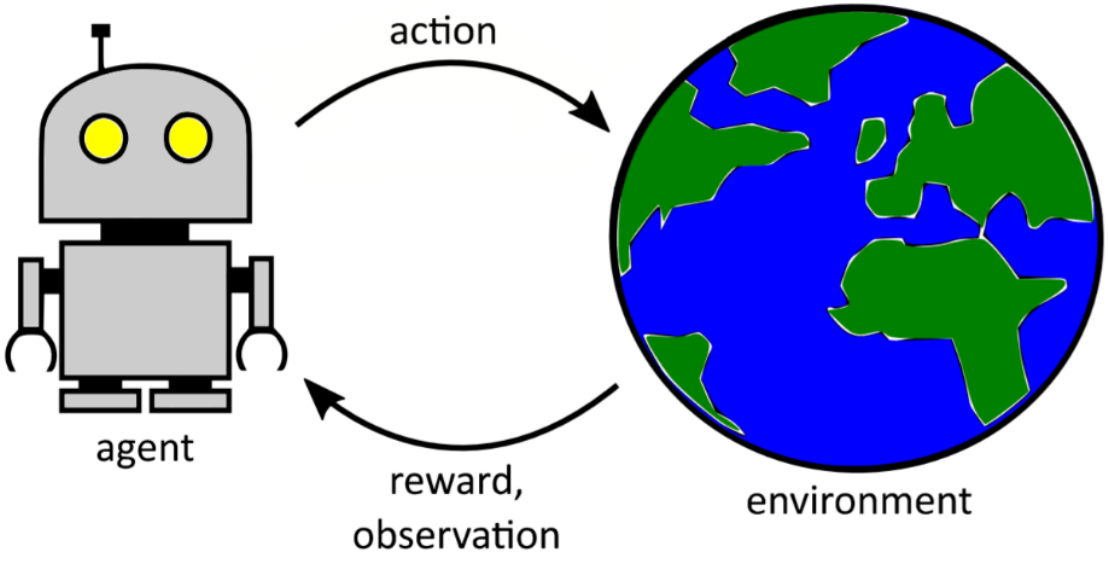
- Model Availability
- Model Uncertainty
- Computational and Memory Efficiency
- Real-time Planning


Pattern Recognition
Scheduling
Learn
Think
Machine Learning
Planning
Artificial Intelligence
Reinforcement Learning
Act
Decision Making
Social Interactivity
Multi-Agent Systems

→ Decision Making

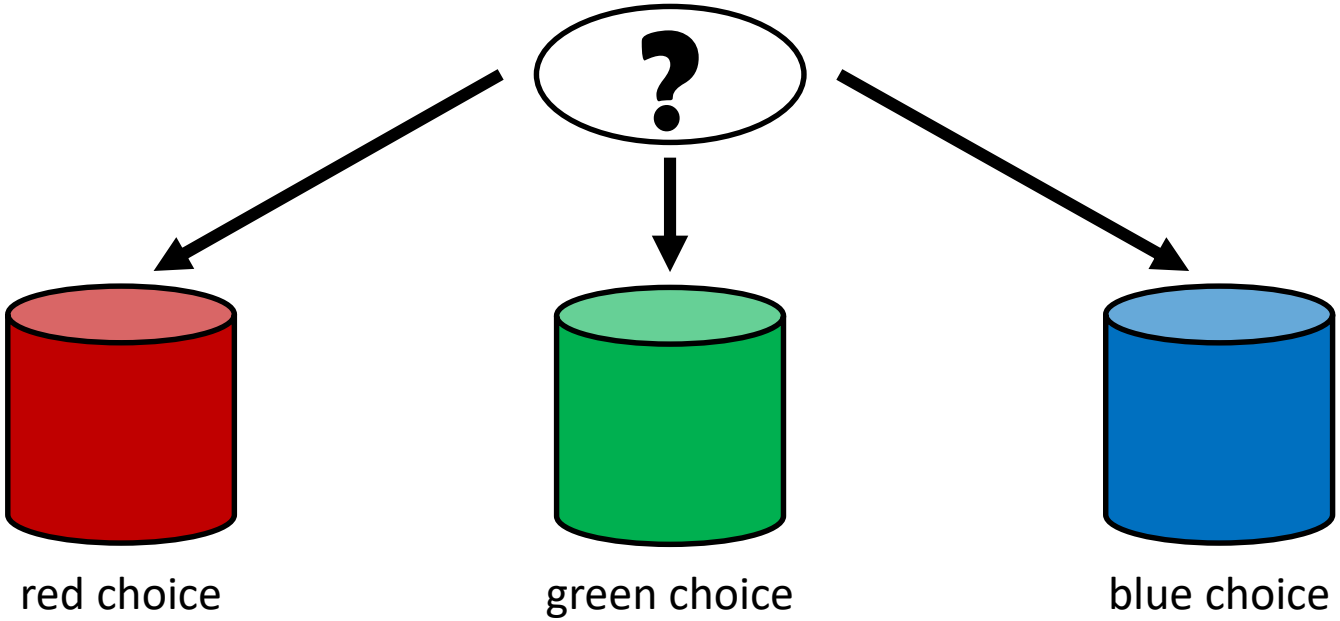
Decision Making

- **Goal:** Autonomously select actions to solve a (complex) task
  - time could be important (but not necessarily)
  - maximize the **expected reward** for each state



