Základy umělé inteligence Seminární cvičení 4

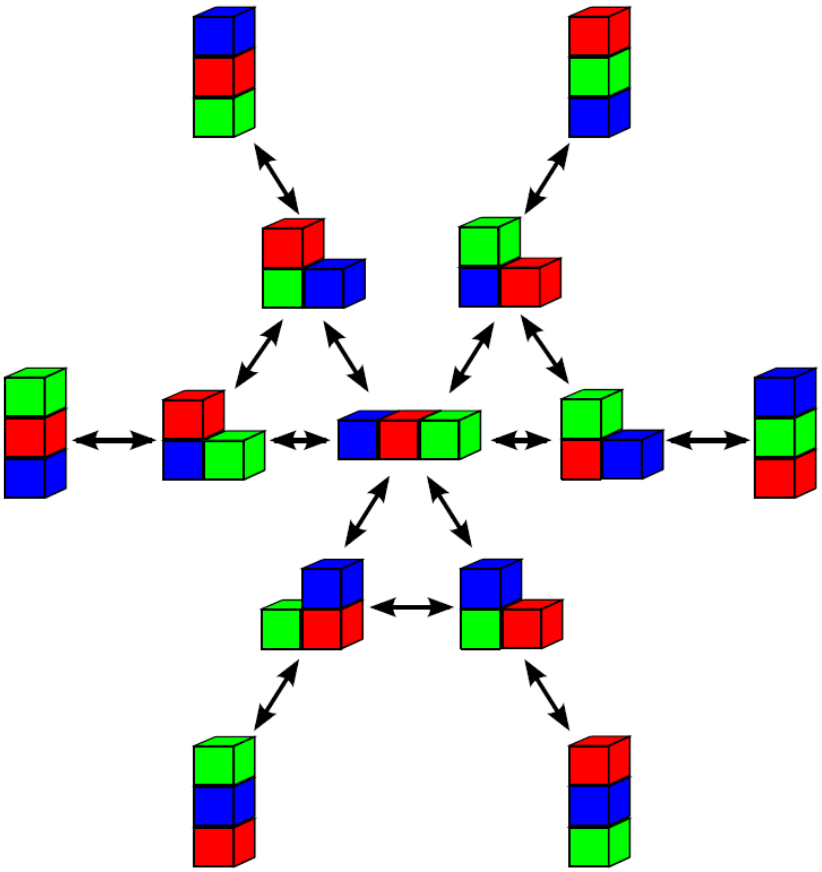
Marcel Jiřina

Plánování

- Vždy musíme nadefinovat 5 věcí:
  - Univerzum objektů, s nimiž budeme pracovat,
  - Predikáty,
  - Akce,
  - Počáteční stav,
  - Cílový stav.

- Každá akce a je dána parametry a trojicí pre(a), add(a) a del(a).
  - **pre(a)** je množina předpokladů – mohou to být pouze v pozitivní formě, žádné negace ani disjunkce atd.,
  - **add(a)** jsou add efekty, tedy co začne po provedení akce platit,
  - **del(a)** jsou delete efekty, tedy co platit přestane.

Kostičky



Kostičky – obecný popis problému

```
(define (domain blocks)
   (:requirements :strips)
   (:predicates
      (on ?x ?y)
      (distinct ?x ?y)
      (on-top ?x)
      (on-ground ?x)
   )
   (:action move
      :parameters (?what ?from ?to)
      :precondition (and
         (distinct ?what ?to)
         (on ?from ?what)
         (on-top ?what)
         (on-top ?to)
       )
      :effect (and
         (on ?to ?what)
         (on-top ?from)
         (not (on ?from ?what))
         (not (on-top ?to))
       )
   )
```

```
   (:action from-ground
      :parameters (?what ?to)
      :precondition (and
         (on-ground ?what)
         (on-top ?what)
         (on-top ?to)
       )
      :effect (and
         (on ?to ?what)
         (not (on-ground ?what))
         (not (on-top ?to))
       )
   )
   (:action to-ground
      :parameters (?what ?from)
      :precondition (and
         (on ?from ?what)
         (on-top ?what)
       )
      :effect (and
         (on-ground ?what)
         (on-top ?from)
         (not (on ?from ?what))
       )
   )
)
```

