

ÚVOD DO UMĚLÉ INTELIGENCE (CVIČENÍ 5)

Simona Ondrčková

2. A 3. DOMÁCÍ ÚKOL (CSP A SAT)

Najděte úplné obarvení grafu dle SATu:

Obarvěte vrcholy i hrany pomocí minimálního počtu barev.

Dva sousední vrcholy musí mít jinou barvu.

Dvě hrany se společným vrcholem musí mít různé barvy.

Hrana a vrchol ke kterému patří musí mít různé barvy.

Knihovny pro python: `networkx`, `python-sat`

PLÁNOVÁNÍ

START

(KLASICKÉ) PLÁNOVÁNÍ

Problém plánování se popíše v PDDL.

Počáteční stav je seznam atomů jež jsou pravdivé.

Cílový stav je seznam atomů jež musíme splnit.

Akce mění stav světa. To je popsáno pomocí efektů. Akce také může mít předpoklady.

Objekty, jež se vyskytují ve světě.

Problém: popisuje objekty, počáteční a cílový stav

Doména: popisuje akce a proměnné.

POSOUVÁNÍ MÍČŮ

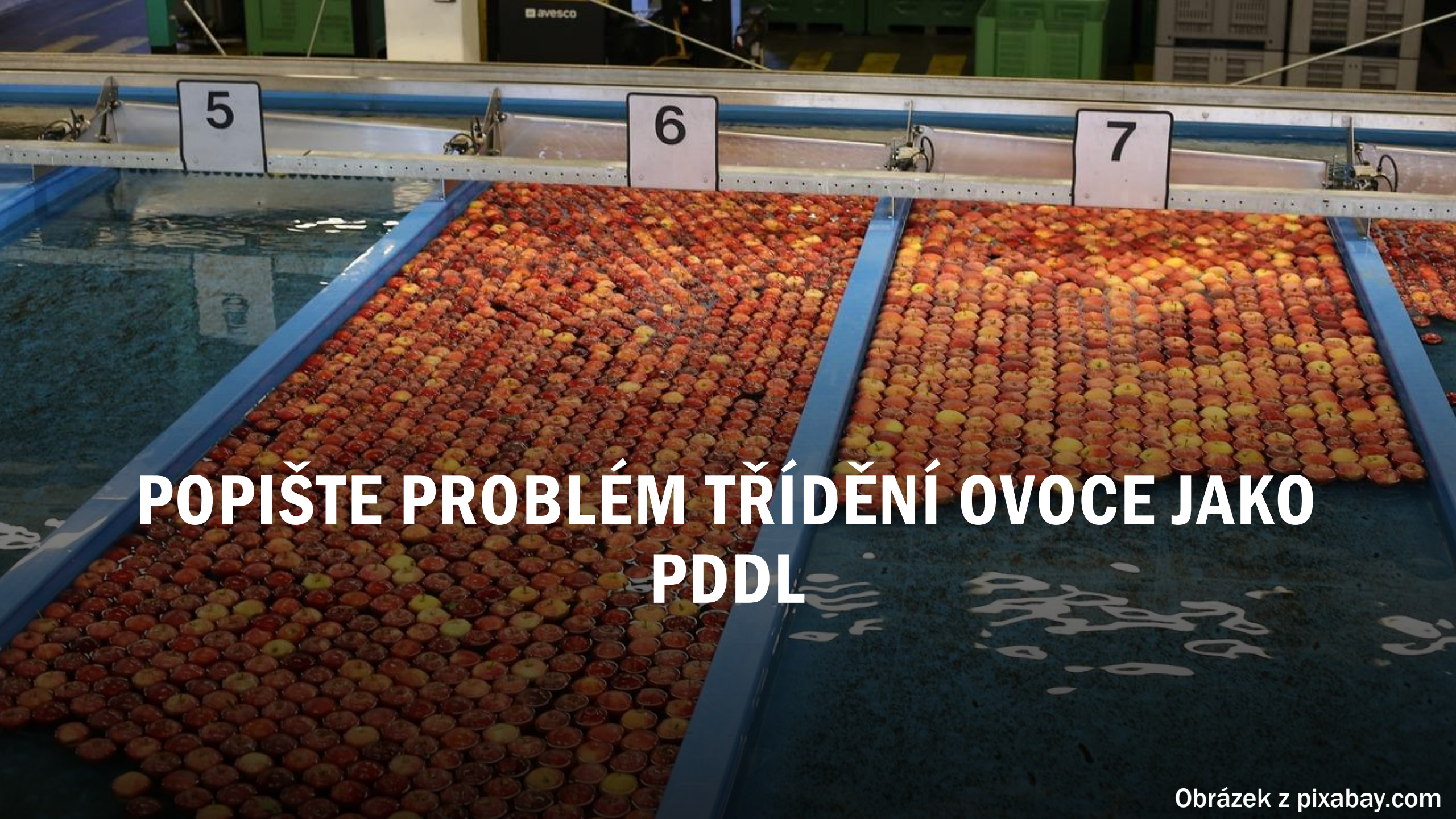
Problém:

```
(define (problem balls)
  (:domain balls)
  (:objects room1 room2 ball1)
  (:init (ROOM room1)
         (ROOM room2)
         (BALL ball1)
         (at ball1 room1))
  (:goal (at ball1 room2)))
```

Doména:

```
(define (domain balls)
  (:action move
    :parameters((?r1,?r2?b))
    :preconditions(ROOM(?r1),ROOM(?r2),BALL(?b) at(?r1 ?b)
    :effects (at ?r2 ?b), not (at(?r1 ?b))
```

Jak dosáhnete cíle?



