

Introduction, problem representations

STRIPS, FDR, heuristic function

Michaela Urbanovská

PUI Tutorial
Week 1

- Classical planning (M. Urbanovska), probabilistic planning (J. Mrkos)
- Each part one assignment - 50 points total
- Classical planning assignment → programming your own **planning system**
 - C, C++, Java
 - more details later
 - Deadline: half of April (TBD)
- **Zapocet:** 25 out of 50 points to pass
- **Exam:** 25 out of 50 points to pass
- **Final grade:** zapocet points + exam points
- Problems? Questions?
 - **urbanm30@fel.cvut.cz**
 - Forum

- STRIPS, FDR problem definitions
- PDDL, compilations
- Relaxation heuristics
- Landmark heuristics
- Abstraction heuristics
- LP based heuristics
- Neural planning (planning + ML)
- ...and much more!

Lecture check

- Any questions regarding the lecture?

- Any questions regarding the lecture?

when your lecturer asks if you have any questions



Planning motivation

- **General problem solving**
- Basically can solve all your problems

Planning motivation

- **General problem solving**
- Basically can solve all your problems
- Problem

Planning motivation

- **General problem solving**
- Basically can solve all your problems
- Problem + representation

Planning motivation

- **General problem solving**
- Basically can solve all your problems
- Problem + representation + solver

Planning motivation

- **General problem solving**
- Basically can solve all your problems
- Problem + representation + solver = **solution**

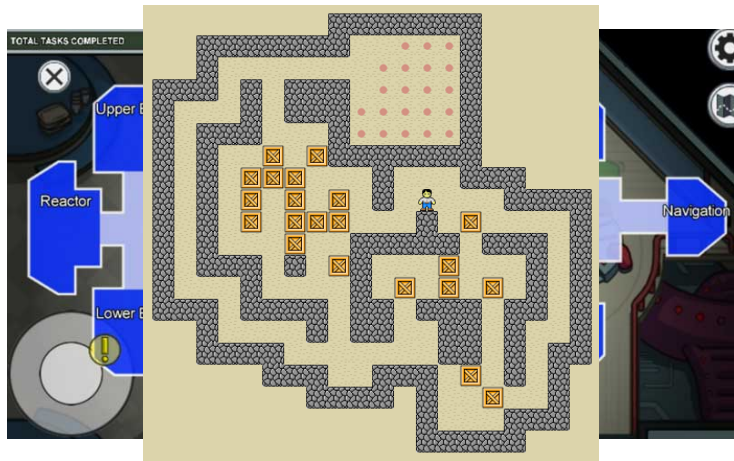
Planning motivation

- **General problem solving**
- Basically can solve all your problems
- Problem + representation + solver = **solution**



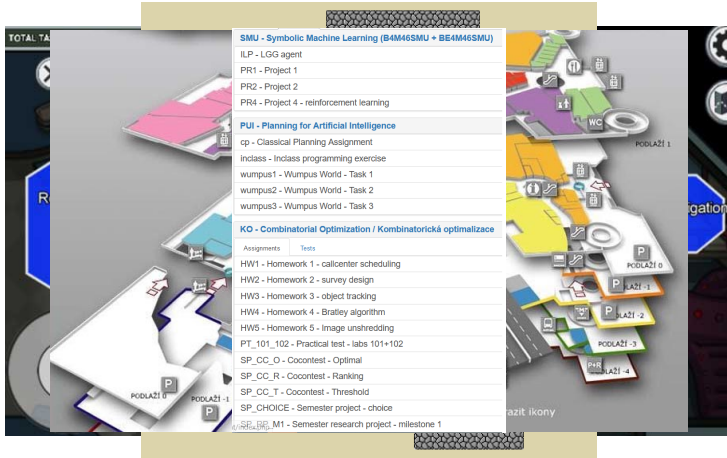
Planning motivation

- **General problem solving**
- Basically can solve all your problems
- Problem + representation + solver = **solution**



Planning motivation

- **General problem solving**
- Basically can solve all your problems
- Problem + representation + solver = **solution**



The screenshot displays a course website with a central list of assignments and tests. The background features a 3D architectural rendering of a building with various rooms and corridors, overlaid with a semi-transparent white box containing the course content.