Kostičky – konkrétní konfigurace

```
(define (problem blocks-3)
    (:domain gripper)
    (:requirements :strips)
                        (:objects red green blue)
    (:init
        (on-ground green)
        (on green red)
        (on red blue)
        (on-top blue)
        (distinct red green)
        (distinct green red)
        (distinct red blue)
        (distinct blue red)
        (distinct green blue)
        (distinct blue green)
    )
    (:goal (and
        (on-ground red)
        (on red green)
        (on green blue)
        (on-top blue)
      )
    )
)
```

Transport balíků mezi ostrovy

- Máme 3 ostrovy
- Na každém ostrově jsou nějaké kontejnery. Třeba na o1 jsou kontejnery A,B, na o2 kontejnery C,D, na o3 E,F.
- U ostrova o1 kotví loďka, která má kapacitu jeden kontejner. Naložená nebo i prázdná může jezdit mezi ostrovy a převážet vždy max. 1 balík.
- Požadujeme nějakou výslednou konfiguraci, kdy na o1 je třeba D,F atd.
- Chceme minimální počet přejetí lodi.

Transport balíků mezi ostrovy

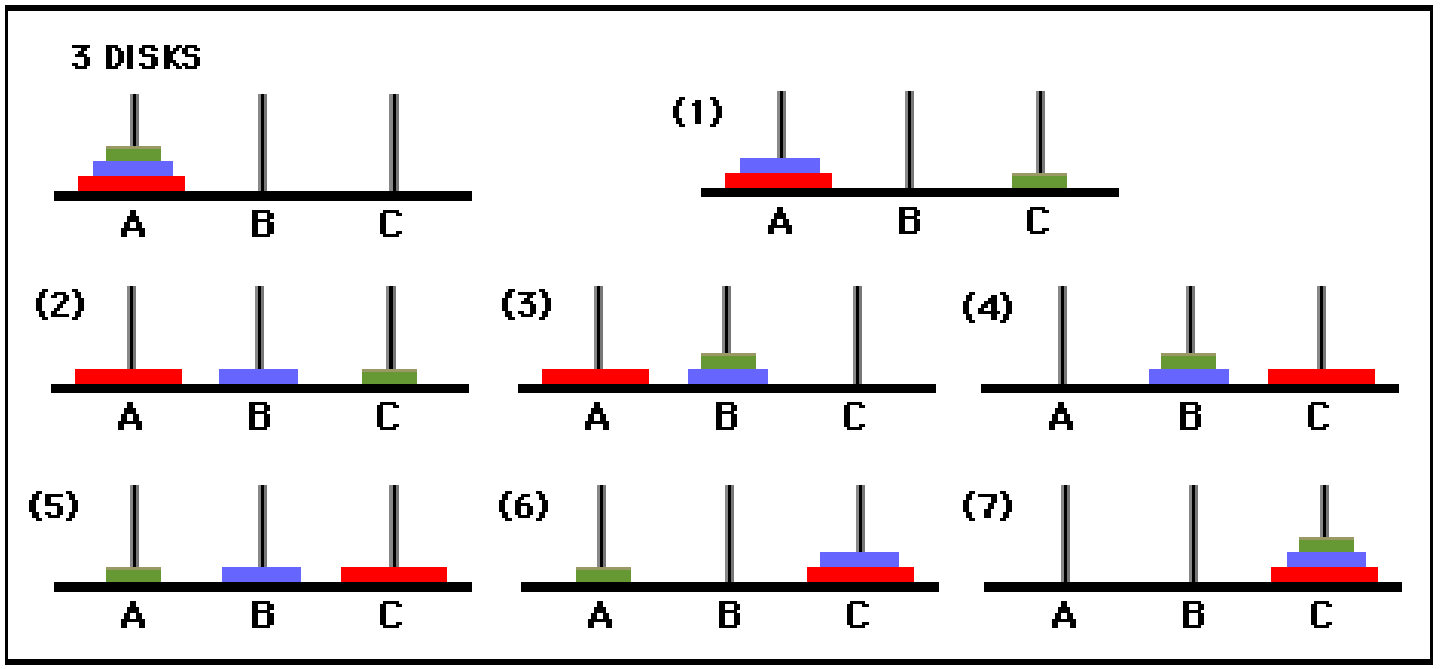
```
(define (domain ships)
  (:requirements :strips)
  (:predicates
      (at ?where ?what)
      (empty ?ship)
      (ship ?ship)
      (place ?place)
      (item ?item)
  )
  (:action load
   :parameters (?where ?ship ?what)
   :precondition (and
       (place ?where)
       (ship ?ship)
       (item ?what)
       (empty ?ship)
       (at ?where ?ship)
       (at ?where ?what)
     )
   :effect (and
       (not (empty ?ship))
       (at ?ship ?what)
       (not (at ?where ?what))
     )
  )
  (:action unload
   :parameters (?where ?ship ?what)
   :precondition (and
       (place ?where)
       (ship ?ship)
       (item ?what)
       (at ?ship ?what)
       (at ?where ?ship)
     )
   :effect (and
       (empty ?ship)
       (not (at ?ship ?what))
       (at ?where ?what)
     )
  )
  (:action move
   :parameters (?ship ?from ?to)
   :precondition (and
       (ship ?ship)
       (place ?from)
       (place ?to)
       (at ?from ?ship)
     )
   :effect (and
       (at ?to ?ship)
       (not (at ?from ?ship))
     )
  )
)
```

