

# Artificial intelligence - Project3

## - Planificare clasica -

Stecko Daiana - Viman Andrei - Timis Iulia

14 January 2022

### 1 Ferma animalelor

Noi am implmentat o aplicatie de plannning in care se realizeaza procesul de hranire la o ferma. Animalele locuiesc in niste boxe. Deasemenea avem boxe pentru depozitarea hranei si apei animalelor. Fermierii trebuie sa hraneasca toate animalele, iar un fermier poate hrani un singur animal o data. Fermierii trebuie sa le dea la toate animalele apa, iar un fermier poate adapa un singur animal o data. Boxa animalelor trebuie curatata dupa fiecare hranire, asa ca fermierul are obligatia de a o curata. Odata cu curatarea boxelor fermierul se poate odihni. Dupa terminarea tuturor activitatilor la ferma, fermierul poate pleaca acasa.



Figure 1: Ferma

## 2 Obiectiv

Obiectivul nostru este acela de a rezolva mai multe probleme care apar in viata unui fermier. De exemplu,ingrijirea animalelor prin curatarea locului unde stau si hranirea acestora,.

## 3 Domeniu

Domeniul contine predicatele pentru a putea determina dacă exista un fermier, daca am oferit apa/ mancare animalelor,daca avem un loc de depozitare pentru mancare, daca exista o boxa pentru animale, daca aceasta este curata si locatia fermierului.

Ca si actiuni avem:

- 1.Luare-hrana - in cazul acestei actiunii fermierul este instiintat daca animalele au nevoie de hrana.
- 2.Hranire - aici fermierul va hranii animalul.
- 3.Deplasare - fermierul aduce apa/mancare animalului.
- 4.Curatare - dupa ce hrana a fost consumata de catre animal, boxa acestuia va fi curatata de catre fermier.
- 5.Odihnire - dupa ce boxa este curata si animalul este hranit, fermierul se va putea odihni.
- 6.Luare-apa - fermierul este instiintat daca animalul are nevoie de apa.
- 7.Adapare - in care animalul va primi apa.
- 8.MersAcasa - dupa ce s-a odihnit, fermierul va putea merge acasa.

### 3.1 Codul pentru domeniu

```
;Aplcatie de plannnning hranire la ferma. Exista boxe, in care pot fi animale
;sau in care pot exista si surse de hrana.
;Fermierii trebuie sa hraneasca toate animalele, iar un fermier poate hrani
;un singur animal o data
```

```
(define (domain ferma)
  (:requirements :strips)
  (:predicates
    (de-hranit ?boxa); animalul dn boxa boxa trebuie hranit
    (de-adapat ?boxa); animalul dn boxa boxa trebuie hranit
    (sursahrana ?boxa) ; boxa e sursahrana
    (sursaapa ?boxa)
    (fermier ?fermier) ; fermier e fermier
    (boxa ?boxa) ; e boxa
    (animal ?boxa) ; la boxa boxa exista un animal
    (luat-hrana ?fermier) ; la fermierul fermier exista hrana
    (luat-apa ?fermier) ; la fermierul fermier exista apa
    (hranit ?boxa) ; animalul din boxa boxa este hranit
```

```

(adapat ?boxa) ; animalul din boxa boxa este adaptat
(liber ?fermier) ; fermierul este liber
(locatie-fermier ?boxa) ; fermierul e la boxa
(boxa-murdara ?boxa) ; boxa e murdara
(boxa-curata ?boxa) ; boxa e curata
(curatare ?boxa)
(odihnire ?fermier)
(mers-acasa ?fermier)

)

(:action luare-hrana ; fermierul e instiintat daca un animal are nevoie de hrana
:parameters
  (?boxa ?fermier)
:precondition
  (and (fermier ?fermier) (boxa ?boxa) (sursahrana ?boxa)
        (locatie-fermier ?boxa))
:effect
  (and (not (liber ?fermier)) (luat-hrana ?fermier) ))

(:action hranire ; hranire animal
:parameters (?fermier ?boxa)
:precondition
  (and (luat-hrana ?fermier) (animal ?boxa) (de-hranit ?boxa)
        (locatie-fermier ?boxa))
:effect
  (and (hranit ?boxa) (not(de-hranit ?boxa)) (not (luat-hrana ?fermier))
        (boxa-murdara ?boxa)))

(:action deplasare ; fermierul duce mancarea la animal
:parameters
  (?boxa-plecare ?boxa-sosire)
:precondition
  (and (boxa ?boxa-plecare) (boxa ?boxa-sosire)
        (locatie-fermier ?boxa-plecare))
:effect
  (and (not (locatie-fermier ?boxa-plecare)) (locatie-fermier ?boxa-sosire)))

