**KAUNO TECHNOLOGIJOS UNIVERSITETAS**

**INFORMATIKOS FAKULTETAS**

INTELEKTIKOS PAGRINDAI (P176B101)

**1 laboratorinio darbo ataskaita.**

Atliko:

IFF-4/1 gr. studentas

Mangirdas Kazlauskas

Priėmė:

Doc. Germanas Budnikas

**KAUNAS 2017**

1. Užduotis „Katinas ir Pelės“

## Programos kodas

; JESS aplinkoje komentarus pasalinkite

;

; (clear)

(deftemplate pele (slot spalva) (slot kiekis) )

(deftemplate katino (slot busena) (slot suvalgyta\_peliu) (slot bado\_dienos))

(deffacts faktu-inicializavimas

(pele (spalva pilka) (kiekis 5))

(pele (spalva balta) (kiekis 3))

(katino (busena "alkanas") (suvalgyta\_peliu 0) (bado\_dienos 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 nori miego"

?fact-id <- (katino (busena ?busena))

(test (eq ?busena "storas"))

=>

(modify ?fact-id (busena "nori miego"))

)

(defrule r5 "Kai katinas nori miego, jis miega"

?fact-id <- (katino (busena ?busena))

(test (eq ?busena "nori miego"))

=>

(modify ?fact-id (busena "eina miegoti"))

)

(defrule r6 "Kai katinas miega, po to jis keliasi"

?fact-id <- (katino (busena ?busena))

(test (eq ?busena "eina miegoti"))

=>

(modify ?fact-id (busena "keliasi"))

)

(defrule r7 "Kai katinas keliasi, jis yra alkanas"

?fact-id1 <- (katino (busena ?busena))

(test (eq ?busena "keliasi"))

=>

(modify ?fact-id1 (busena "alkanas")(suvalgyta\_peliu 0))

)

(defrule r8 "Jei nera peliu, katinas badauja"

?fact-id <- (katino (busena "nori valgyti")(bado\_dienos ?badauja))

(pele (spalva balta)(kiekis 0))

(pele (spalva pilka)(kiekis 0))

=>

(modify ?fact-id (bado\_dienos (+ ?badauja 1)) )

)

(defrule r9 "Jei katinas badauja 7 dienas, jis mirsta"

(declare (salience 10))

?fact-id <- (katino (busena "nori valgyti")(bado\_dienos ?badauja))

(test (= ?badauja 7))

=>

(modify ?fact-id (busena "mirsta"))

)

; JESS aplinkoje komentarus pasalinkite

;

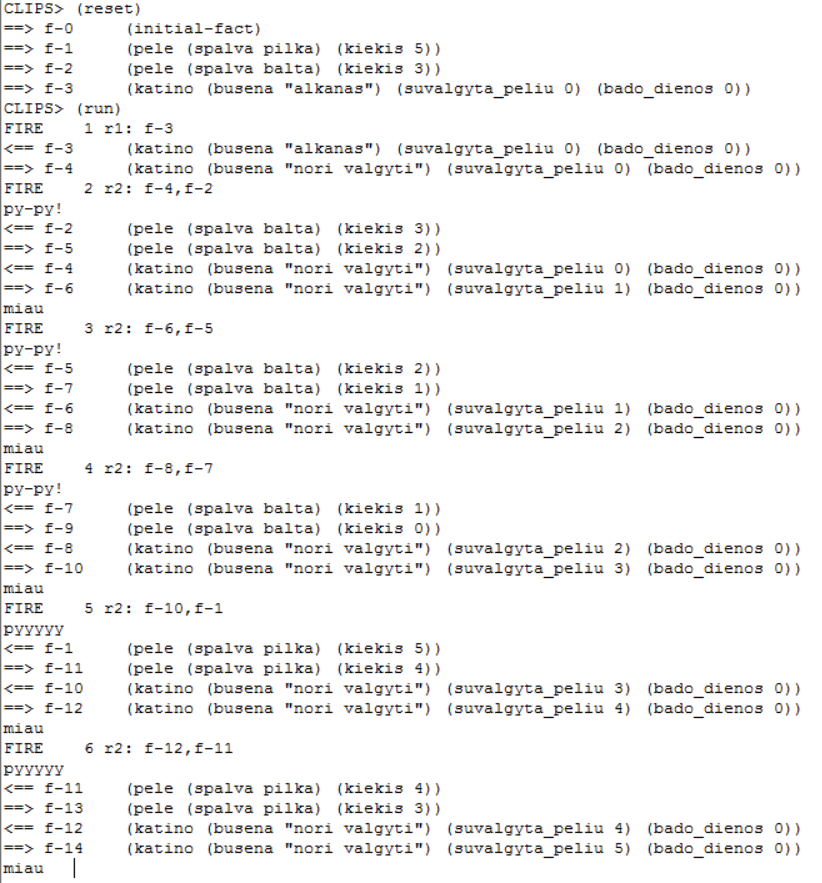
; (reset)

