

# Planning and Optimization

## A Brief Introduction to PDDL

Clemens Büchner, Thomas Keller and Silvan Sievers

Universität Basel

# PDDL

- short for Planning Domain Definition Language
- widely used in the planning community
- modelling language to describe planning domains
- separates domain description from instance description

# History of PDDL

More and more expressive versions have been published:

- 1998: PDDL 1.2 (basic version)
- 2002: PDDL 2.1 (numeric and temporal features)
- 2004: PDDL 2.2 (derived predicates and timed initial literals)
- 2004: PPDDL (probabilistic)
- 2006: PDDL 3 (soft goals and trajectory constraints)
- 2006: PDDL+ (continuous state spaces)

# History of PDDL

More and more expressive versions have been published:

- 1998: PDDL 1.2 (basic version)
- 2002: PDDL 2.1 (numeric and temporal features)
- 2004: PDDL 2.2 (derived predicates and timed initial literals)
- 2004: PPDDL (probabilistic)
- 2006: PDDL 3 (soft goals and trajectory constraints)
- 2006: PDDL+ (continuous state spaces)

We only consider a [subset](#) of [PDDL 1.2](#)!

## Components of a PDDL planning task

- **Objects** that exist in the task
- **Predicates** that describe properties of and relations between objects
- **Action schemas** that describe how the current state of objects can be changed
- An **initial state** and a **goal** that describe initial and desired properties of objects

# PDDL Skeleton

## Domain File

```
(define (domain <domain name>)  
  (:requirements :strips)  
  (:predicates  
    <list of predicate schemata>  
  )  
  <list of action schemata>  
)
```

# PDDL Skeleton

## Action Schema Specification

```
(:action <action name>  
  :parameters (<list of parameters>  
  :precondition (<precondition description>  
  :effect (<effect description>  
)
```

# PDDL Skeleton

## Instance File

```
(define (problem <problem name>)
  (:domain <domain name>)
  (:objects
    <list of objects>
  )
  (:init
    <predicates that hold in initial state>
  )
  (:goal
    <goal description>
  )
)
```



## PDDL Skeleton with action costs

### Domain File

```
(define (domain <domain name>)
  (:requirements :strips :action-costs)
  (:predicates
    <list of predicate schemata>
  )
  (:functions (total-cost) - number)
  <list of action schemata>
)
```

## PDDL Skeleton with action costs

### Action Schema Specification

```
(:action <action name>
  :parameters (<list of parameters>)
  :precondition (<precondition description>)
  :effect (and <effect description>
             (increase (total-cost) <action cost>))
  )
)
```

## PDDL Skeleton with action costs

### Instance File

```
(define (problem <problem name>)
  (:domain <domain name>)
  (:objects
    <list of objects>
  )
  (:init
    <predicates that hold in initial state>
    (= (total-cost) 0)
  )
  (:goal
    <goal description>
  )
  (:metric minimize (total-cost))
)
```