```

```
(:action curatare
:parameters (?fermier ?boxa)
:precondition
  (and (hranit ?boxa) (not(de-hranit ?boxa)) (not (luat-hrana ?fermier))
        (boxa-murdara ?boxa))
:effect
  (and (curatare ?boxa ) (not( boxa-murdara ?boxa)) (sursaapa ?boxa)
        (not(odihnire ?fermier))))
```

```
(:action odihnire
:parameters(?fermier ?boxa)
:precondition
  (and (curatare ?boxa ) (not( boxa-murdara ?boxa)) (sursaapa ?boxa)
        (not(odihnire ?fermier)))
:effect
  (and (odihnire ?fermier)))
```

```
(:action luare-apa ;fermierul e instiintat daca un animal are nevoie de apa
:parameters (?fermier ?boxa)
:precondition
  (and (fermier ?fermier) (boxa ?boxa) (sursaapa ?boxa) (curatare ?boxa )
        (not( boxa-murdara ?boxa)) )
:effect
  (and (not (liber ?fermier)) (luat-apa ?fermier)))
```

```
(:action adapare ;hranire animal
:parameters (?fermier ?boxa)
:precondition
  (and (luat-apa ?fermier) (animal ?boxa) (locatie-fermier ?boxa))
:effect
  (and (adapat ?boxa) (mers-acasa ?fermier)))
```

```

(:action mersAcasa ;hranire animal
:parameters (?fermier ?boxa)
:precondition
  (and (adapat ?boxa) (odihnire ?fermier) (locatie-fermier ?boxa)
        (mers-acasa ?fermier))
:effect
  (and (liber ?fermier) ))
)

```

## 4 Problemele

În probleme definim mai multi fermieri cu diferite nume, locul in care se afla, adica la o ferma si actiunile pe care urmeaza sa le faca el. Spre exemplu el poate sa faca actiuni de hranire, de adapare, de curatare, dar in acelasi timp el poate sa se odihneasca doar dupa ce a executat o actiune de curatare. Si nu va putea pleca acasa dect atunci cand isi va termina toata treaba.

Signalele Goal-urile ale fermierilor sunt de a-si termina treaba la ferma sau de a pleca acasa dupa ce au terminat treaba.

### 4.1 Codul pentru problema 1

```

(define (problem problema1)
  (:domain ferma)
  (:objects Vasile boxa1 boxa2 boxa3 boxa-hrana)
  (:init
    (fermier Vasile)
    (liber Vasile)
    (locatie-fermier boxa-hrana)
    (boxa boxa1)
    (boxa boxa2)
    (boxa boxa3)
    (boxa boxa-hrana)
    (sursahrana boxa-hrana)
    (animal boxa1)
    (animal boxa2)
    (animal boxa3)
    (de-hranit boxa1)
    (de-hranit boxa2)
    (de-hranit boxa3)
  )
  (:goal (and (hranit boxa1)
              (hranit boxa2)

```

```

(hranit boxa3)
(curatare boxa1)
(curatare boxa2)
(curatare boxa3)
(adapat boxa1)
(adapat boxa2)
(adapat boxa3)
))
)

```

## 4.2 Codul pentru problema 2

```

(define (problem problema1)
  (:domain ferma)
  (:objects Mihai boxa1 boxa2 boxa3 boxa-hrana)
  (:init
    (fermier Mihai)
    (liber Mihai)
    (locatie-fermier boxa-hrana)
    (boxa boxa1)
    (boxa boxa-hrana)
    (sursahrana boxa-hrana)
    (animal boxa1)

    (de-hranit boxa1)

  )
  (:goal (and
    (curatare boxa1)
    (odihnire Mihai)
    (liber Mihai)
  ))
)

```

## 4.3 Codul pentru problema 3

```

(define (problem problema1)
  (:domain ferma)
  (:objects Vasile boxa1 boxa2 boxa3 boxa-hrana)
  (:init
    (fermier Vasile)
    (liber Vasile)
    (locatie-fermier boxa-hrana)
    (boxa boxa1)
  )
)

```

```

        (boxa boxa2)
        (boxa boxa3)
        (boxa boxa-hrana)
        (sursahrana boxa-hrana)
        (animal boxa1)
        (animal boxa2)
        (animal boxa3)
        (de-hranit boxa1)
        (de-hranit boxa2)
        (de-hranit boxa3)
    )
    (:goal (and (hranit boxa1)
                (hranit boxa2)
                (hranit boxa3)

                ))
    )

