



**KAUNO TECHNOLOGIJOS UNIVERSITETAS**

**INFORMATIKOS FAKULTETAS**

**INTELEKTIKOS PAGRINDAI (P176B101)**

## **1 laboratorinis darbas**

Atliko: IFF-6/11 gr. studentas Nerijus Dulkė  
Priėmė: doc. Germanas Budnikas

KAUNAS  
2019

## 1. Pirma užduotis

### 1.1. Užduotis

Duota užduotis:

Pelės:

- Jos baltos ir pilkos
- Jų yra daug

Katinas:

- Alkanas
- Kai katinas alkanas, jis nori valgyti
- Kai katinas nori valgyti ir yra pelių, jis valgo peles baltas ir pilkas

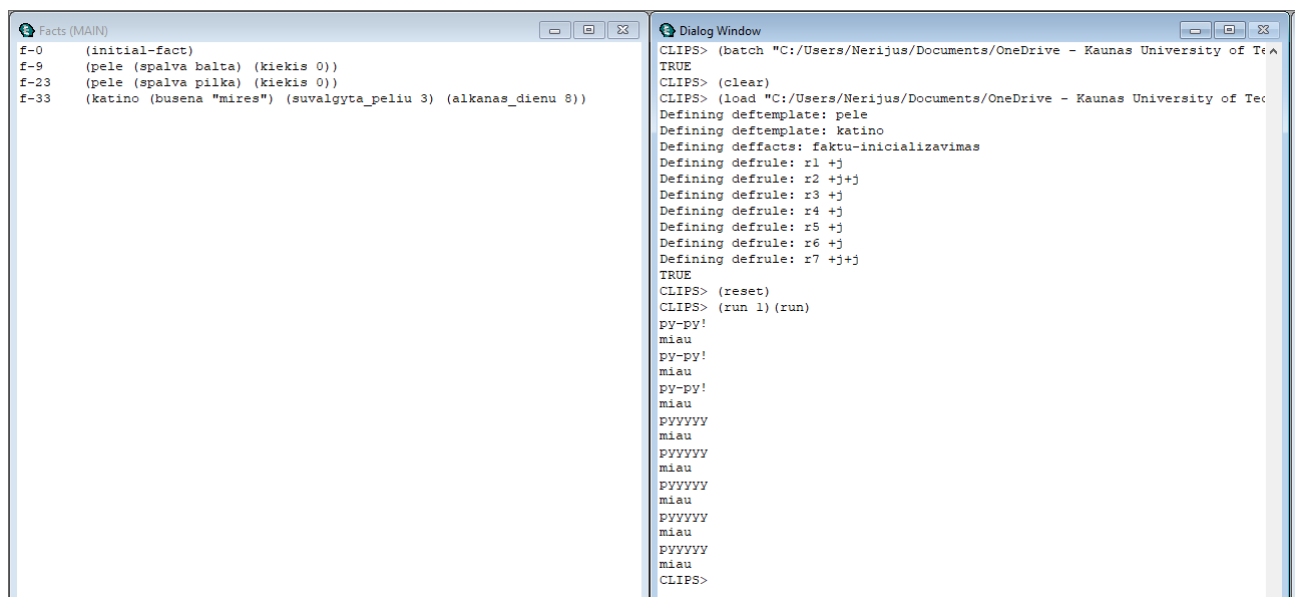
- Prieš mirt baltoji pelė *šaukia* „py-py!“, pilkoji – „pyyyyyy“  
Po valgio katinas paprastai *sako* „miau“ ;-)
- Kai katinas suvalgo 5 peles, jis tampa storu katinu
- Stori katinai pelių nevalgo

Užduotyje reikėjo savarankiškai atlikti papildymus:

1. Stori katinai po miego tampa alkani ir pamiršta, kiek suvalgė pelių
2. Nevalgę bet vis dar norintys valgyti katinai po 7 dienų miršta

### 1.2. Realizacija

Išvestis kai programa baigia vykdyti:



The screenshot shows two windows from a CLIPS environment. The left window, titled 'Facts (MAIN)', displays a list of facts: f-0 (initial-fact), f-9 (pele (spalva balta) (kiekis 0)), f-23 (pele (spalva pilka) (kiekis 0)), and f-33 (katino (busena "mires") (suvalgyta\_peliu 3) (alkanas\_dienu 8)). The right window, titled 'Dialog Window', shows the CLIPS command line interface. It contains the following text: CLIPS> (batch "C:/Users/Nerijus/Documents/OneDrive - Kaunas University of Te... TRUE CLIPS> (clear) CLIPS> (load "C:/Users/Nerijus/Documents/OneDrive - Kaunas University of Te... Defining deftemplate: pele Defining deftemplate: katino Defining deffacts: faktu-inicializavimas Defining defrule: r1 +j Defining defrule: r2 +j+j Defining defrule: r3 +j Defining defrule: r4 +j Defining defrule: r5 +j Defining defrule: r6 +j Defining defrule: r7 +j+j TRUE CLIPS> (reset) CLIPS> (run 1) (run) py-py! miau py-py! miau py-py! miau pyyyyy miau pyyyyy miau pyyyyy miau pyyyyy miau pyyyyy miau CLIPS>

Programos kodas:

```
(deftemplate pele (slot spalva) (slot kiekis) )
(deftemplate katino (slot busena) (slot suvalgyta_peliu) )
(deftemplate skaitliukas (slot dienu_kiekis))

(deffacts faktu-inicializavimas
  (pele (spalva pilka) (kiekis 5))
  (pele (spalva balta) (kiekis 3))
  (katino (busena "alkanas") (suvalgyta_peliu 0))
  (skaitliukas (dienu_kiekis 0))
)

(defrule r1 "Kai katinas alkanas, jis nori valgyti"
  ?fact-id <- (katino (busena ?busena))
  (test (eq ?busena "alkanas"))
  =>
  (modify ?fact-id (busena "nori valgyti"))
)

(defrule r2 "Kai katinas nori valgyti ir yra peliu, jis valgo peles"
  ?fact-id1 <- (katino (busena "nori valgyti") (suvalgyta_peliu ?suvalgyta))
  ?fact-id2 <- (pele (spalva ?spalva) (kiekis ?kiekis))
  (test (> ?kiekis 0))
  =>

  (if (eq ?spalva balta) then (printout t "py-py!" crlf)
      else (printout t "pyyyyy" crlf))
  (modify ?fact-id2 (kiekis (- ?kiekis 1)) )

  (modify ?fact-id1 (suvalgyta_peliu (+ ?suvalgyta 1)) )
  (printout t "miau" crlf)
)

(defrule r3 "kai katinas suvalgo 5 peles, jis tampa storu katinu"
  (declare (salience 10))
  ?fact-id1 <- (katino (busena "nori valgyti") (suvalgyta_peliu ?suvalgyta))
  (test (= ?suvalgyta 5))
  =>
  (modify ?fact-id1 (busena "storas"))
)

(defrule r4 "Kai katinas storas jis miega"
  ?fact-id <- (katino (busena ?busena))
  (test (eq ?busena "storas"))
  =>
  (bind ?busena "miega")
  (modify ?fact-id (busena ?busena))
)

(defrule r5 "Stori katinai po miego tampa alkani ir pamirsta kiek suvalge peliu"
```

```

    ?fact-id <- (katino (busena ?busena))
    (test (eq ?busena "miega"))
    =>
    (modify ?fact-id (busena "alkanas")(suvalgyta_peliu 0))
  )

(defrule r6 "Po 7 dienu norintys valgyt katinai mirsta"
  ?fact-id1 <- (katino (busena ?busena))
  ?fact-id2 <- (skaitliukas (dienu_kiekis ?dienu_kiekis))
  (test (eq ?busena "nori valgyti"))
  (test (> ?dienu_kiekis 7))
  =>
  (modify ?fact-id1 (busena "mires"))
)

(defrule r7 "Skaitliukas dideja kai nera peliu"
  ?fact-id1 <- (pele (kiekis ?kiekis))
  ?fact-id2 <- (skaitliukas (dienu_kiekis ?dienu_kiekis))
  (test (= ?kiekis 0))
  =>
  (modify ?fact-id2 (dienu_kiekis (+ ?dienu_kiekis 1)))
)