; (facts)

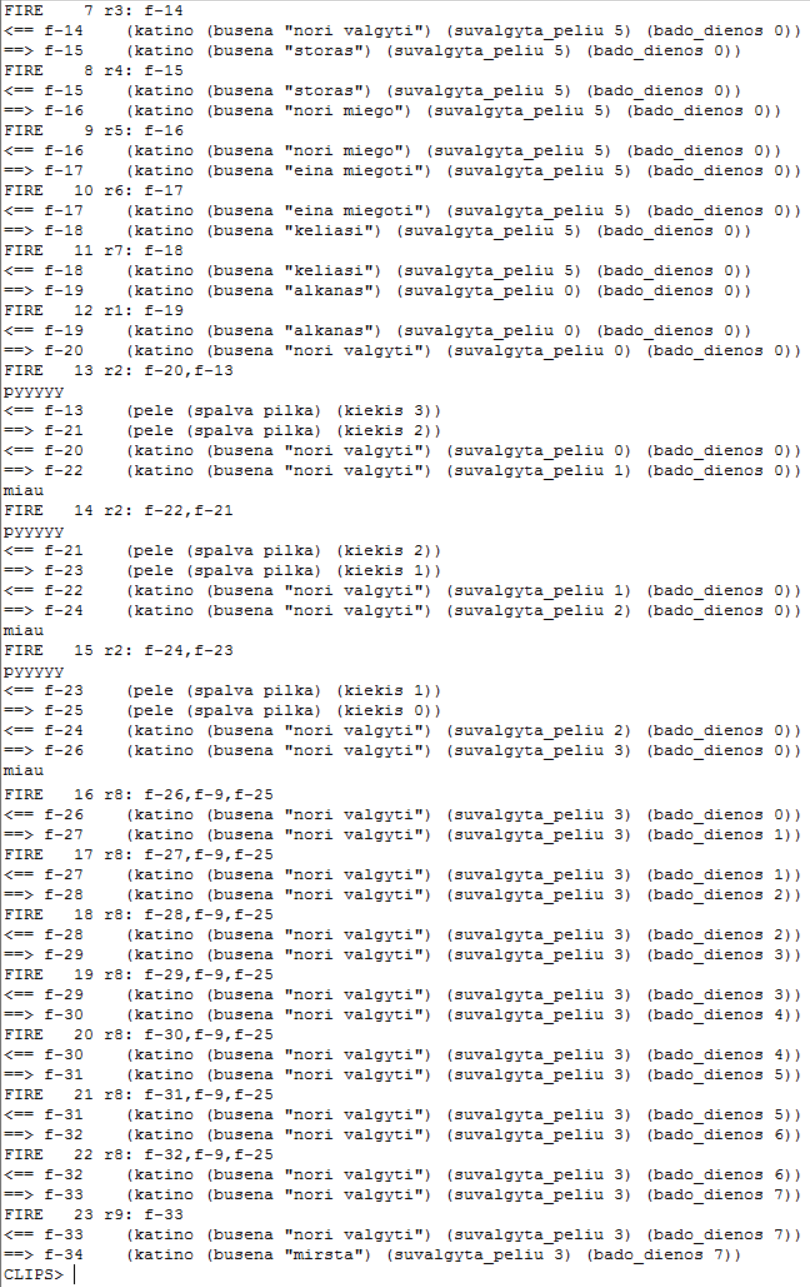
; (watch all)

; (run)

## Programos vykdymas



1 pav. Programos “Katinas ir pelės” vykdymas (FIRE 1-6)

