

**МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**

Кафедра математического обеспечения и применения ЭВМ

**ОТЧЕТ
по практической работе №2
по дисциплине «Представление знаний в системах искусственного
интеллекта»**

**Тема: Построение ЭС с использованием неупорядоченных фактов
(шаблонов) и различных типов условных элементов в антецедентах
правил**

Студент гр. 1310

Комаров Д.Е.

Преподаватель

Сучков А.И.

Санкт-Петербург
2025

Цель работы.

Целью выполнения практической работы освоение методов построение экспертной системы с использованием неупорядоченных фактов и различных типов условных элементов в антецедентах правил.

Основные теоретические положения.

Неупорядоченные факты представляют собой список взаимосвязанных именованных полей, называемых слотами. Наличие имен полей позволяет осуществлять доступ к полям по именам, в отличие от упорядоченных фактов, где поля специфицируются своим местоположением в факте. Существует два типа слотов: одиночные и мультислоты. Одиночный слот (или просто слот) содержит единственное поле, тогда как мультислот может содержать любое число полей.

Литеральное ограничение задает точное значение (константу) целого, вещественного, символьного или строкового типа, которое должно сопоставляться с полем. При работе с объектами литературное ограничение задает имя экземпляра. УЭ-образец с литературными ограничениями не содержит полей масок и переменных. Все ограничения литературного образца должны точно совпадать со всеми полями сопоставляемой сущности.

УЭ-проверка используется для оценки выражений в процессе сопоставления с образцом. УЭ-проверка удовлетворяется, если вызываемая в нем функция возвращает значение, отличное от FALSE, и не удовлетворяется в противном случае. Использование данного типа УЭ позволяет, в частности, проверять любые соотношения между значениями различных полей (слотов) фактов

Предикатное ограничение используется в тех случаях, когда необходимо ограничить поле, основываясь на истинности некоторого булевого выражения. Для этого используется предикатная функция, возвращающая символьное значение FALSE в случае неудачи и другое значение, в случае успеха. Функция вызывается в процессе сопоставления с образцом. Если она

возвращает значение FALSE, то ограничение не удовлетворяется, в противном случае – оно удовлетворяется. Предикатное ограничение задается с помощью символа “：“, за которым следует вызов предикатной функции.

Ограничение возвращаемым значением использует в качестве ограничения поля значение, возвращаемое некоторой функцией, которая вызывается непосредственно из условного элемента. Возвращаемое значение должно быть одного из примитивных типов. Это значение подставляется непосредственно в образец на позицию, из которой была вызвана функция, как если бы оно было литеральным ограничением. При этом функция вычисляется каждый раз, когда проверяется ограничение.

Постановка задачи.

В ходе выполнения практической работы необходимо:

1. Используя редактор CLIPS, сформировать базу данных, содержащую не менее десяти неупорядоченных фактов на основе предоставленного шаблона.

2. Составить в соответствии с вариантом задания, правила, реализующие описанные ниже функции, с использованием заданных типов условных элементов.

2.1. Используя только литеральные ограничения, составить правила для нахождения БД фактов, удовлетворяющих заданным условиям, и выдачи соответствующих сообщений.

2.2. Изменить сформированные в п. 2.1. правила путем добавления в антецедент новых условий и изменения выводимых сообщений. При реализации новых УЭ использовать УЭ-проверки.

2.3. Изменить сформированные в п. 2.2. правила путем добавления в антецеденты предикатных условных элементов для проверки типов значений слотов.

2.4. Изменить сформированные в п. 2.3. правила путем добавления в антецеденты условных элементов с ограничением по возвращаемому значению.

В команде, выполнившей данную практическую работу, было следующее разделение ролей:

- Будаев Геннадий – написание 3 и 4 правил, формирование отчёта;
- Комаров Дмитрий – формирование исходных фактов, написание 1 и 2 правила, формирование отчета;
- Шейнов Кирилл – написание 3 и 4 правил, формирование отчёта.

Выполнение работы.

1. Формирование исходных фактов.