```

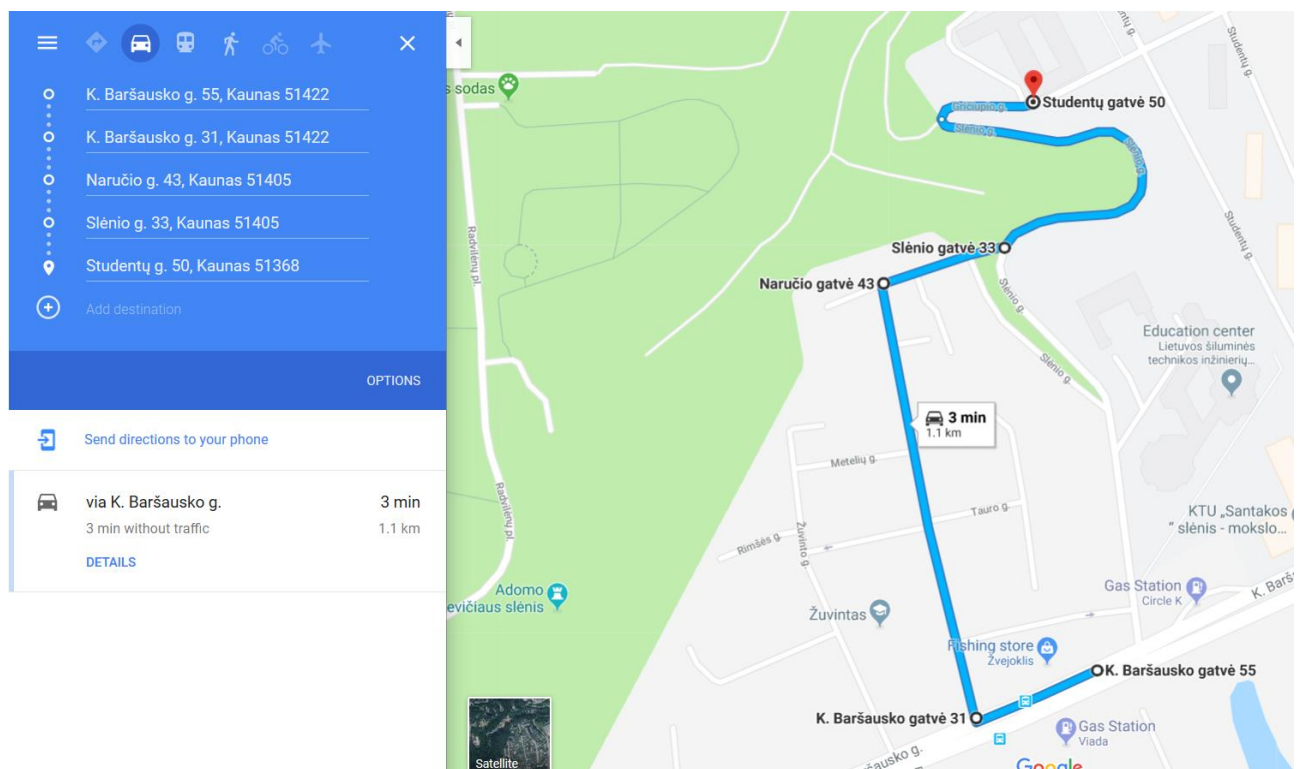
## 2. Individuali užduotis

### 2.1. Užduotis

Sukurti taisyklėmis grįstą sistemą, aprašančią autonominės mašinos judėjimą pasirinktu maršrutu. Maršrute, kuris skaidomas į fragmentus ties kiekvieną sankryžą, gali būti kliūčių - pėsčiųjų, šviesoforų, kitų mašinų, spec. tarnybų automobilių. Judėjimas maršruto atkarpoje leidžiamas tik nesusidūrus su kliūtimis. Sistemoje iliustruoti autonominės mašinos judėjimą nurodytu maršrutu informuojant apie pravažiuojamas gatves ir kliūčių sumažėjimą važiuojamoje maršruto atkarpoje. Numatyti galimybę programoje papildyti norimą kiekį kliūčių bei pakoreguoti maršrutą.

