

Foundations of Artificial Intelligence

A5. Introduction: Environments and Problem Solving Methods

Malte Helmert

University of Basel

March 2, 2026

Foundations of Artificial Intelligence

March 2, 2026 — A5. Introduction: Environments and Problem Solving Methods

A5.1 Environments of Rational Agents

A5.2 Problem Solving Methods

A5.3 Classification of AI Topics

A5.4 Summary

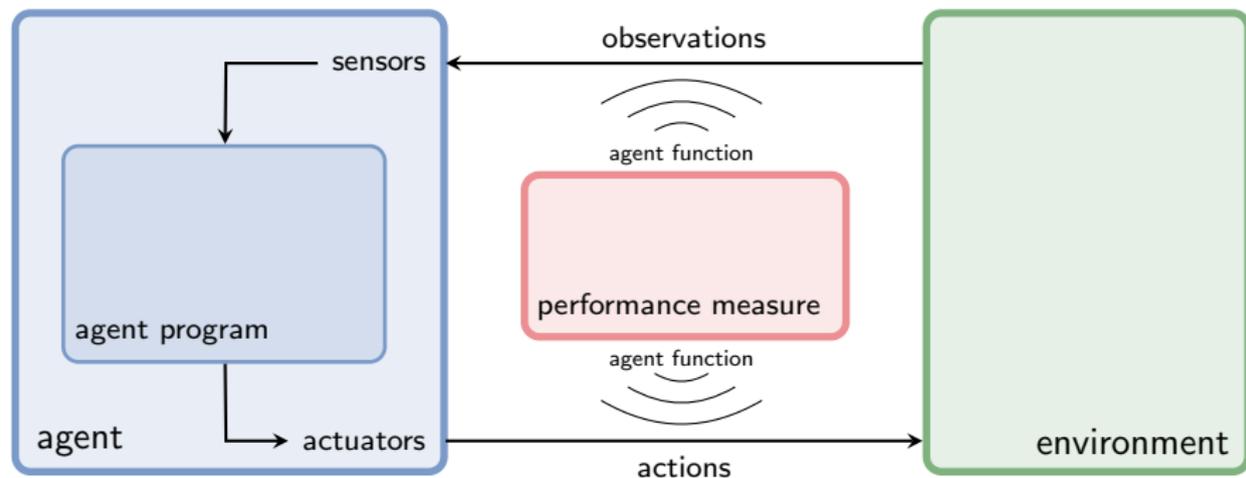
Introduction: Overview

Chapter overview: introduction

- ▶ A1. Organizational Matters
- ▶ A2. What is Artificial Intelligence?
- ▶ A3. AI Past and Present
- ▶ A4. Rational Agents
- ▶ A5. Environments and Problem Solving Methods

A5.1 Environments of Rational Agents

Environments of Rational Agents



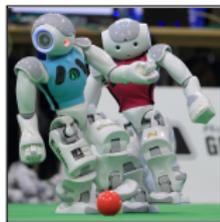
- ▶ Which environment aspects are **relevant for the agent**?
- ▶ How do the agent's actions **change the environment**?
- ▶ What does the agent **observe**?

Properties of Environments

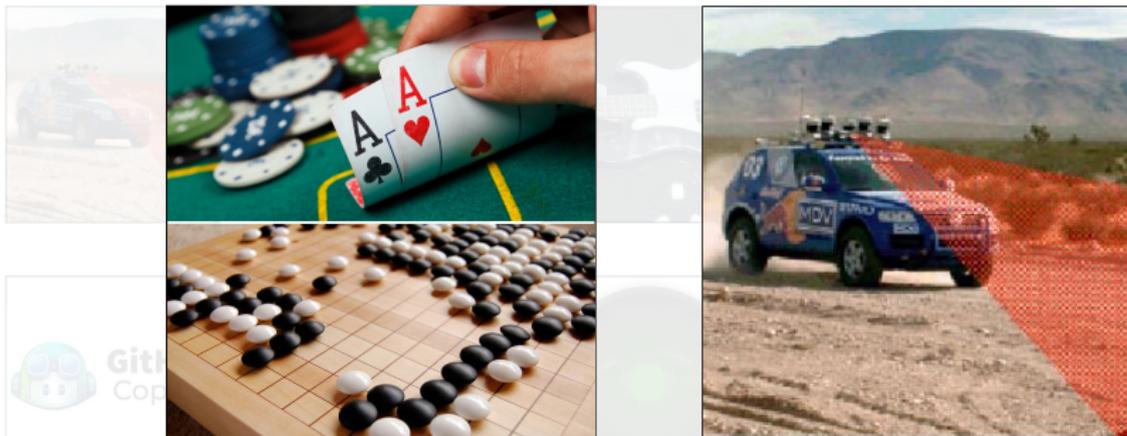
Environment properties determine **character** of AI problem.