Для выполнения практической работы сформируем неупорядоченные факты по шаблону, представленному в таблице 1.

Таблица 1 – Шаблон для неупорядоченных фактов

Поле	Тип	Значение
<i>name</i>	symbol	Имя студента
<i>age</i>	integer	Возраст
<i>year</i>	integer	Год обучения
<i>spec</i>	string	Специализация
<i>aver_mark</i>	float	Средний балл

Сформированные неупорядоченные факты представлены в таблице 2. Код сформированных фактов на языке CLIPS представлен в приложении А.

Таблица 2 – Неупорядоченные факты

<i>name</i>	<i>age</i>	<i>year</i>	<i>spec</i>	<i>aver_mark</i>
Karl	22	5	“soft”	5.0
Frederick	21	4	“ai”	4.5
Vladimir	20	4	“hard”	5.0
Iosif	19	2	“hard”	4.7

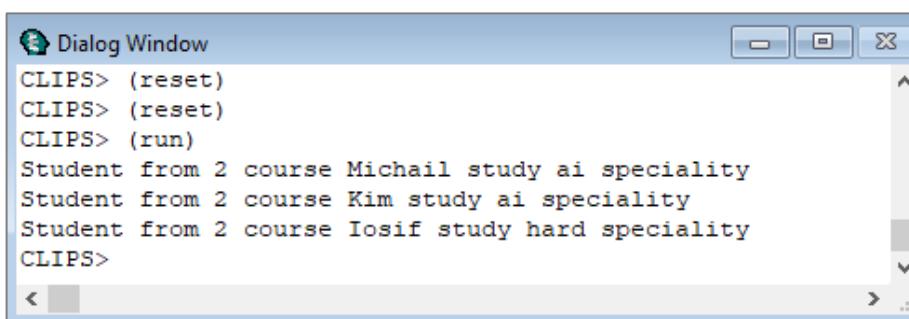
Продолжение таблицы 2

Yakov	20	3	“hard”	4.0
Felix	18	1	“soft”	4.2
Lev	22	4	“ai”	3.0
Mao	22	5	“hard”	3.7
Kim	19	2	“ai”	3.9
Ernst	18	1	“soft”	3.3
Michail	23	2	“ai”	4.8

2. Составление правил.

2.1 Литеральные ограничения.

Используя только литературные ограничения, было составлено правило, условием в антецеденте которого является то, что студент учится на 2 курсе, с выводом в консеквенте сообщения “Студент 2-го курса *<name>* учится по специальности *<spec>*”. Составление правила на языке CLIPS представлено в приложении А. Результат выполнения данного правила представлен на рисунке 1.



The screenshot shows the CLIPS Dialog Window. The input area contains the command "(run)". The output area displays the results of the rule execution:

```
CLIPS> (reset)
CLIPS> (reset)
CLIPS> (run)
Student from 2 course Michail study ai speciality
Student from 2 course Kim study ai speciality
Student from 2 course Iosif study hard speciality
CLIPS>
```

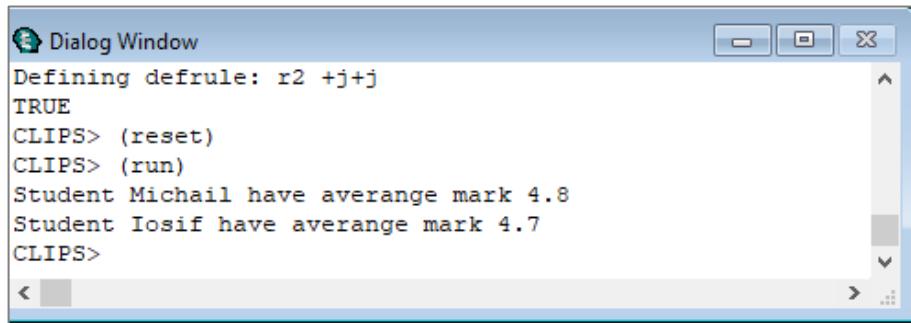
Рисунок 1 – Результат выполнения правила с литературными

2.2 Условные элементы проверки.

