

Úvod do umělé inteligence (NAIL120)

5. cvičení

Jirka Fink

<https://ktiml.mff.cuni.cz/~fink/>

Katedra teoretické informatiky a matematické logiky
Matematicko-fyzikální fakulta
Univerzita Karlova v Praze

Letní semestr 2023/24

Poslední změna 18. března 2024

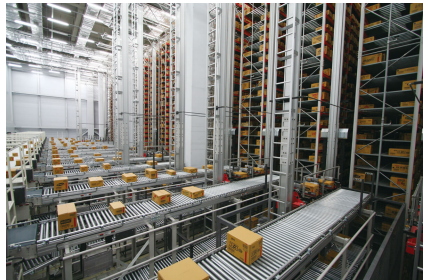
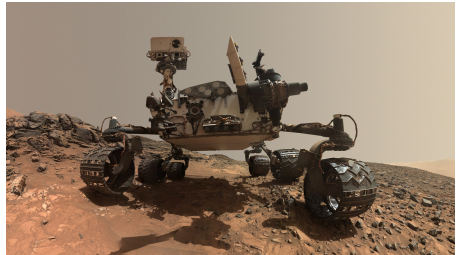
Licence: Creative Commons BY-NC-SA 4.0

Zadání (zkráceno)

Pomocí CSP a SAT najděte úplné obarvení grafu (total chromatic index)

- Máme obarvit vrcholy i hrany grafu pomocí minimálního počtu barev
- Každé dva sousední vrcholy musí mít různou barvu
- Každé dvě hrany sdílející společný vrchol musí mít různou barvu
- Incidentní vrchol a hrana musí mít různou barvu
- Šablona na gitu

- Lineární programování
- Konvexní optimalizace
- Constraint satisfaction programming (splňování podmínek)
- SAT (splnitelnost logických formulí)
- **Automatické plánování**



Základní idea

Najít posloupnost akcí, která postupně mění stav z počátečního do cílového.

- Stav je popsán konečnou množinou proměnných
 - Základní řešiče mají jen logické proměnné, pokročilé řešiče mají i číselné proměnné
- Počáteční stav: Ohodnocení všech proměnných na začátku plánu
- Cílový stav: Logické nebo aritmetické podmínky na hledaný stav
- Ohodnocení proměnných se mění pomocí akcí
 - Každá akce má předpoklady, které musí být splněny, aby bylo možné akci provést
 - Každá akce má efekty měnící hodnoty proměnných
 - Parametry, aby akce mohli být obecné

Planning Domain Definition Language (PDDL)

- PDDL je jazyk, který se snaží standardizovat plánování v UI
- Popis modelu je rozdělen do dvou souborů
 - Problém: Seznam objektů, počáteční a cílový stav
 - Doména: Seznam proměnných a akcí

Problem

```
1 (define (problem balls)
2   (:domain balls)
3   (:objects room1 room2 ball1)
4   (:init (room room1)
5         (room room2)
6         (ball ball1)
7         (at ball1 room1))
8   (:goal (at ball1 room2)))
```

Domain

```
1 (define (domain balls)
2   (:predicates (room ?r) (ball ?b) (at ?b ?r))
3   (:action move
4     :parameters (?b ?from ?to)
5     :precondition (and (ball ?b)
6                       (room ?from)
7                       (room ?to)
8                       (at ?b ?from))
9     :effect (and (at ?b ?to)
10                (not (at ?b ?from))))
```

Nalezený plán

```
1 (move ball1 room1 room2)
```

Špatný popis úlohy můžeme vést k nereálnému výsledku

Problem

```
1 (define (problem balls)
2   (:domain balls)
3   (:objects room1 room2 room3 room4 ball1)
4   (:init (room room1) (room room2) (room room3) (room room4)
5         (ball ball1)
6         (at ball1 room1))
7   (:goal (and (at ball1 room2)
8             (at ball1 room3)
9             (at ball1 room4))))
```

