

# Laboratorinis darbas 1

**Tikslai:** susipažinti su žinių vaizdavimo būdais,  
loginio išvedimo mechanizmu,  
žiniomis grįstos sistemos pagrindiniais komponentais.

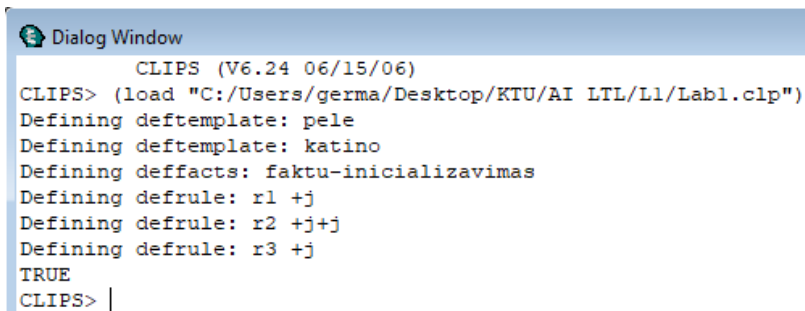
**Darbo eiga:**

1) [Lab1\(LT\).pptx](#)

2) [CLIPSWin](#)

Programos užkrovimas į sistemą CLIPS

File -> Load -> Lab1.clp



```
Dialog Window
CLIPS (V6.24 06/15/06)
CLIPS> (load "C:/Users/germa/Desktop/KTU/AI LTL/L1/Lab1.clp")
Defining deftemplate: pele
Defining deftemplate: katino
Defining deffacts: faktų-inicializavimas
Defining defrule: r1 +j
Defining defrule: r2 +j+j
Defining defrule: r3 +j
TRUE
CLIPS> |
```

TRUE pabaigoje pažymi, kad programa neturi sintaksės klaidų. Klaidos pranešimo iliustracija bei komandos, susijusios su jų taisyčių pateiktos priede.

Parodomi langai skirti darbinės atminties turiniui (facts) ir sąrašui su parengtomis vykdyti taisyklėmis (agenda) vaizduoti

Window -> Facts  
Window -> Agenda

Language vaizduojamos informacijos struktūra

**Facts**

(<fakto-id> <fakto-reikšmė>)

<fakto-id> taip pat gali būti suprantamas kaip fakto adresas.

**Agenda**

(<prioritetas> <taisyklės-id> <faktų adresai-reikalingi-taisyklei-įvykdyti>)

Kad dialogo lange matyti faktus naudojamus taisyklėms vykdyti bei jų naujas reikšmes, įjungiamo šiuos nustatymus

Execution -> Watch ...  
x Compilations  
x Facts  
x Rules

Faktams iš programos į darbinę atmintį perrašyti bei programos taisyklėms inicializuoti naudojama komanda:

(reset)

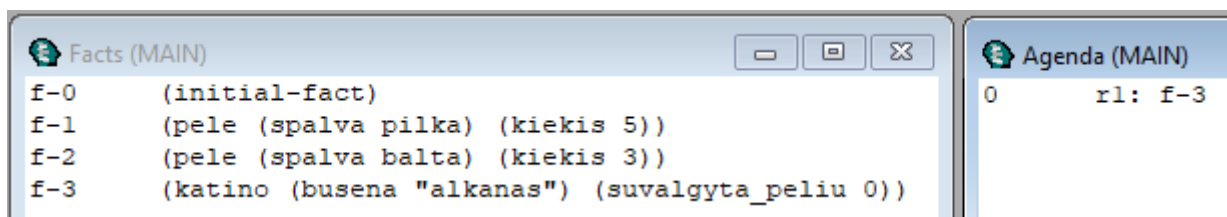
arba meniu punktas

### Execution -> Reset

Dialogo lange vaizduojami faktų adresai ir faktai, kurie įrašomi į darbinę atmintį:

```
CLIPS> (reset)
==> f-0      (initial-fact)
==> f-1      (pele (spalva pilka) (kiekis 5))
==> f-2      (pele (spalva balta) (kiekis 3))
==> f-3      (katino (busena "alkanas") (suvalgyta_peliu 0))
CLIPS> |
```

Darbinėje atmintyje vaizduojami iš programos konstrukcijos (deffacts (...)) įkelti faktai.