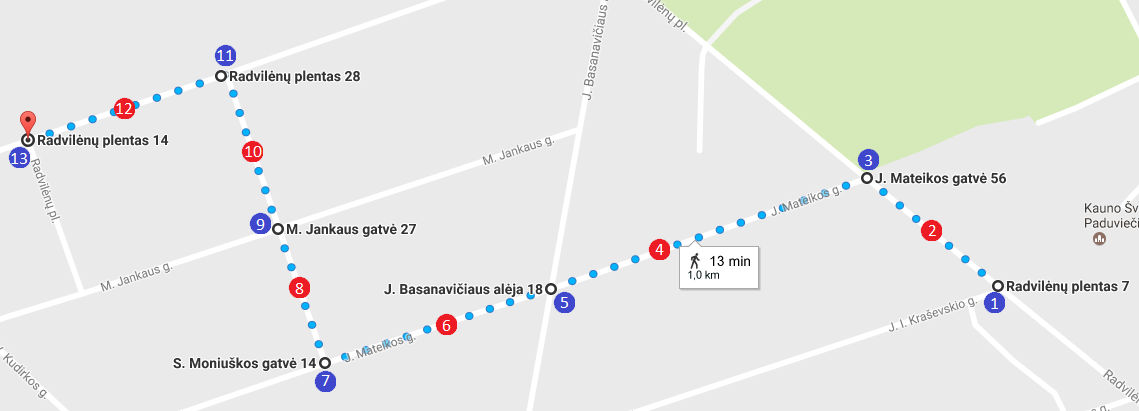


2 pav. Programos “Katinas ir pelės” vykdymas (FIRE 7-23)

# 1 laboratorinio darbo papildoma užduotis “autonominis automobilis”

## Užduoties aprašymas

Reikia sukurti taisyklėmis grįstą sistemą, aprašančia autonominės mašinos judėjimą duotu maršrutu (3 pav.). Maršrute, kuris skaidomas į fragmentus ties kiekviena sankryža, gali būti kliūčių – pėsčiųjų, šviesoforų, kitų mašinų, spec. tarnybų automobilių. Judėjimas maršruto atkarpoje leidžiamas tik nesant kliūtims. Sistema turi iliustruoti autonominės mašinos judėjimą duotu maršrutu, informuojant apie pravažiuojamas gatves ir kliūčių sumažėjimą važiuojamoje maršruto atkarpoje.



3 pav. Duotasis maršrutas, kurį turi įveikti autonominis automobilis

Maršruto paaiškinimai:

1. Mėlyni rutuliukai su skaičiais viduje žymi atitinkamų maršruto atkarpų pradinius ir galutinius taškus. Mėlynas rutuliukas, pažymėtas skaičiumi 1 – maršruto pradžios taškas, o mėlynas rutuliukas su didžiausia skaitine reikšme visada žymės maršruto pabaigos tašką (šiuo atveju, 13). Atkreipkite dėmesį, jog mėlynų rutuliukų skaitinės reikšmės yra tik nelyginiai skaičiai (reikalinga, kad tarp atkarpos pradžios ir pabaigos taškų būtų galima įterpti kliūties numerį maršrute).
2. Raudoni rutuliukai su skaičiais viduje žymi atitinkamų maršruto atkarpų vidinį tašką, atitinkantį maršruto kelyje sutinkamas kliūtis. Raudono rutuliuko skaitinė reikšmė visada bus didesnė už artimesnio gretimo mėlyno rutuliuko (atkarpos pradžios taško) reikšmę ir visada mažesnė už tolimesnio gretimo mėlyno rutuliuko (atkarpos pabaigos taško) reikšmę.
3. Mėlyna taškuota linija žymi autonominio automobilio kelionės maršrutą.

## Užduoties sprendimas

Užduotis realizuota tokiu principu: visos autonominio automobilio kelionės metu yra sekama mašinos padėtis, kuri kelionės pradžioje atitinką 0. Kai automobilis pasiekia patį pirmą maršruto tašką (mėlyną rutuliuką, su reikšme 1), pradedama autonominio automobilio kelionė nurodytu maršrutu. Kai automobilis yra bet kuriame maršruto atkarpos pradiniame taške (mėlyname kelio atkarpos rutuliuke, su mažesne reikšme), sistema tikrinama, ar gretimoje kelio atkarpoje (raudonas rutuliukas, kurio skaitinė reikšmė yra vienetu didesnė už pradinį tašką) yra kliūčių. Jei kliūčių yra, automobilis laukia, kol visos kliūtys pasišalins iš kelio atkarpos (kliūčių pasišalinimas yra simuliuojamas programos vykdymo metu). Kai kelio ruože kliūčių nėra, automobilis juda į maršruto atkarpos pabaigos tašką, kuris tampa kitos maršruto kelio atkarpos pradžios tašku. Tokiu būdu autonominis automobilis apkeliauja visą maršrutą iki pat jo pabaigos.