Transport balíků mezi ostrovy

```
(define (problem islands-3)
                    (:domain ships)
  (:requirements :strips)
  (:objects ship p1 p2 p3 A B C D E F G H I J)
  (:init
     (place p1)
     (place p2)
     (place p3)
     (item A)
     (item B)
     (item C)
     (item D)
     (item E)
     (item F)
     (item G)
     (item H)
     (item I)
     (item J)
```

```
  (at p1 A)
     (at p1 B)
     (at p2 C)
     (at p2 D)
     (at p3 E)
     (at p3 F)
     (at p3 G)
     (ship ship)
     (empty ship)
     (at p1 ship)
  )
  (:goal (and
     (at p1 E)
     (at p1 F)
     (at p2 B)
     (at p2 G)
     (at p3 A)
     (at p3 D)
     (at p3 C)
    )
  )
)
```

Hanojské věže



Hanojské věže – obecný popis problému

```
(define (domain hanoi)
                (:requirements :strips)
 (:predicates
                (clear ?x)
                (on ?x ?y)
                (smaller ?x ?y)
 )
 (:action move
                :parameters (?disc ?from ?to)
                :precondition (and (smaller ?to ?disc)
                                                (on ?disc ?from)
                                                (clear ?disc)
                                                (clear ?to)
                                    )
                :effect (and (clear ?from)
                                    (on ?disc ?to)
                                    (not (on ?disc ?from))
                                    (not (clear ?to))
                                )
 )
)
```

Hanojské věže – konkrétní konfigurace

```
(define (problem hanoi-3)
  (:domain hanoi)
  (:requirements :strips)
  (:objects p1 p2 p3 d1 d2 d3)
  (:init
                (smaller p1 d1)(smaller p1 d2)(smaller p1 d3)
                (smaller p2 d1)(smaller p2 d2)(smaller p2 d3)
                (smaller p3 d1)(smaller p3 d2)(smaller p3 d3)
                (smaller d2 d1)(smaller d3 d1)(smaller d3 d2)
                (clear p2) (clear p3) (clear d1)
                (on d3 p1)
                (on d2 d3)
                (on d1 d2)
  )
  (:goal (and (on d3 p3)
                    (on d2 d3)
                    (on d1 d2))
  )
)
```