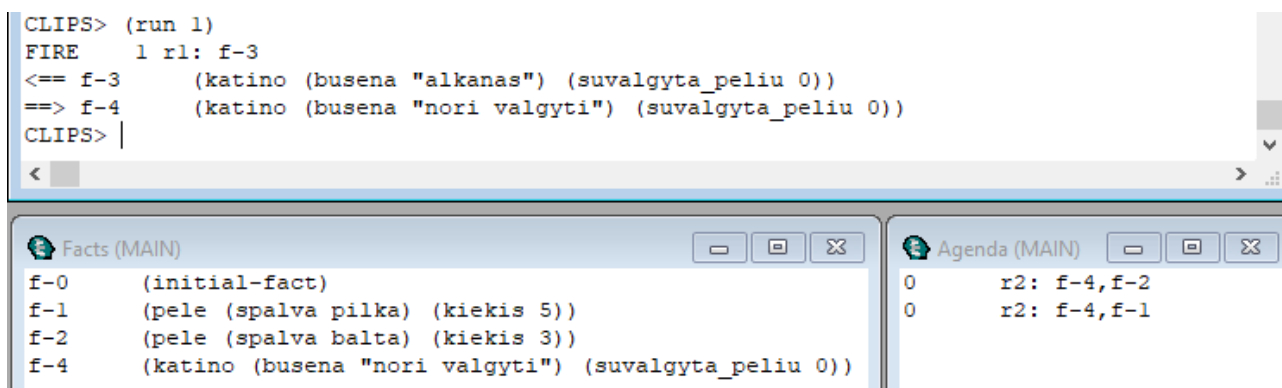
Agenda lange esantis įrašas parodo, kad taisyklė r1 yra parengta vykdyti ir jos vykdymui faktų bazėje yra faktas adresu f-3. Taisyklės r1 vykdymo prioritetą yra 0 (kai programos kode nenurodytas prioritetas – jo reikšmė yra 0).

Kad suprasti kaip veikia loginis išvedimas, programą reikia vykdyti žingsniais

### Run -> Step

ir stebėti darbinės atminties turinį bei parengtų vykdyti taisyklių sąrašą (facts ir agenda langai).

Po pirmo įvykdymo žingsnio dialogo lange matome, kad po taisyklės r1 įvykdymo, darbinė atmintis pasipildė faktu su atributo reikšme „nori valdyti“. Naujo fakto adresas – f-4. Pasikeitus faktams darbinėje atmintyje, pasikeitė ir interpretatoriaus parinktų parengtų vykdyti taisyklių sąrašas (žr. agenda langą).



### Atsakykite klausimus:

1. Kodėl agenda lange nėra taisyklės r1 ?
2. Kodėl agenda lange yra dvi r2 taisyklės ?
3. Kodėl vaizduojami du skirtingi adresų rinkiniai šalia r2 taisyklių identifikatorių ?
4. Programos kode parodykite r2 taisyklės sąlygas. Kaip jos siejamos su adresais, nurodytais agenda lange.

Po kelių pažingsninių vykdymų (Run -> Step) darbinė atmintis ir parengtų vykdymui taisyklių sąrašas bus tokie:

Facts (MAIN)	Agenda (MAIN)
f-0 (initial-fact)	10 r3: f-14
f-9 (pele (spalva balta) (kiekis 0))	0 r2: f-14, f-13
f-13 (pele (spalva pilka) (kiekis 3))	
f-14 (katino (busena "nori valgyti") (suvalgyta_peliu 5))	

#### Atsakykite klausimus:

1. Kodėl agenda lange nėra dviejų taisyklių r2 ?
2. Kodėl taisyklė r3 bus vykdoma prieš r2 ? Kurioje programos vietoje panaudota ši direktyva?

Po dar vieno programos vykdymo žingsnio (Run -> Step) darbinė atmintis ir parengtų vykdymui taisyklių sąrašas bus tokie:

Facts (MAIN)	Agenda (MAIN)
f-0 (initial-fact)	
f-9 (pele (spalva balta) (kiekis 0))	
f-13 (pele (spalva pilka) (kiekis 3))	
f-15 (katino (busena "storas") (suvalgyta_peliu 5))	

#### Atsakykite klausimus:

1. Kodėl agenda lange nėra nurodyta taisyklė r2 ? Jug ji turėjo būti vykdoma po r3 taisyklės įvykdymo (žr.agenda langą ankstesnėje iliustracijoje).
2. Kodėl nėra nei vienos taisyklės agenda lange? Atsakymą argumentuoti naudojant programos kodą.

Prieš užkraunant naują programos versiją visada būtina išvalyti darbinę atmintį:

Execution -> Clear CLIPS

3) Papildoma užduotis ir pagalbinių literatūra nurodyta [Lab1\(LT\).pptx](#)

#### Darbo ataskaitoje (el.dokumentas) pateikti

1. Titulinį lapą;
2. Visos programos kodą;
3. 2 ekranų kopijas, iliustruojančios papildomos užduoties reikalavimų įvykdymą;

Taip pat žinoti atsakymus į pateiktus apraše klausimus bei kaip veikia tiesioginis loginis išvedimas.