- ▶ static vs. dynamic
- ▶ deterministic vs. nondeterministic vs. stochastic
- ▶ fully observable vs. partially observable
- ▶ discrete vs. continuous
- ▶ single-agent vs. multi-agent

Properties of Environments



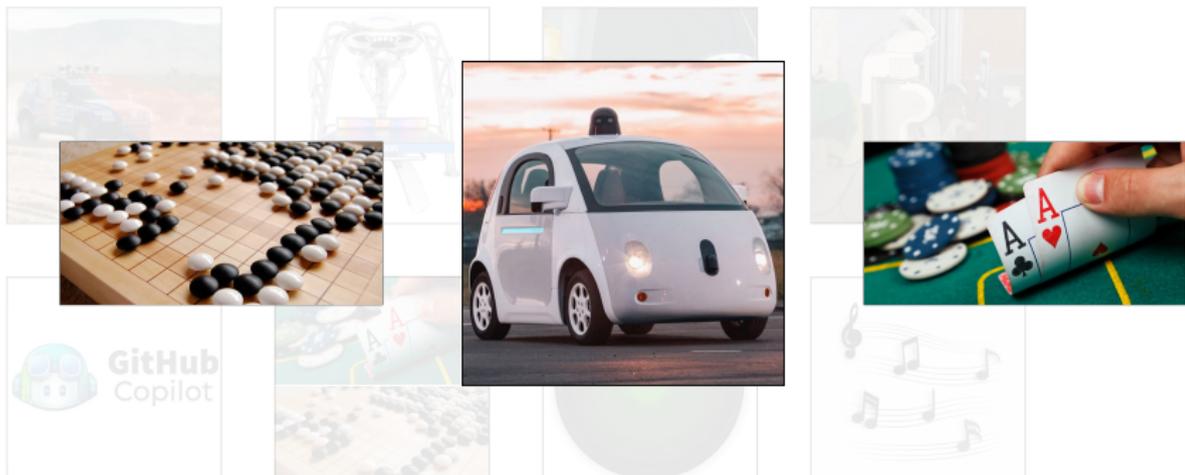
Properties of Environments



static vs. dynamic

Does the state of the environment remain the same while the agent is contemplating its next action?

Properties of Environments



deterministic vs. nondeterministic vs. stochastic

Is the next state of the environment fully determined by the current state and the next action? Are probabilities involved?

Properties of Environments



fully observable vs. partially observable

Can the agent fully observe the state of the environment at every decision step or not?

special case of partially observable: **unobservable**

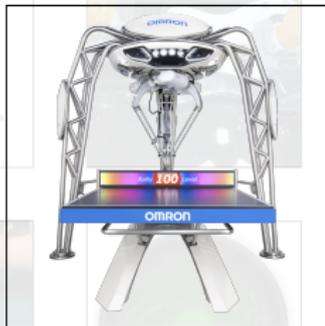
Properties of Environments



discrete vs. continuous

Is the state of the environment (and actions, observations, time) given by discrete or by continuous quantities?

Properties of Environments



single-agent vs. multi-agent

Are other agents relevant for own performance?

subcases of multi-agent: are the other agents

adversarial, **cooperative**, or **selfish**?

Properties of Environments



suitable problem-solving algorithms

Environments of different kinds (according to these criteria)
usually require different algorithms.



real world

The “real world” combines all unpleasant
(in the sense of: difficult to handle) properties.



A5.2 Problem Solving Methods

Three Approaches to Solving AI Problems

We can solve a **concrete AI problem** (e.g., playing chess) in several ways:

Problem Solving Methods

- 1 **problem-specific**: implement algorithm **tailored to problem**
- 2 **general**: create problem description as input for general **solver**
- 3 **learning**: **learn** (aspects of) algorithm from **data**

problem-specific algorithms:

- ▶ designed to solve a **specific problem**
- ▶ allow **exploiting problem-specific** knowledge
- ▶ solve **just one** (type of) **problem**

Three Approaches to Solving AI Problems

We can solve a **concrete AI problem** (e.g., playing chess) in several ways:

Problem Solving Methods

- 1 **problem-specific**: implement algorithm **tailored to problem**
- 2 **general**: create problem description as input for general **solver**
- 3 **learning**: **learn** (aspects of) algorithm from **data**

general problem solvers:

- ▶ user creates **model** of problem instance in **formalism** (“language”)
- ▶ **solver** takes modeled instance as **input**
- ▶ solver implements general **algorithm** to compute solution

Three Approaches to Solving AI Problems

We can solve a **concrete AI problem** (e.g., playing chess) in several ways:

Problem Solving Methods

- 1 **problem-specific**: implement algorithm **tailored to problem**
- 2 **general**: create problem description as input for general **solver**
- 3 **learning**: **learn** (aspects of) algorithm from **data**

learners:

- ▶ **general approach** that learns to solve **specific problem**
- ▶ adapts via **experience** instead of via **reasoning**
- ▶ requires **data** and **feedback** instead of **model** of the AI problems

Three Approaches to Solving AI Problems

We can solve a **concrete AI problem** (e.g., playing chess) in several ways:

Problem Solving Methods

- 1 **problem-specific**: implement algorithm **tailored to problem**
- 2 **general**: create problem description as input for general **solver**
- 3 **learning**: **learn** (aspects of) algorithm from **data**

- ▶ all three approaches have strengths and weaknesses
- ▶ combinations are possible (and common in practice)
- ▶ we will mostly focus on **general** algorithms, but also consider other approaches

A5.3 Classification of AI Topics

Classification of AI Topics

Many areas of AI are essentially characterized by

- ▶ the **properties of environments** they consider and
- ▶ which of the three **problem solving approaches** they use.

We conclude the introduction by giving some examples

- ▶ within this course and
- ▶ beyond the course (“advanced topics”).

Examples: Classification of AI Topics

Course Topic: Informed Search Algorithms

environment:

- ▶ static vs. dynamic
- ▶ deterministic vs. nondeterministic vs. stochastic
- ▶ fully observable vs. partially observable
- ▶ discrete vs. continuous
- ▶ single-agent vs. multi-agent

problem solving method:

- ▶ problem-specific vs. general vs. learning

Examples: Classification of AI Topics

Course Topic: Constraint Satisfaction Problems

environment:

- ▶ static vs. dynamic
- ▶ deterministic vs. nondeterministic vs. stochastic
- ▶ fully observable vs. partially observable
- ▶ discrete vs. continuous
- ▶ single-agent vs. multi-agent

problem solving method:

- ▶ problem-specific vs. general vs. learning

Examples: Classification of AI Topics

Course Topic: Board Games

environment:

- ▶ static vs. dynamic
- ▶ deterministic vs. nondeterministic vs. stochastic
- ▶ fully observable vs. partially observable
- ▶ discrete vs. continuous
- ▶ single-agent vs. multi-agent (adversarial)

problem solving method:

- ▶ problem-specific vs. general vs. learning

Examples: Classification of AI Topics

Advanced Topic: General Game Playing

environment:

- ▶ static vs. dynamic
- ▶ deterministic vs. nondeterministic vs. (stochastic)
- ▶ fully observable vs. partially observable
- ▶ discrete vs. continuous
- ▶ single-agent vs. multi-agent (adversarial)

problem solving method:

- ▶ problem-specific vs. general vs. learning

Examples: Classification of AI Topics

Course Topic: Classical Planning

environment:

- ▶ static vs. dynamic
- ▶ deterministic vs. nondeterministic vs. stochastic
- ▶ fully observable vs. partially observable
- ▶ discrete vs. continuous
- ▶ single-agent vs. multi-agent

problem solving method:

- ▶ problem-specific vs. general vs. learning

Examples: Classification of AI Topics

Course Topic: Acting under Uncertainty

environment:

- ▶ static vs. dynamic
- ▶ deterministic vs. nondeterministic vs. stochastic
- ▶ fully observable vs. partially observable
- ▶ discrete vs. continuous
- ▶ single-agent vs. multi-agent

problem solving method:

- ▶ problem-specific vs. general vs. learning

Examples: Classification of AI Topics

Advanced Topic: Reinforcement Learning

environment:

- ▶ static vs. dynamic
- ▶ deterministic vs. nondeterministic vs. stochastic
- ▶ fully observable vs. partially observable
- ▶ discrete vs. continuous
- ▶ single-agent vs. multi-agent

problem solving method:

- ▶ problem-specific vs. general vs. learning

A5.4 Summary

Summary (1)

AI problem: performance measure + agent model + environment

Properties of **environment** critical for choice of suitable algorithm:

- ▶ **static** vs. **dynamic**
- ▶ **deterministic** vs. **nondeterministic** vs. **stochastic**
- ▶ **fully observable** vs. **partially observable**
- ▶ **discrete** vs. **continuous**
- ▶ **single-agent** vs. **multi-agent**

Summary (2)

Three **problem solving methods**:

- ▶ **problem-specific**
- ▶ **general**
- ▶ **learning**

general problem solvers:

- ▶ **models** characterize problem instances mathematically
- ▶ **formalisms/languages** describe models compactly
- ▶ algorithms use languages as **problem description** and to **exploit problem structure**