Было изменено правило, составленное в пункте 2.1, путем добавления в антецедент условного элемента проверки на то, чтобы средний балл был не ниже 4.5 и изменено сообщение в консеквенте на “Студент *<name>* имеет

средний балл `<aver_mark>`. Составление правило на языке CLIPS представлено в приложении А.

Результат выполнения данного правила представлен на рисунке 2.

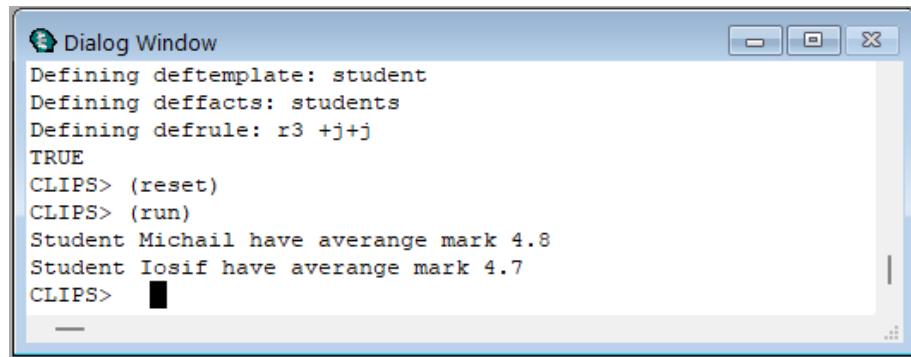


```
Dialog Window
Defining defrule: r2 +j+j
TRUE
CLIPS> (reset)
CLIPS> (run)
Student Michail have averange mark 4.8
Student Iosif have averange mark 4.7
CLIPS>
```

Рисунок 2 – Результат выполнения правила с условными элементами проверки

2.3 Предикатные условные элементы.

Было изменено правило, составленное в пункте 2.2, путем добавления в антецедент предикатного условия проверки типов слотов `year` и `aver_mark`. Составление правило на языке CLIPS представлено в приложении А. Результат выполнения данного правила представлен на рисунке 3.



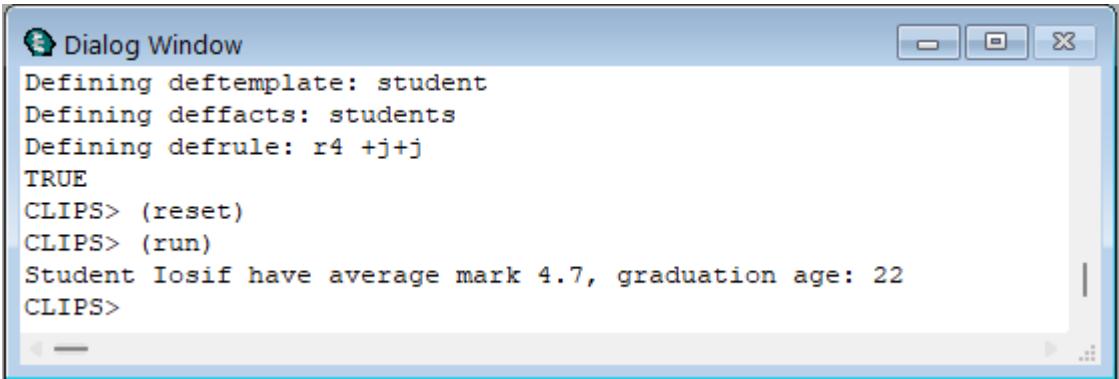
```
Dialog Window
Defining deftemplate: student
Defining deffacts: students
Defining defrule: r3 +j+j
TRUE
CLIPS> (reset)
CLIPS> (run)
Student Michail have average mark 4.8
Student Iosif have average mark 4.7
CLIPS> ■
```

Рисунок 3 – Результат выполнения правила с условными элементами проверки типов слотов

2.4 Условные элементы с ограничением по возвращаемому значению.