## Programos kodas

; JESS aplinkoje komentarus pasalinkite

;

; (clear)

; Templates

(deftemplate fragment

(slot from) (slot to) (slot from\_street\_name) (slot to\_street\_name))

(deftemplate car

(slot location))

(deftemplate obstacles

(slot location) (slot tlights) (slot cars) (slot pedestrians) (slot spec\_service))

; Facts

(deffacts facts-initialization

; Car initialization

(car (location 0))

; Route initialization

(fragment (from 1) (to 3) (from\_street\_name "Radvilenu pl.") (to\_street\_name "J.Mateikos g."))

(fragment (from 3) (to 5) (from\_street\_name "J.Mateikos g.") (to\_street\_name "J.Basanaviciaus al."))

(fragment (from 5) (to 7) (from\_street\_name "J.Basanaviciaus al.") (to\_street\_name "S.Moniuskos g."))

(fragment (from 7) (to 9) (from\_street\_name "S.Moniuskos g.") (to\_street\_name "M.Jankaus g."))

(fragment (from 9) (to 11) (from\_street\_name "M.Jankaus g.") (to\_street\_name "Radvilenu pl."))

(fragment (from 11) (to 13) (from\_street\_name "Radvilenu pl.") (to\_street\_name "Radvilenu pl."))

; Obstacles initialization

(obstacles (location 2) (tlights "green") (cars 2) (pedestrians 1) (spec\_service 0))

(obstacles (location 4) (tlights "green") (cars 5) (pedestrians 0) (spec\_service 0))

(obstacles (location 6) (tlights "red") (cars 3) (pedestrians 2) (spec\_service 1))

(obstacles (location 8) (tlights "green") (cars 0) (pedestrians 4) (spec\_service 2))

(obstacles (location 10) (tlights "red") (cars 2) (pedestrians 0) (spec\_service 0))

(obstacles (location 12) (tlights "yellow") (cars 1) (pedestrians 2) (spec\_service 1)))

; Rules

(defrule r1 "Start and finish of the trip"

?fact-id <- (car (location ?current\_location))

(fragment (to ?to1))

(not (fragment (to ?to2&:(> ?to2 ?to1)))) ; Searching for the fact "fragment" which has maximum slot "to" value among facts

(test

(or

(= ?current\_location 0) ; Location - start

(= ?current\_location ?to1) ; Location - finish

)

)

=>

(if (= ?current\_location 0)

then (modify ?fact-id (location 1)) (printout t "Car has started the trip." crlf)

else (modify ?fact-id (location (+ ?to1 1))) (printout t "Car has reached the destination." crlf)

)

)

(defrule r2 "Car movement"

?fact-id <- (car (location ?current\_location))

(fragment (from ?from) (to ?to) (from\_street\_name ?fsn) (to\_street\_name ?tsn))

(obstacles (location ?obs\_loc) (tlights ?tlights) (cars ?cars) (pedestrians ?pedestrians) (spec\_service ?sp\_serv))

(test

; No obstacles in front of the car

(and

(> ?obs\_loc ?current\_location)

(< ?obs\_loc (+ ?current\_location 2))

(eq ?tlights "green")

(= ?cars 0)

(= ?pedestrians 0)

(= ?sp\_serv 0)

(= ?from ?current\_location)

)

)

=>

(modify ?fact-id (location (+ ?current\_location 2)))

(printout t "No obstacles left in path No. " ?obs\_loc " and traffic light (if there is one) is green. Car moves from \"\(" ?from "\) " ?fsn "\" to \"\(" ?to "\) " ?tsn "\"." crlf)

)

(defrule r3 "Obstacles simulation"

(car (location ?current\_location))

?fact-id <- (obstacles (location ?obs\_loc) (tlights ?tlights) (cars ?cars) (pedestrians ?pedestrians) (spec\_service ?sp\_serv))

(test

(and

(= ?obs\_loc (+ ?current\_location 1) )

(or

(neq ?tlights "green")

(!= ?cars 0)

(!= ?pedestrians 0)

(!= ?sp\_serv 0)

)

)

)

=>

(if (> ?sp\_serv 0)

then (modify ?fact-id (spec\_service (- ?sp\_serv 1))) (printout t "Car lets special service car through." crlf)

else (if (> ?pedestrians 0)

then (modify ?fact-id (pedestrians (- ?pedestrians 1))) (printout t "Car lets pedestrians through." crlf)

else (if (> ?cars 0)

then (modify ?fact-id (cars (- ?cars 1))) (printout t "Car lets another car through." crlf)

else (if (eq ?tlights "red")

then (modify ?fact-id (tlights "yellow")) (printout t "Traffic light is red at the moment. Car is waiting for the traffic light to change to yellow." crlf)

else (if (eq ?tlights "yellow")

then (modify ?fact-id (tlights "green")) (printout t "Traffic light is yellow at the moment. Car is waiting for the traffic light to change to green." crlf))))))