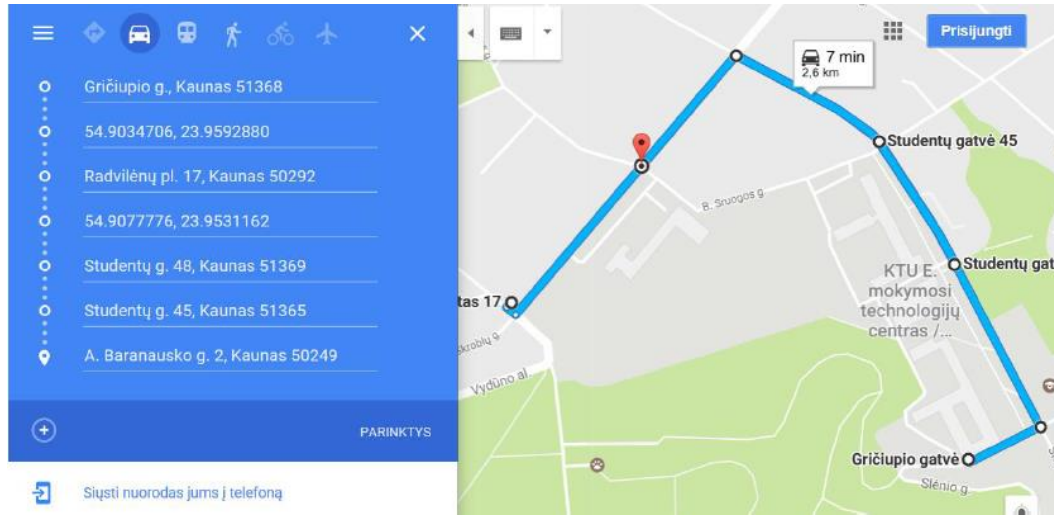
### Individuali užduotis

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 nesant kliūtims. Sistemoje iliustruoti autonominės mašinos judėjimą nurodytu maršrutu informuojant apie pravažiuojamas gatves ir kliūčių sumažėjimą važiuojamojo maršruto atkarpoje. Numatyti galimybę programoje papildyti norimą kiekį kliūčių bei pakoreguoti maršrutą.

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

Uždavinio realizacijai gali būti naudojami freimai:

(deftemplate fragment (slot from) (slot to)  
(slot from\_street\_name) (slot to\_street\_name))  
(deftemplate car (slot location))  
(deftemplate obstacles (slot location)  
(slot t\_lights) (slot cars) (slot pedestrians) (slot spec\_service))



#### Darbo ataskaitoje (prijungti prie lab.darbo ataskaitos) pateikti

1. Pasirinkto maršruto žemėlapi (kaip pavyzdyje);
2. Programos kodą;
3. Programos darbo ekrano kopiją (-jas), iliustruojančią užduoties įvykdymą.

**Atlikimo terminas – 3 studijų savaitė**

## PRIEDAS

### Klaidos pranešimo iliustracija

```
CLIPS> (clear)
CLIPS> (load "C:/Users/germa/Desktop/KTU/AI LTL/L1/Lab1.clp")
Defining deftemplate: pele
Defining deftemplate: katino
Defining deffacts: faktų-inicializavimas
Defining defrule: r1
[PRNTUTIL2] Syntax Error: Check appropriate syntax for binding patterns.

ERROR:
(defrule MAIN::r1 "Kai katinas alkanas, jis nori valgyti"
  ?fact-id (
Defining defrule: r2 +j+j
Defining defrule: r3 +j
FALSE
CLIPS> |
```

Klaidingas kodas	Teisingas kodas
(defrule r1 "Kai katinas alkanas, jis nori valgyti" ?fact-id (katino (busena ?busena)))	(defrule r1 "Kai katinas alkanas, jis nori valgyti" ?fact-id <- (katino (busena ?busena)))
	<- operatorius perduoda fakto (katino (...)) adresą kintamajam ?fact-id

Po programos pataisymo prieš ją užkraunant į CLIPS sistemą būtina išvaldyti darbinę atmintį naudojant komandą

**(clear)**

To nepadarius, rodomas pranešimas

```
ERROR:
(deftemplate MAIN::katino

[CSTRCPSR1] WARNING: Redefining deffacts: faktu-inicializavimas

[CSTRCPSR1] WARNING: Redefining defrule: r1 +j

[CSTRCPSR1] WARNING: Redefining defrule: r2 +j+j

[CSTRCPSR1] WARNING: Redefining defrule: r3 +j
FALSE
,
```