Domain

```
1 (define (domain balls)
2   (:predicates (room ?r) (ball ?b) (at ?b ?r))
3   (:action move
4     :parameters (?b ?from ?to)
5     :precondition (at ?b ?from)
6     :effect (at ?b ?to)))
```

Nalezený plán

```
1 (move ball1 room1 room4)
2 (move ball1 room1 room3)
3 (move ball1 room1 room2)
```

Popis problému

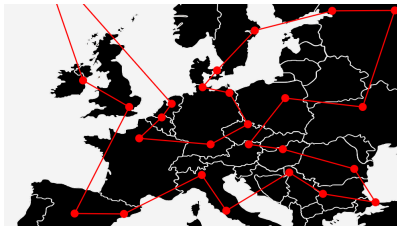
Máme robota, který má dvě chapadla, do kterých umí uchopit míč, položit jej na zem a přesouvat se mezi místnostmi.

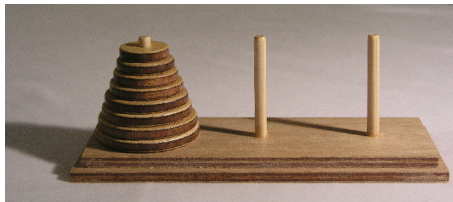
```
1 (define (problem strips-gripper2)
2   (:domain gripper-strips)
3   (:objects rooma roomb ball1 ball2 left right)
4   (:init (room rooma)
5         (room roomb)
6         (ball ball1)
7         (ball ball2)
8         (gripper left)
9         (gripper right)
10        (at-robby rooma)
11        (free left)
12        (free right)
13        (at ball1 rooma)
14        (at ball2 rooma))
15  (:goal (at ball1 roomb)))
```



```
1 (define (domain gripper-strips)
2   (:predicates (room ?r) (ball ?b) (gripper ?g) (at-robby ?r)
3               (at ?b ?r) (free ?g) (carry ?o ?g))
4   (:action move
5     :parameters (?from ?to)
6     :precondition (and (room ?from)
7                       (room ?to)
8                       (at-robby ?from))
9     :effect (and (at-robby ?to)
10                (not (at-robby ?from))))
```

```
1 (:action pick
2  :parameters (?obj ?room ?gripper)
3  :precondition (and (ball ?obj)
4                    (room ?room)
5                    (gripper ?gripper)
6                    (at ?obj ?room)
7                    (at-robby ?room)
8                    (free ?gripper))
9  :effect (and (carry ?obj ?gripper)
10             (not (at ?obj ?room))
11             (not (free ?gripper))))
12 (:action drop
13  :parameters (?obj ?room ?gripper)
14  :precondition (and (ball ?obj)
15                  (room ?room)
16                  (gripper ?gripper)
17                  (carry ?obj ?gripper)
18                  (at-robby ?room))
19  :effect (and (at ?obj ?room)
20             (free ?gripper)
21             (not (carry ?obj ?gripper))))
```





- Cílem je přemístit všechny kotouče na druhou věž.
- V jednom tahu lze přemístit vrchní kotouč z některé věže a položit jej na jinou věž mající větší kotouč nvrchu.
- Popište hledání řešení pomocí PDDL.
- Ilustrativní animace

15	2	1	12
8	5	6	11
4	9	10	7
3	14	13	

Zadání (zkráceno)

- Máme dány počáteční a cílové pozice krabic, které máme převést pomocí aut
- Každé auto kapacitu jedné krabice
- Napište PDDL doménu obsahující právě tyto akce
 - load(box car place): Naloží krabici do auta.
 - unload(box car place): Vyloží krabici z auta.
 - move(car origin destination): Přesune auto.
- Napište jednu doménu, která vyřeší všech 10 problémů.

Knihovna pro Python

- pyperplan: Jednoduchý řešič PDDL
<https://github.com/aibase1/pyperplan>