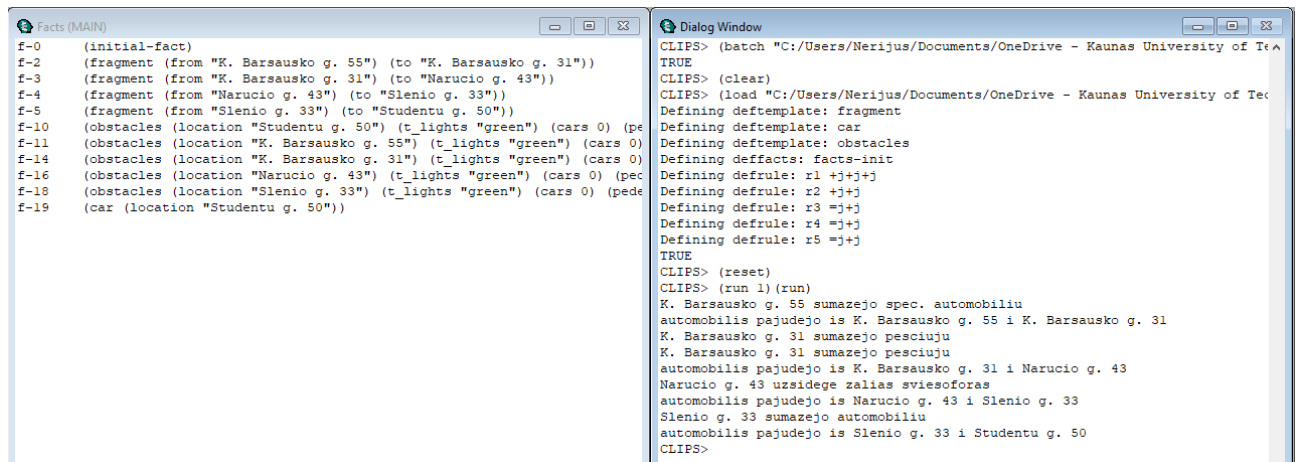
Maršrutui nustatyti parinkti savo namų vietą. Nustatyti judėjimo kryptį per x sankryžų, kur x – raidžių skaičius jūsų varde arba pavardėje (kas trumpiau).

Pasirinktas maršrutas:



## 2.2. Realizacija

Išvestis kai programa baigia vykdyti:



```
Facts (MAIN)
f-0 (initial-fact)
f-2 (fragment (from "K. Barsausko g. 55") (to "K. Barsausko g. 31"))
f-3 (fragment (from "K. Barsausko g. 31" (to "Narucio g. 43"))
f-4 (fragment (from "Narucio g. 43") (to "Slenio g. 33"))
f-5 (fragment (from "Slenio g. 33") (to "Studentu g. 50"))
f-10 (obstacles (location "Studentu g. 50") (t_lights "green") (cars 0) (pedestrians 0) (spec_service 1))
f-11 (obstacles (location "K. Barsausko g. 55") (t_lights "green") (cars 0) (pedestrians 2) (spec_service 0))
f-14 (obstacles (location "K. Barsausko g. 31") (t_lights "green") (cars 0) (pedestrians 0) (spec_service 0))
f-16 (obstacles (location "Narucio g. 43") (t_lights "green") (cars 0) (pedestrians 0) (spec_service 0))
f-18 (obstacles (location "Slenio g. 33") (t_lights "green") (cars 0) (pedestrians 0) (spec_service 0))
f-19 (car (location "Studentu g. 50"))

Dialog Window
CLIPS> (batch "C:/Users/Nerijus/Documents/OneDrive - Kaunas University of Technology/CLIPS/CLIPS.bat")
TRUE
CLIPS> (clear)
CLIPS> (load "C:/Users/Nerijus/Documents/OneDrive - Kaunas University of Technology/CLIPS/CLIPS.clp")
Defining deftemplate: fragment
Defining deftemplate: car
Defining deftemplate: obstacles
Defining deftemplate: facts-init
Defining defrule: r1 +j+j+j
Defining defrule: r2 +j+j+j
Defining defrule: r3 +j+j+j
Defining defrule: r4 +j+j+j
Defining defrule: r5 +j+j+j
TRUE
CLIPS> (reset)
CLIPS> (run 1) (run)
K. Barsausko g. 55 sumazejo spec. automobiliu
automobilis pajudejo is K. Barsausko g. 55 i K. Barsausko g. 31
K. Barsausko g. 31 sumazejo pasciuju
K. Barsausko g. 31 sumazejo pasciuju
automobilis pajudejo is K. Barsausko g. 31 i Narucio g. 43
Narucio g. 43 uzsidege zalias sviesoforas
automobilis pajudejo is Narucio g. 43 i Slenio g. 33
Slenio g. 33 sumazejo automobiliu
automobilis pajudejo is Slenio g. 33 i Studentu g. 50
CLIPS>
```

