Planning and Optimization

X1. Hands-On and Repetition

Thomas Keller and Martin Wehrle

Universität Basel

September 26, 2016

Introduction

Hands-On: Outline for this week

Working with an existing planning system (Fast Downward).

- Domain modeling
- Recognizing the difference: blind vs. informed planning
- Implementation in Fast Downward

Hands-On: Overview

Chapter overview: hands-on

- 1. The Planning Domain Definition Language (PDDL)
- 2. Getting to Know a Planner
- 3. Heuristics
- 4. A* search algorithm

X1.1 PDDL

Representation of State Spaces

Representation of State Spaces

- explicit graphs
- black box
- declarative representations

In this Course: Declarative Representations

- compact description of state space as input to algorithms
  $\rightsquigarrow$ state space exponentially larger than input
- algorithms operate directly on compact description

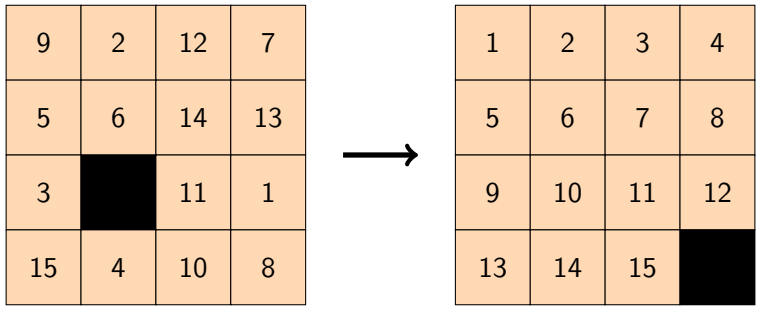
$\rightsquigarrow$ allows automatic reasoning about problem (abstractions etc.)

Representation of State Spaces

PDDL: Planning Domain Definition Language

- PDDL is the standard language used in practice to describe planning tasks.
- descriptions in (restricted) predicate logic instead of propositional logic ($\rightsquigarrow$ even more compact)
- There exist defined PDDL fragments for STRIPS and ADL; many planners only support the STRIPS fragment.
- In this week: restriction to STRIPS

Illustrating Example: 15-Puzzle



15-Puzzle in PDDL

Example: 15-Puzzle in PDDL

X1.2 Getting to Know a Planner

Planner: Fast Downward

Fast Downward

We use the Fast Downward planner in this course

- because we know it well (developed by our research group)
- because it implements many search algorithms and heuristics
- because it is the classical planner most commonly used as a basis for other planners these days

$\rightsquigarrow$ `http://www.fast-downward.org`

Solving the 15-Puzzle

Demo

```
$ ./fast-downward \
      tile/puzzle.pddl tile/puzzle01.pddl \
      --heuristic "h=ff()" \
      --search "eager_greedy(h)"
```

X1.3 Hands-On