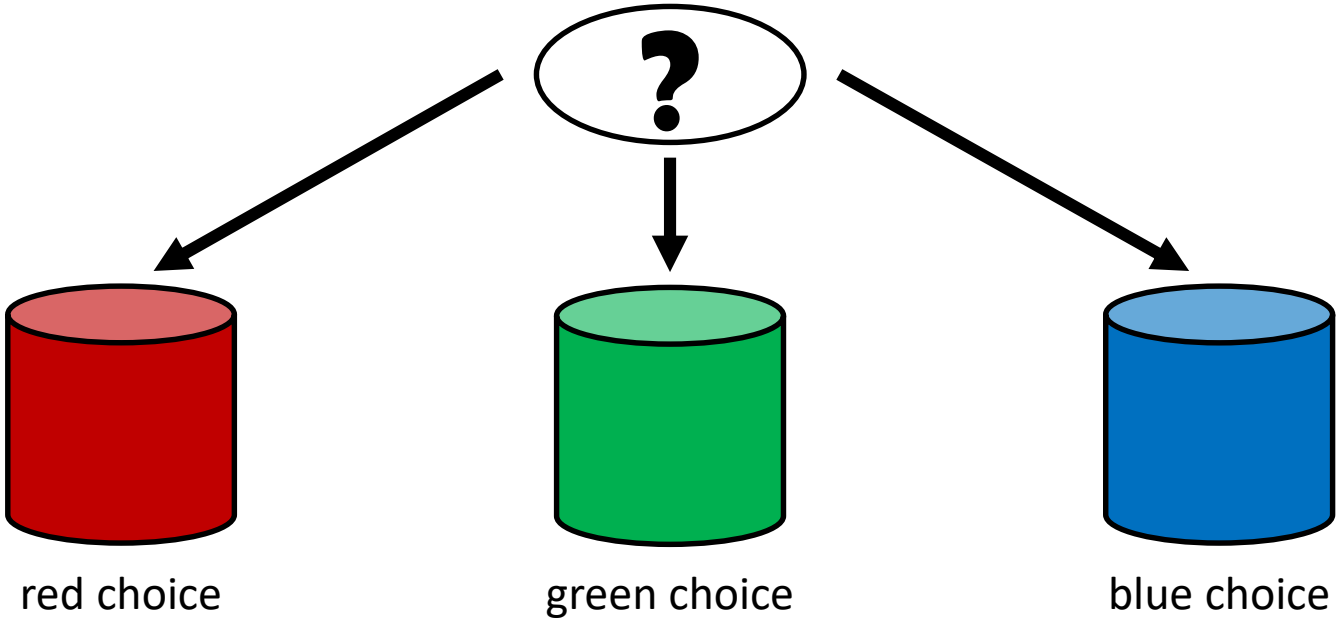
Decision Making Example

- Consider a situation, where you have to make a **choice**
- Each choice has its own **reward distribution** (which is unknown)
- Problems:
  - How to determine the **best choice**?
  - How to adapt to **changing** reward distributions?



Decision Making Example

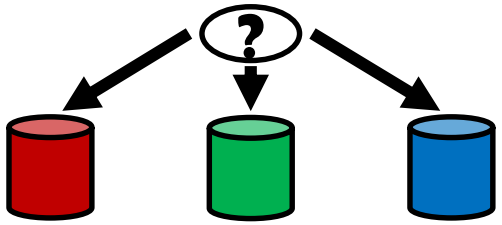
- Consider a situation, where you have to make a **choice**
- Each choice has its own **reward distribution** (which is unknown)
- Approach:
  - **Try out** each choice (= Explore)
  - Maintain **statistic** about each choice (= Learn)



Multi-Armed Bandits

- **Multi-Armed Bandit**: situation, where you have to learn to make a good (long-term) choice

- **Explore** choices to gather information (= Exploration)
  - Example: random choice



- **Prefer** promising choices (= Exploitation)
  - Example: greedy choice (e.g., using `argmax`)

- A good Multi-Armed Bandit solution should **always** balance between Exploration and Exploitation

Multi-Armed Bandits Example

- ϵ-greedy ($\epsilon > 0$):