POPIŠTE PROBLÉM TŘÍDĚNÍ OVOCE JAKO PDDL

TŘÍDĚNÍ OVOCE

Máte robota, jež má roztřídit ovoce dle typu.

Máte tři koše: vstupní koš, koš na jablka, koš na hrušky.

TŘÍDĚNÍ OVOCE

Goal: $\forall x \in \text{apple}: \text{at}(x, \text{appleBasket}), \forall y \notin \text{apple}: \text{at}(y, \text{pearBasket})$

Init: $\text{at}(x, \text{inputBasket}), \dots, \text{at}(x_n, \text{inputBasket}), \text{basket}(\text{inputBasket}), \text{basket}(\text{PearBasket}), \text{basket}(\text{AppleBasket})$

$\forall x: \text{isApple}(x) \vee \text{not}(\text{isApple}(x)), \text{fruit}(x)$

(:action sortApple

:paramaters (?f, inputBasket, appleBasket)

:preconditions(fruit(f), basket(inputBasket), basket(appleBasket), at(?f, inputBasket), isApple(?f))

:effects (at(?f, appleBasket), not(at(?f, inputBasket)))

(:action sortPear

:paramaters (?f, inputBasket, pearBasket)

:preconditions((fruit(f), basket(inputBasket), basket(pearBasket), at(?f, inputBasket), not(isApple(?f))))

:effects (at(?f, pearBasket), not(at(?f, inputBasket)))



POPIŠTE PROBLÉM HAMILTONOVSKÉ KRUŽNICE JAKO PDDL

HAMILTONOVSKÁ KRUŽNICE

Vymyslíte řešení jen s jednou akcí?

Goal: Pro každé město budeme mít požadavek: visited city

finalmove(s1) – s1 je počáteční město

Init: cesty mezi městy a at(s1): path(s1,s2)

Pro každé město s: CITY(s)

(:action move

:parameters(?c1,?c2)

:preconditions(CITY(?c1),CITY(?c2),
at(?c1), not(visited,c2)
path(c1,c2)

:effects(at(?c2)
not(at(?c1))
visited(c1))

(:action finalmove

:parameters(?c1,?c2)

:preconditions(CITY(?c1),CITY(?c2),
at(?c1),path(c1,c2)

:effects(not(at(?c1))
finalmove(c2))

A photograph of a wooden Tower of Hanoi puzzle. On the left, a stack of 11 wooden disks of varying sizes is balanced on a central wooden peg. To the right, two other wooden pegs stand empty. The entire puzzle is set on a dark wooden base against a plain, light-colored background. The lighting is soft, casting gentle shadows.

POPIŠTE HANOJSKÉ VĚŽE POMOCÍ PDDL

HANOJSKÉ VĚŽE

Cílem je dostat všechny disky na jednu věž od nejmenšího po největší.

Lze hýbat jen s jedním diskem najednou.

Nesmí se položit větší disk na menší.

Začne se se všemi disky na jedné věži a chci dostat všechny disky na jinou věž.

HANOJSKÉ VĚŽE

Goal: $\forall x$: at(disc_x,disc_x+1,t3),at(disc7,empty,t3),top(disc1,t3)

Init: $\forall x$: at(disc_x,disc_x+1,t1),

$\forall x, y \ x < y$: canPlace(disc_x,disc_y),

$\forall x$: canPlace(disc_x,empty), DISC(disc_x)

top(disc1,t1),top(empty,t2),top(empty,t3),at(disc7,empty,t1),TOWER(t1),TOWER(t2),TOWER(t3),DISC(empty)

(:action moveDisc

 :parameters(?d1, ?d2, ?t1,?d3, ?t2)

 :preconditions(TOWER(?t1),TOWER(?t2),DISC(?d1),DISC(?d2),DISC(?d3)

 at(?d1,?d2,?t1), top(?d1,?t1), top(?d3,?t2),canPlace(?d1,?d3))

 :effects(not(at(?d1,?d2,?t1), not(top(?d1,?t1)),not(top(?d3,?t2)),

 top(?d2,?t1),top(?d1,?t2), at(?d1,?d3,?t2))

4. DOMACÍ ÚKOL PDDL

(Přibližné zadání:) Jsou dané počáteční a cílové pozice krabic.

Krabice převežete pomocí aut.

Napište PDDL doménu s těmito akcemi:

Load(box,car,place): naloží krabici do auta.

Unload(box,car,place): vyloží krabici z auta

Move(place1,place2): Přejede autem z jednoho místa na druhé.

Do každého auta se vejde jen jedna krabice.

Napište jednu doménu, jež vyřeší všech 10 problémů.

Knihovny: pyperplan