)

; JESS aplinkoje komentarus pasalinkite

;

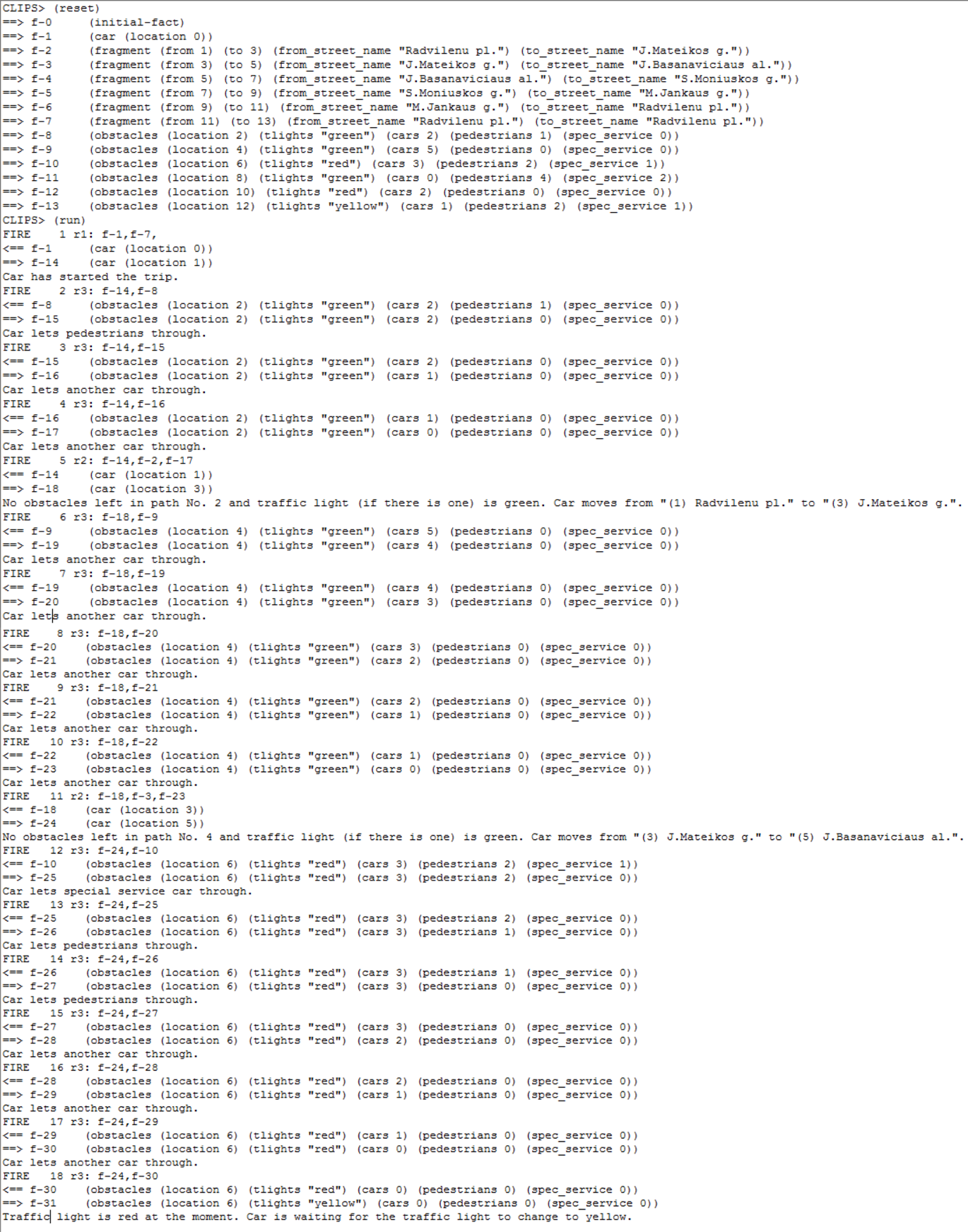
; (reset)

; (facts)