SMU - Symbolic Machine Learning (B4M6SMU + BE4M6SMU)

- ILP - LGG agent
- PR1 - Project 1
- PR2 - Project 2
- PR4 - Project 4 - reinforcement learning

PUI - Planning for Artificial Intelligence

- cp - Classical Planning Assignment
- inclass - Inclass programming exercise
- wumpus1 - Wumpus World - Task 1
- wumpus2 - Wumpus World - Task 2
- wumpus3 - Wumpus World - Task 3

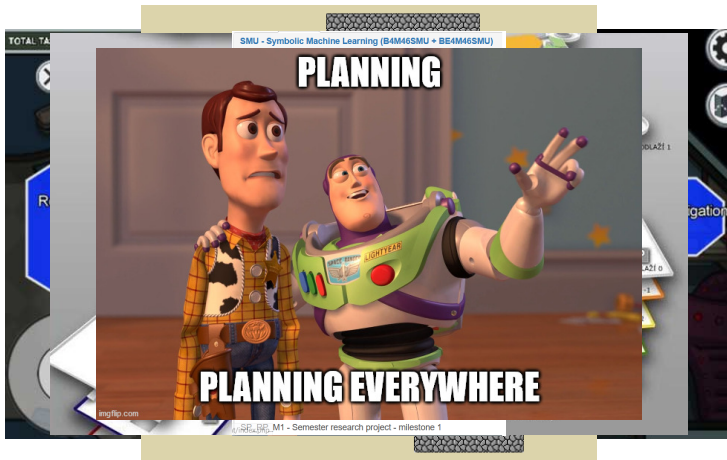
KO - Combinatorial Optimization / Kombinatorická optimalizace

Assignments Tests

- HW1 - Homework 1 - calicenter scheduling
- HW2 - Homework 2 - survey design
- HW3 - Homework 3 - object tracking
- HW4 - Homework 4 - Bratley algorithm
- HW5 - Homework 5 - Image unshredding
- PT_101_102 - Practical test - labs 101+102
- SP_CC_O - Cocontest - Optimal
- SP_CC_R - Cocontest - Ranking
- SP_CC_T - Cocontest - Threshold
- SP_CHOICE - Semester project - choice
- SP_RE_M1 - Semester research project - milestone 1

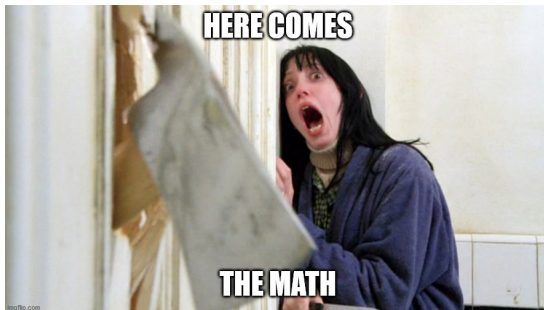
Planning motivation

- **General problem solving**
- Basically can solve all your problems
- Problem + representation + solver = **solution**



- **STRIPS**
- **FDR**
- **Specify** the model, capture the **structure**

- STRIPS
- FDR
- **Specify** the model, capture the **structure**



Problem in STRIPS

STRIPS problem $\Pi = \langle F, O, s_{init}, s_{goal}, c \rangle$

- $F = \{f_1, f_2, \dots, f_n\}$ (facts)
- $O = \{o_1, o_2, \dots, o_m\}$ (operators)
- $s_{init} \subseteq F$ (initial state)
- $s_{goal} \subseteq F$ (goal state specification)
- $c(o_i) \in \mathbb{R}^+$ (cost function)

Problem in STRIPS

STRIPS problem $\Pi = \langle F, O, s_{init}, s_{goal}, c \rangle$

- $F = \{f_1, f_2, \dots, f_n\}$ (facts)
- $O = \{o_1, o_2, \dots, o_m\}$ (operators)
- $s_{init} \subseteq F$ (initial state)
- $s_{goal} \subseteq F$ (goal state specification)
- $c(o_i) \in \mathbb{R}^+$ (cost function)