```

## 5 Fisierul de iesire

### 5.1 Fisierul de iesire pentru problema 1

```

;rulata cu:
;./fast-downward.py projdomain1.pddl projproblem1.pddl
;--heuristic "h=ff()"
;--search "astar(h)"

(luare-hrana boxa-hrana vasile)
(deplasare boxa-hrana boxa1)
(hranire vasile boxa1)
(curatare vasile boxa1)
(luare-apa vasile boxa1)
(adapare vasile boxa1)
(deplasare boxa1 boxa-hrana)
(luare-hrana boxa-hrana vasile)
(deplasare boxa-hrana boxa2)
(adapare vasile boxa2)
(hranire vasile boxa2)
(deplasare boxa2 boxa-hrana)
(luare-hrana boxa-hrana vasile)
(deplasare boxa-hrana boxa3)
(hranire vasile boxa3)
(adapare vasile boxa3)
(curatare vasile boxa2)

```

```

(curatare vasile boxa3)
; cost = 18 (unit cost)

;rulata cu:
;./fast-downward.py projdomain1.pddl projproblem1.pddl
;--heuristic "h=ff()"
;--search "lazy-greedy([ff()], preferred=[ff()])"

(luare-hrana boxa-hrana vasile)
(deplasare boxa-hrana boxa1)
(hranire vasile boxa1)
(curatare boxa-hrana boxa1)
(luare-apa vasile boxa1)
(adapare vasile boxa1)
(deplasare boxa1 boxa-hrana)
(luare-hrana boxa-hrana vasile)
(deplasare boxa-hrana boxa2)
(adapare vasile boxa2)
(deplasare boxa2 boxa3)
(adapare vasile boxa3)
(hranire vasile boxa3)
(deplasare boxa3 boxa-hrana)
(luare-hrana boxa-hrana vasile)
(deplasare boxa-hrana boxa2)
(hranire vasile boxa2)
(curatare boxa-hrana boxa2)
(curatare boxa-hrana boxa3)
; cost = 19 (unit cost)

;rulata cu: ;
;./fast-downward.py projdomain1.pddl projproblem1.pddl
;--heuristic "h=cg()"
;--search "astar(h)"
(luare-hrana boxa-hrana vasile)
(deplasare boxa-hrana boxa1)
(hranire vasile boxa1)
(curatare vasile boxa1)
(deplasare boxa1 boxa-hrana)
(luare-hrana boxa-hrana vasile)
(deplasare boxa-hrana boxa2)
(luare-apa vasile boxa1)
(adapare vasile boxa2)
(deplasare boxa2 boxa1)

```



```

(adapare vasile boxa1)
(deplasare boxa1 boxa3)
(adapare vasile boxa3)
(hranire vasile boxa3)
(deplasare boxa3 boxa-hrana)
(luare-hrana boxa-hrana vasile)
(deplasare boxa-hrana boxa2)
(hranire vasile boxa2)
(curatare vasile boxa2)
(curatare vasile boxa3)
; cost = 20 (unit cost)

; rulata cu:
; ./fast-downward.py projdomain1.pddl projproblem1.pddl
; --heuristic "h=cg()"
; --search "lazy-greedy([cg()], preferred=[cg()])"

(luare-hrana boxa-hrana vasile)
(deplasare boxa-hrana boxa1)
(hranire vasile boxa1)
(deplasare boxa1 boxa2)
(deplasare boxa2 boxa-hrana)
(luare-hrana boxa-hrana vasile)
(deplasare boxa-hrana boxa2)
(curatare boxa3 boxa1)
(luare-apa vasile boxa1)
(adapare vasile boxa2)
(deplasare boxa2 boxa3)
(adapare vasile boxa3)
(hranire vasile boxa3)
(deplasare boxa3 boxa2)
(curatare boxa3 boxa3)
(deplasare boxa2 boxa-hrana)
(luare-hrana boxa-hrana vasile)
(deplasare boxa-hrana boxa2)
(hranire vasile boxa2)
(deplasare boxa2 boxa1)
(adapare vasile boxa1)
(curatare boxa3 boxa2)
; cost = 22 (unit cost)

```

## 5.2 Fisierul de iesire pentru problema 2

```

; rulata cu:
; ./fast-downward.py projdomain1.pddl projproblem2.pddl
; --heuristic "h=ff()"
; --search "astar(h)"

```

```

(luare-hrana boxa-hrana mihai)
(deplasare boxa-hrana boxa1)
(hranire mihai boxa1)
(curatare mihai boxa1)
(luare-apa mihai boxa1)
(adapare mihai boxa1)
(odihnire mihai boxa1)
(mersacasa mihai boxa1)
; cost = 8 (unit cost)