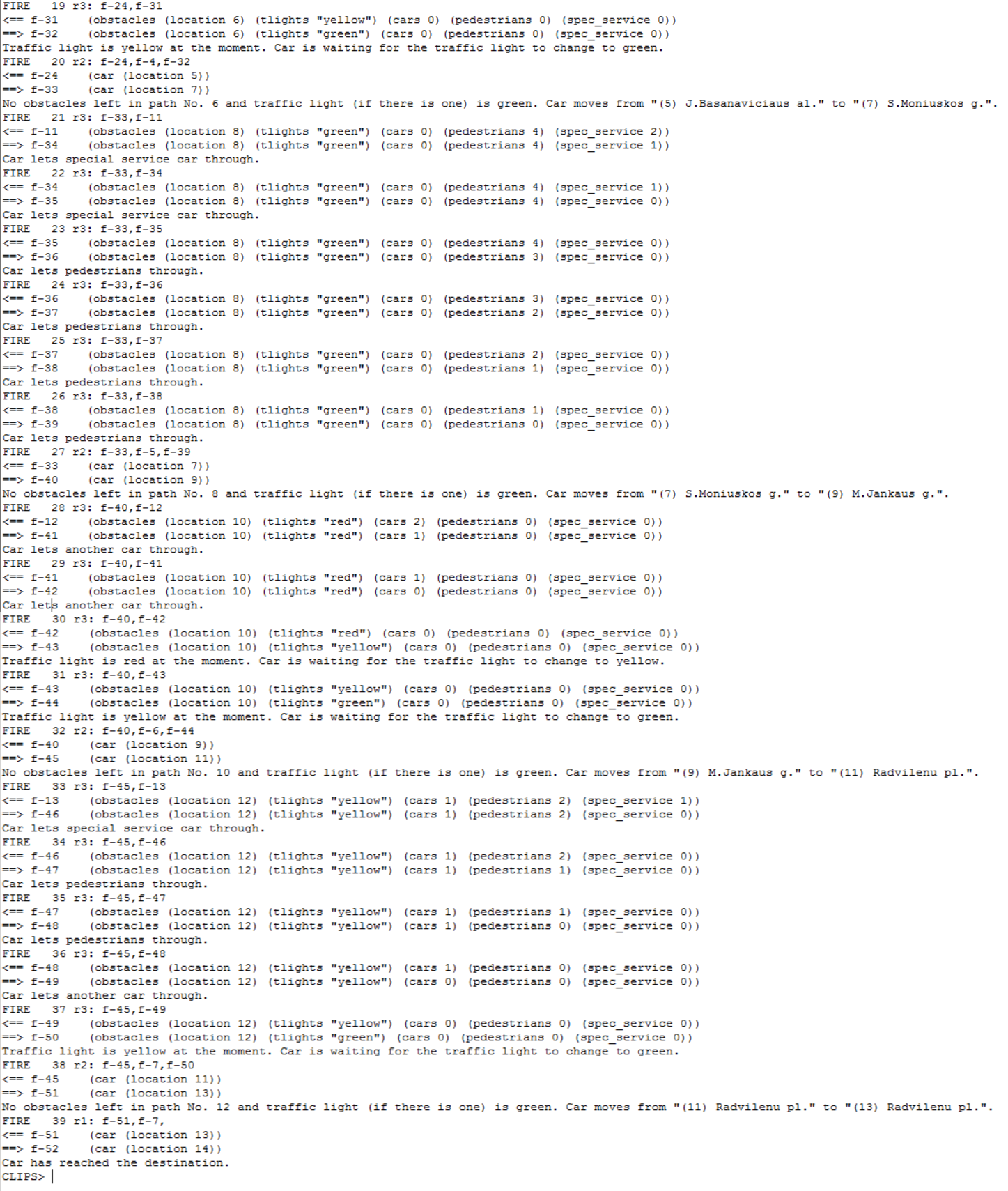
; (watch all)

; (run)

## Programos vykdymas



4 pav. Programos “Autonominis automobilis” vykdymas (FIRE 1-18)



5 pav. Programos “Autonominis automobilis” vykdymas (FIRE 19-39)

# Išvados

Laboratorinio darbo metu susipažinome su taisyklėmis grįsta programavimo kalba CLIPS bei vienu iš žinių ir jų loginiu išvedimu. Taip pat buvo išspręsti du uždaviniai, jų sprendimui buvo naudojamos tokios CLIPS priemonės bei aprašymo būdai, kaip šablonai, faktai, taisyklės, o pats uždavinio sprendimas buvo grįstas taisyklių ir faktų susiejimu realaus pasaulio tematikos uždaviniui spręsti.