STRIPS operator $o = \langle pre(o), add(o), del(o) \rangle$

- $pre(o) \subseteq F$ (set of preconditions)
- $add(o) \subseteq F$ (set of add effects)
- $del(o) \subseteq F$ (set of delete effects)

Problem in STRIPS

STRIPS problem $\Pi = \langle F, O, s_{init}, s_{goal}, c \rangle$

- $F = \{f_1, f_2, \dots, f_n\}$ (facts)
- $O = \{o_1, o_2, \dots, o_m\}$ (operators)
- $s_{init} \subseteq F$ (initial state)
- $s_{goal} \subseteq F$ (goal state specification)
- $c(o_i) \in \mathbb{R}^+$ (cost function)

STRIPS operator $o = \langle pre(o), add(o), del(o) \rangle$

- $pre(o) \subseteq F$ (set of preconditions)
- $add(o) \subseteq F$ (set of add effects)
- $del(o) \subseteq F$ (set of delete effects)
- operators are **well-formed**
 - $add(o) \cap del(o) = \emptyset$
 - $pre(o) \cap add(o) = \emptyset$

Problem in STRIPS

Applicable operator

Operator o is applicable in state s if $pre(o) \subseteq s$.

Resulting state $res(o, s) = (s \setminus del(o)) \cup add(o)$.

State s is a **goal state** iff $s \subseteq s_{goal}$.

Applicable operator

Operator o is applicable in state s if $pre(o) \subseteq s$.

Resulting state $res(o, s) = (s \setminus del(o)) \cup add(o)$.

State s is a **goal state** iff $s \subseteq s_{goal}$.

Sequence of applicable operators

Sequence of operators $\pi = \langle o_1, o_2, \dots, o_n \rangle$ is applicable in state s_0 if there are states s_1, s_2, \dots, s_n such that o_i is applicable in s_{i-1} and $s_i = res(o_i, s_{i-1})$ for $1 \leq i \leq n$.

- $res(\pi, s_0) = s_n$ (result of the applied operator sequence π)
- $c(\pi) = \sum_{o \in \pi} c(o)$ (cost of applying the operator sequence π)

Problem in STRIPS

Applicable operator

Operator o is applicable in state s if $pre(o) \subseteq s$.

Resulting state $res(o, s) = (s \setminus del(o)) \cup add(o)$.

State s is a **goal state** iff $s \subseteq s_{goal}$.

Sequence of applicable operators

Sequence of operators $\pi = \langle o_1, o_2, \dots, o_n \rangle$ is applicable in state s_0 if there are states s_1, s_2, \dots, s_n such that o_i is applicable in s_{i-1} and $s_i = res(o_i, s_{i-1})$ for $1 \leq i \leq n$.

- $res(\pi, s_0) = s_n$ (result of the applied operator sequence π)
- $c(\pi) = \sum_{o \in \pi} c(o)$ (cost of applying the operator sequence π)

Sequence π is called a **plan** if $s_{goal} \subseteq res(\pi, s_{init})$.

- π is an **optimal plan** if $c(\pi)$ is the minimal cost over all plans

Reachable state

State s is **reachable** if there exists an applicable sequence of operators π such that $res(\pi, s_{init} = s)$.

Set of all reachable states is denoted \mathcal{R}_Π .

BUT WAIT



THERE'S MORE

FDR problem $P = \langle \mathcal{V}, \mathcal{O}, s_{init}, s_{goal}, c \rangle$

- $\mathcal{V} = \{V_1, V_2, \dots, V_n\}$ (*finite set of variables*)
- $\mathcal{O} = \{o_1, o_2, \dots, o_m\}$ (*set of operators*)
- s_{init} (*initial state*)
- s_{goal} (*goal state*)
- $c(o_i) \in \mathbb{R}^+$

FDR problem $P = \langle \mathcal{V}, \mathcal{O}, s_{init}, s_{goal}, c \rangle$