Programos kodas:

```
(deftemplate fragment (slot from) (slot to))
(deftemplate car (slot location))
(deftemplate obstacles (slot location) (slot t_lights) (slot cars) (slot pedestrians) (slot spec_service))

(deffacts facts-init
  (car (location "K. Barsausko g. 55"))
  (fragment (from "K. Barsausko g. 55") (to "K. Barsausko g. 31"))
  (fragment (from "K. Barsausko g. 31") (to "Narucio g. 43"))
  (fragment (from "Narucio g. 43") (to "Slenio g. 33"))
  (fragment (from "Slenio g. 33") (to "Studentu g. 50"))
  (obstacles (location "K. Barsausko g. 55") (t_lights "green") (cars 0) (pedestrians 0) (spec_service 1))
  (obstacles (location "K. Barsausko g. 31") (t_lights "green") (cars 0) (pedestrians 2) (spec_service 0))
  (obstacles (location "Narucio g. 43") (t_lights "red") (cars 0) (pedestrians 0) (spec_service 0))
  (obstacles (location "Slenio g. 33") (t_lights "green") (cars 1) (pedestrians 0) (spec_service 0))
  (obstacles (location "Studentu g. 50") (t_lights "green") (cars 0) (pedestrians 0) (spec_service 0))
)

(defrule r1 "Move"
  ?fact-id1 <- (fragment (from ?from)(to ?to))
  ?fact-id2 <- (car (location ?location))
  ?fact-id3 <- (obstacles (location ?obs_location) (t_lights "green") (cars 0) (pedestrians 0) (spec_service 0))
  (test (eq ?location ?from))
  (test (eq ?location ?obs_location))
=>
```

```

        (modify ?fact-id2 (location ?to))
        (printout t "automobilis pajudejo is " ?from " i " ?to crlf)
    )

(defrule r2 "Toggle traffic lights"
    ?fact-id1 <- (car (location ?location))
    ?fact-id2 <- (obstacles (location ?obs_location) (t_lights "red"))
    (test (eq ?location ?obs_location))
    =>
    (modify ?fact-id2 (t_lights "green"))
    (printout t ?location " uzsidege zalias sviesofofas" crlf)
)

(defrule r3 "Move cars"
    ?fact-id1 <- (car (location ?location))
    ?fact-id2 <- (obstacles (location ?obs_location) (cars ?cars))
    (test (eq ?location ?obs_location))
    (test (> ?cars 0))
    =>
    (modify ?fact-id2 (cars (- ?cars 1)))
    (printout t ?location " sumazejo automobiliu" crlf)
)

(defrule r4 "Move spec_services"
    ?fact-id1 <- (car (location ?location))
    ?fact-id2 <- (obstacles (location ?obs_location) (spec_service ?spec_service))
    (test (eq ?location ?obs_location))
    (test (> ?spec_service 0))
    =>
    (modify ?fact-id2 (spec_service (- ?spec_service 1)))
    (printout t ?location " sumazejo spec. automobiliu" crlf)
)

(defrule r5 "Move pedestrians"
    ?fact-id1 <- (car (location ?location))
    ?fact-id2 <- (obstacles (location ?obs_location) (pedestrians ?pedestrians))
    (test (eq ?location ?obs_location))
    (test (> ?pedestrians 0))
    =>
    (modify ?fact-id2 (pedestrians (- ?pedestrians 1)))
    (printout t ?location " sumazejo pesciuju" crlf)
)

```