With probability $\begin{cases} \epsilon, \text{ select randomly} \\ 1 - \epsilon, \text{ select choice highest average reward} \end{cases}$

  - Many approaches use ϵ-greedy with annealing ϵ

- UCB1 (Upper Confidence Bound):
  - Select by maximizing:

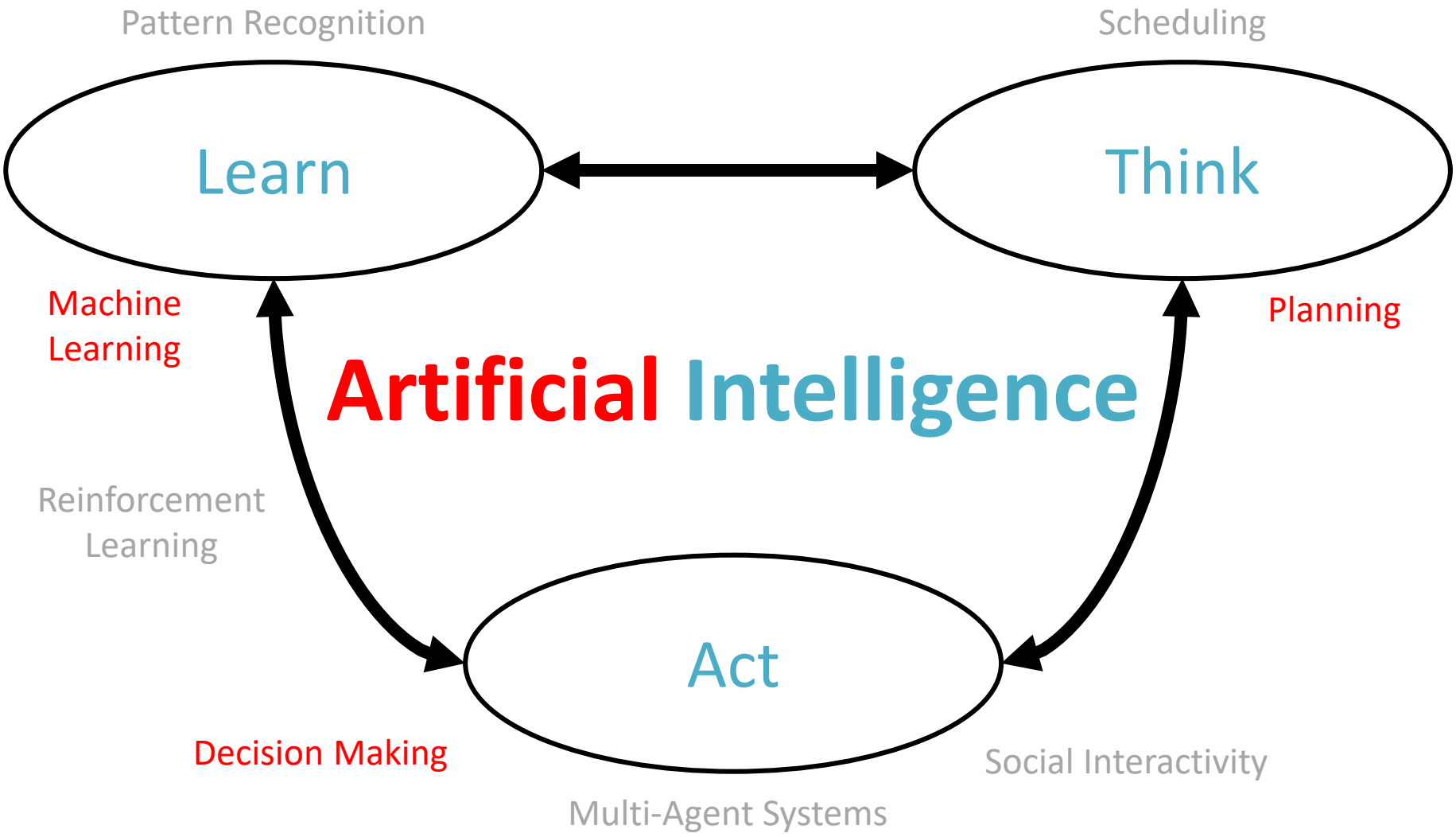
$$\overline{R_i} + c\sqrt{\frac{2logN}{N_i}}$$

Average reward of choice i

Exploration bonus (with total number of selections N)

Decision Making Challenges and Outlook

- Sequential Decision Making
- Problem Complexity
- Sparse/Delayed Feedback
- Sample Efficiency
- Uncertainty


Pattern Recognition
Scheduling
Learn
Think
Machine Learning
Planning
Artificial Intelligence
Reinforcement Learning
Act
Decision Making
Social Interactivity
Multi-Agent Systems

Thank you!