## Foundations of Artificial Intelligence B2. State-Space Search: Representation of State Spaces

#### Malte Helmert

University of Basel

March 6, 2024

M. Helmert (University of Basel)

Foundations of Artificial Intelligence

March 6, 2024 1 / 16

Foundations of Artificial Intelligence March 6, 2024 — B2. State-Space Search: Representation of State Spaces

B2.1 Representation of State Spaces

B2.2 Explicit Graphs

B2.3 Declarative Representations

B2.4 Black Box

B2.5 Summary

## State-Space Search: Overview

#### Chapter overview: state-space search

- ▶ B1–B3. Foundations
  - B1. State Spaces
  - B2. Representation of State Spaces
  - B3. Examples of State Spaces
- ▶ B4–B8. Basic Algorithms
- ▶ B9–B15. Heuristic Algorithms

# B2.1 Representation of State Spaces

M. Helmert (University of Basel)

Foundations of Artificial Intelligence

March 6, 2024 4 / 16

### Representation of State Spaces

- practically interesting state spaces are often huge (10<sup>10</sup>, 10<sup>20</sup>, 10<sup>100</sup> states)
- How do we represent them, so that we can efficiently deal with them algorithmically?

#### three main options:

- as explicit (directed) graphs
- With declarative representations
- as a black box

German: explizit, deklarativ, Black Box

# B2.2 Explicit Graphs

## State Spaces as Explicit Graphs

#### State Spaces as Explicit Graphs

represent state spaces as explicit directed graphs:

- vertices = states
- directed arcs = transitions

 $\rightsquigarrow$  represented as adjacency list or adjacency matrix

German: Adjazenzliste, Adjazenzmatrix

Example (explicit graph for bounded inc-and-square) ai-b02-bounded-inc-and-square.graph

M. Helmert (University of Basel)

## State Spaces as Explicit Graphs

#### State Spaces as Explicit Graphs

represent state spaces as explicit directed graphs:

- vertices = states
- directed arcs = transitions

 $\rightsquigarrow$  represented as adjacency list or adjacency matrix

German: Adjazenzliste, Adjazenzmatrix

Example (explicit graph for 8-puzzle) ai-b02-puzzle8.graph

M. Helmert (University of Basel)

### State Spaces as Explicit Graphs: Discussion

#### discussion:

- impossible for large state spaces (too much space required)
- if spaces small enough for explicit representations, solutions easy to compute: Dijkstra's algorithm O(|S| log |S| + |T|)
- interesting for time-critical all-pairs-shortest-path queries (examples: route planning, path planning in video games)

# **B2.3 Declarative Representations**

# State Spaces with Declarative Representations

# State Spaces with Declarative Representations represent state spaces declaratively:

- compact description of state space as input to algorithms ~> state spaces exponentially larger than the input
- algorithms directly operate on compact description
- allows automatic reasoning about problem: reformulation, simplification, abstraction, etc.

# Example (declarative representation for 8-puzzle)

 $\verb"puzzle8-domain.pddl + \verb"puzzle8-problem.pddl"$ 

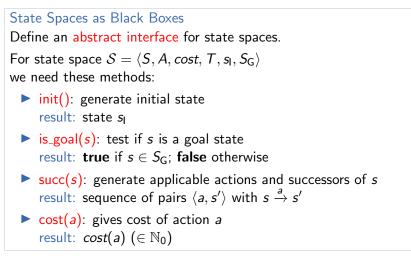
# B2.4 Black Box

M. Helmert (University of Basel)

Foundations of Artificial Intelligence

March 6, 2024 12 / 16

# State Spaces as Black Boxes



# Remark: we will extend the interface later in a small but important way

M. Helmert (University of Basel)

## State Spaces as Black Boxes: Example and Discussion

# Example (Black Box Representation for 8-Puzzle) demo: puzzle8.py

- in the following: focus on black box model
- explicit graphs only as illustrating examples
- near end of semester: declarative state spaces (classical planning)

# B2.5 Summary

### Summary

- explicit graphs: adjacency lists or matrices; only suitable for small problems
- declaratively: compact description as input to search algorithms
- black box: implement an abstract interface