- \mathcal{V} (*finite set of variables*)
 - $V \in \mathcal{V}$ (*variable*)
 - D_V (*finite domain of variable V*)
- s (*state*) is partial variable assignment over \mathcal{V}
 - $\text{vars}(s) = V \in \mathcal{V}$ assigned in s
 - $s[V]$ = value of V in s
 - s is **consistent** with s' if $s[V] = s'[V]$ for all $V \in \text{vars}(s')$
 - atom $V = v$ is true in s if $s[V] = v$

FDR operator $o = \langle pre(o), eff(o) \rangle$

- \mathcal{O} (set of operators)
 - $pre(o)$ = partial assignment over \mathcal{V} (*preconditions*)
 - $eff(o)$ = partial assignment over \mathcal{V} (*effects*)
 - $V = v$ cannot be in both $pre(o)$ and $eff(o)$

Applicable operator

Operator o is applicable in state s if $pre(o)$ is **consistent** with s .

Resulting state $res(o, s) = \begin{cases} eff(o)[V], & \text{if } V \in vars(eff(o)) \\ s[V], & \text{otherwise} \end{cases}$

Applicable operator

Operator o is applicable in state s if $pre(o)$ is **consistent** with s .

Resulting state $res(o, s) = \begin{cases} eff(o)[V], & \text{if } V \in vars(eff(o)) \\ s[V], & \text{otherwise} \end{cases}$

Sequence of applicable operators

Sequence of operators $\pi = \langle o_1, o_2, \dots, o_n \rangle$ is applicable in state s_0 if there are state s_1, s_2, \dots, s_n such that o_i is applicable in s_{i-1} and $s_i = res(o_i, s_{i-1})$ for $1 \leq i \leq n$.

- $res(\pi, s_0) = s_n$ (*result of the applied operator sequence π*)
- $c(\pi) = \sum_{o \in \pi} c(o)$ (*cost of applying the operator sequence π*)

Applicable operator

Operator o is applicable in state s if $pre(o)$ is **consistent** with s .

Resulting state $res(o, s) = \begin{cases} eff(o)[V], & \text{if } V \in vars(eff(o)) \\ s[V], & \text{otherwise} \end{cases}$

Sequence of applicable operators

Sequence of operators $\pi = \langle o_1, o_2, \dots, o_n \rangle$ is applicable in state s_0 if there are state s_1, s_2, \dots, s_n such that o_i is applicable in s_{i-1} and $s_i = res(o_i, s_{i-1})$ for $1 \leq i \leq n$.

- $res(\pi, s_0) = s_n$ (*result of the applied operator sequence π*)
- $c(\pi) = \sum_{o \in \pi} c(o)$ (*cost of applying the operator sequence π*)

Sequence π is called a **plan** if $res(\pi, s_{init})$ is consistent with s_{goal} .

- π is an **optimal plan** if $c(\pi)$ is the minimal cost over all plans

Heuristic functions

- helps to navigate the search → informed search
- returns plan cost estimate for a state

- helps to navigate the search \rightarrow informed search
- returns plan cost estimate for a state

s-plan

Let Π denote a STRIPS planning task. Sequence of operators π is an **s-plan** iff π is applicable in s and $res(\pi, s)$ is a goal state.

Heuristic $h : \mathcal{R}_{\Pi} \mapsto \mathbb{R} \cup \{\infty\}$ estimates costs of optimal s-plans.

- h^* **optimal** heuristic
- h is **admissible** iff $h(s) \leq h^*(s)$ for every $s \in \mathcal{R}_\Pi$
- h is **goal-aware** iff $h(s) \leq 0$ for every reachable goal state s
- h is **safe** iff $h(s) = \infty$ implies $h^* = \infty$ (*there's not plan*)
- h is **consistent** iff $h(s) \leq h(\text{res}(o, s)) + c(o)$ for all reachable states $s \in \mathcal{R}_\Pi$ and $o \in \mathcal{O}$ applicable in s

- think that planning isn't that lame

- think that planning isn't that lame ✓

- think that planning isn't that lame ✓
- STRIPS and FDR problem definition
- what is a plan
- what is a heuristic
- heuristic function properties

For more details and structure check out

- [Notes on Classical Planning by Daniel Fiser](#)



Feedback form