```

```

; rulata cu:
; ./fast-downward.py projdomain1.pddl projproblem2.pddl
; --heuristic "h=ff()"
; --search "lazy-greedy([ff()], preferred=[ff()])"

```

```

(luare-hrana boxa-hrana mihai)
(deplasare boxa-hrana boxa1)
(hranire mihai boxa1)
(curatare boxa-hrana boxa1)
(luare-apa mihai boxa1)
(adapare mihai boxa1)
(odihnire mihai boxa1)
(mersacasa mihai boxa1)
; cost = 8 (unit cost)

```

```

; rulata cu:
; ./fast-downward.py projdomain1.pddl projproblem2.pddl
; --heuristic "h=cg()"
; --search "astar(h)"

```

```

(luare-hrana boxa-hrana mihai)
(deplasare boxa-hrana boxa1)
(hranire mihai boxa1)
(curatare mihai boxa1)
(luare-apa mihai boxa1)
(adapare mihai boxa1)
(odihnire mihai boxa1)
(mersacasa mihai boxa1)
; cost = 8 (unit cost)

```

```

; rulata cu:
; ./fast-downward.py projdomain1.pddl projproblem2.pddl
; --heuristic "h=cg()"
; --search "lazy-greedy([cg()], preferred=[cg()])"

(luare-hrana boxa-hrana mihai)
(deplasare boxa-hrana boxa1)
(hranire mihai boxa1)
(curatare boxa3 boxa1)
(luare-apa mihai boxa1)
(adapare mihai boxa1)
(odihnire mihai boxa1)
(mersacasa mihai boxa1)
; cost = 8 (unit cost)

```

### 5.3 Fisierul de iesire pentru problema 3

```

; rulata cu:
; ./fast-downward.py projdomain1.pddl projproblem3.pddl
; --heuristic "h=ff()"
; --search "astar(h)"

(luare-hrana boxa-hrana victor)
(deplasare boxa-hrana boxa1)
(hranire victor boxa1)
(deplasare boxa1 boxa-hrana)
(luare-hrana boxa-hrana victor)
(deplasare boxa-hrana boxa2)
(hranire victor boxa2)
(deplasare boxa2 boxa-hrana)
(luare-hrana boxa-hrana victor)
(deplasare boxa-hrana boxa3)
(hranire victor boxa3)
; cost = 11 (unit cost)

```

```

; rulata cu:
; ./fast-downward.py projdomain1.pddl projproblem3.pddl
; --heuristic "h=ff()"
; --search "lazy-greedy([ff()], preferred=[ff()])"

```

```

(luare-hrana boxa-hrana victor)
(deplasare boxa-hrana boxa1)
(hranire victor boxa1)
(deplasare boxa1 boxa-hrana)
(luare-hrana boxa-hrana victor)
(deplasare boxa-hrana boxa2)
(hranire victor boxa2)
(deplasare boxa2 boxa-hrana)
(luare-hrana boxa-hrana victor)
(deplasare boxa-hrana boxa3)
(hranire victor boxa3)
; cost = 11 (unit cost)

```

```

; rulata cu:
; ./fast-downward.py projdomain1.pddl projproblem3.pddl
; --heuristic "h=cg()"
; --search "astar(h)"

```

```

(luare-hrana boxa-hrana victor)
(deplasare boxa-hrana boxa1)
(hranire victor boxa1)
(deplasare boxa1 boxa-hrana)
(luare-hrana boxa-hrana victor)
(deplasare boxa-hrana boxa2)
(hranire victor boxa2)
(deplasare boxa2 boxa-hrana)
(luare-hrana boxa-hrana victor)
(deplasare boxa-hrana boxa3)
(hranire victor boxa3)
; cost = 11 (unit cost)

```

```

; rulata cu:
; ./fast-downward.py projdomain1.pddl projproblem3.pddl
; --heuristic "h=cg()"
; --search "lazy-greedy([cg()], preferred=[cg()])"

```

```

(luare-hrana boxa-hrana victor)
(deplasare boxa-hrana boxa1)
(hranire victor boxa1)
(deplasare boxa1 boxa2)

```

```
(deplasare boxa2 boxa—hrana)
(luare—hrana boxa—hrana victor)
(deplasare boxa—hrana boxa2)
(hranire victor boxa2)
(deplasare boxa2 boxa3)
(deplasare boxa3 boxa—hrana)
(luare—hrana boxa—hrana victor)
(deplasare boxa—hrana boxa3)
(hranire victor boxa3)
; cost = 13 (unit cost)
```