Было изменено правило, составленное в пункте 2.3, путем добавления в антецедент условного элемента с ограничением по возвращаемому значению на то, что студент оканчивает университет в возрасте не старше 24 лет.

Составление правила на языке CLIPS представлено в приложении А. Результат выполнения данного правила представлен на рисунке 4.



```
Dialog Window
Defining deftemplate: student
Defining deffacts: students
Defining defrule: r4 +j+j
TRUE
CLIPS> (reset)
CLIPS> (run)
Student Iosif have average mark 4.7, graduation age: 22
CLIPS>
```

Рисунок 4 – Результат выполнения правила с условными элементами с ограничением по возвращаемому значению

Выводы.

В ходе выполнения практической работы были основаны методы построения экспертной системы с использованием неупорядоченных фактов и различных типов условных элементов в антецедентах правил.

Была сформирована база исходных неупорядоченных фактов. Для работы с данными фактами были написаны несколько правил, использующих различные условные элементы, такие как: литеральные ограничения, условные элементы проверки, предикатные условные элементы, условные элементы с ограничением по возвращаемому значению.

Одной из небольших проблем была равномерность данных, сформированных в начале работы, что не всегда позволяло наглядно увидеть работу изменённого правила. Так, в ходе выполнения работы, были скорректированы некоторые начальные данные.

Код фактов и правил, написанный в ходе выполнения лабораторной работы представлен в приложении А.

ПРИЛОЖЕНИЕ А ИСХОДНЫЙ КОД

Код сформированных неупорядоченных фактов:

```
(deftemplate student
  (slot name) ; имя студента
  (slot age) ; возраст
  (slot year) ; год обучения (курс)
  (slot spec) ; специализация
  (slot aver_mark)) ; средний балл

(deffacts students
  (student (name Karl) (age 22) (year 5) (spec "soft") (aver_mark 5.0))
  (student (name Frederick) (age 21) (year 4) (spec "ai") (aver_mark 4.5))
  (student (name Vladimir) (age 20) (year 4) (spec "hard") (aver_mark 5.0))
  (student (name Iosif) (age 19) (year 2) (spec "hard") (aver_mark 4.7))
  (student (name Yakov) (age 20) (year 3) (spec "hard") (aver_mark 4.0))
  (student (name Felix) (age 18) (year 1) (spec "soft") (aver_mark 4.2))
  (student (name Lev) (age 22) (year 4) (spec "ai") (aver_mark 3.0))
  (student (name Mao) (age 22) (year 5) (spec "hard") (aver_mark 3.7))
  (student (name Kim) (age 19) (year 2) (spec "ai") (aver_mark 3.9))
  (student (name Ernst) (age 18) (year 1) (spec "soft") (aver_mark 3.3))
  (student (name Michail) (age 23) (year 2) (spec "ai") (aver_mark 4.8))
)
```

Код правила с литеральными ограничениями:

```
(defrule r1 (student (name ?n) (year 2) (spec ?sp))=>(printout t
  "Student from 2 course \" ?n\" study \" ?sp \" speciality" crlf))
```

Код правила с условными элементами проверки:

```
(defrule r2 (student (name ?n) (year 2) (aver_mark ?m)) (test (>= ?m
  4.5))=>(printout t "Student " ?n " have averange mark " ?m crlf))
```

Код правила с предикатными условными элементами:

```
(defrule r3 (student (name ?n) (year ?y&:(integerp ?y)) (aver_mark  
?m&:(floatp ?m))) (test (= ?y 2)) (test (>= ?m 4.5)) => (printout t  
"Student " ?n " have averange mark " ?m crlf))
```

Код правила с условными элементами с ограничением по возвращаемому значению:

```
(defrule r4 (student (name ?n) (year ?y&:(integerp ?y)) (age  
?a&:(integerp ?a)&:<= (+ ?a (- 5 ?y)) 24) (aver_mark ?m&:(floatp  
?m))) (test (= ?y 2)) (test (>= ?m 4.5)) => (printout t "Student "  
?n " graguate earlier 24 age" crlf))
```