



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет имени
Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

Лабораторная работа № 2 по дисциплине «Функциональные и логические языки программирования»

Тема Пролог. Списки.

Студент Одинцов Е.В.

Группа ИУ7-53БВ

Преподаватели Строганов Ю.В.

Москва, 2024

Содержание

ВВЕДЕНИЕ	4
1 Аналитическая часть	5
2 Конструкторская часть	6
2.1 ER диаграмма	6
3 Технологическая часть	7
3.1 Используемые технологии	7
3.2 База знаний	7
3.3 Запросы к базе знаний	7
3.4 Работа с числами и термами	8
ЗАКЛЮЧЕНИЕ	11
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	12

ВВЕДЕНИЕ

Целью данной работы является разработка базы знаний на языке Prolog, содержащей информацию о КБЖУ продуктов. В рамках работы ставятся следующие задачи:

- 1) Сформировать базу знаний, содержащую информацию о калорийности, содержании белков, жиров и углеводов (КБЖУ) различных продуктов.
- 2) Создать программу на Prolog, которая взаимодействует с данной базой знаний, и реализовать правила, использующие предикат `findall`, для поиска объектов, удовлетворяющих заданным ограничениям. Необходимо разработать не менее трёх правил, каждое из которых должно включать от одного до трёх ограничений.
- 3) Реализовать следующие алгоритмы:
 - нахождение наибольшего числа из двух чисел, а также из трёх чисел (с и без использования отсечений);
 - нахождение наибольшего элемента в списке целых чисел (двумя способами: самописное правило и использование стандартного предиката);
 - нахождение наибольшего элемента в списке термов (двумя способами: самописное правило и использование стандартного предиката);
 - нахождение наибольшего элемента в списке списков чисел (двумя способами: самописное правило и использование стандартного предиката);
 - нахождение наибольшего элемента в списке списков термов (двумя способами: самописное правило и использование стандартного предиката);
 - нахождение среднего арифметического в списке целых чисел;
 - нахождение общих элементов в двух списках (двумя способами).

1 Аналитическая часть

Анализируется связь различных видов продуктов питания, способов их приготовления и их КБЖУ.

2 Конструкторская часть

2.1 ER диаграмма

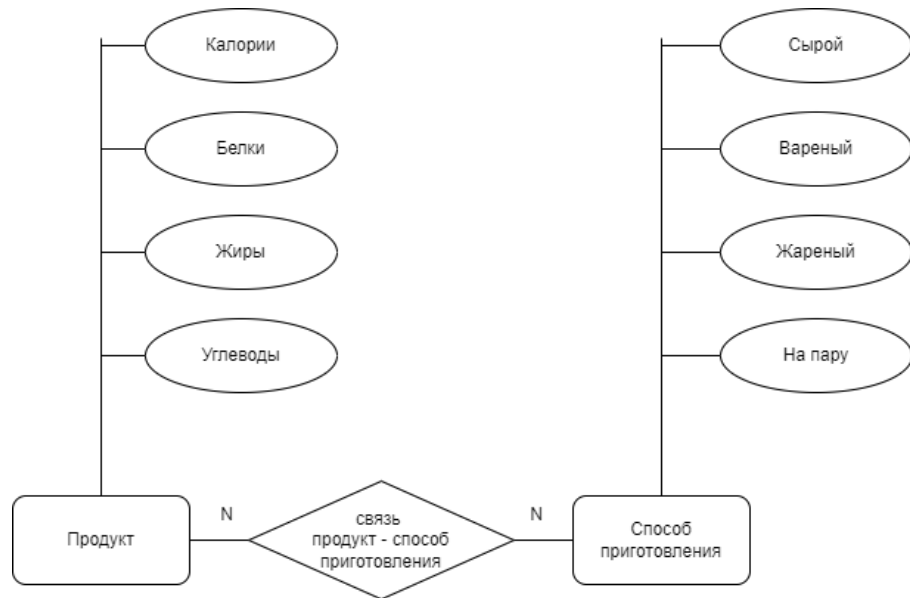


Рисунок 2.1 — ER диаграмма для базы данных на Prolog

3 Технологическая часть

3.1 Используемые технологии

— Язык: Prolog

— IDE: TexStudio

3.2 База знаний

% Яйцо

продукт(яйцо, сырое, 155, 13, 11, 1.1).

продукт(яйцо, вареное, 160, 12.6, 11.2, 1.1).

продукт(яйцо, жареное, 196, 13, 15, 1.1).

% Курица

продукт(курица, сырое, 110, 21, 3, 0).

продукт(курица, вареное, 120, 22, 4, 0).

продукт(курица, жареное, 175, 20, 8, 0).

продукт(курица, на_пару, 150, 21, 5, 0).

% Картофель

продукт(картофель, сырой, 77, 2, 0.1, 17).

продукт(картофель, вареное, 86, 1.7, 0.1, 20).

продукт(картофель, жареное, 150, 2, 10, 25).

продукт(картофель, на_пару, 93, 2.5, 0.1, 21).

% Рис

продукт(рис, сырой, 360, 7.5, 0.6, 78).

продукт(рис, вареное, 130, 2.7, 0.3, 28).

продукт(рис, жареное, 200, 4, 6, 35).

% Сыр

продукт(сыр, сырой, 402, 25, 33, 1.3).

продукт(сыр, жареное, 450, 25, 37, 2.5).

3.3 Запросы к базе знаний

найти_низкокалорийные(Продукты) :-

findall(Название, (продукт(Название, _, Калории, _, _, _), Калории < 100), Продукты).

```

найти_высокобелковые_веганские(Продукты) :-
findall(Название, (продукт(Название, _, _, Белки, _, _),
(Название = картофель ; Название = рис),
Белки > 2), Продукты).

```

```

найти_жирные_низкоуглеводные(Продукты) :-
findall(Название, (продукт(Название, _, _, _, Жиры, Углеводы), Жиры > 5, Углеводы

```

3.4 Работа с числами и термами

```

% Нахождение наибольшего из двух чисел
% без отсечений:
max2(A, B, R) :- (A >= B -> R = A ; R = B).
% с отсечениями
max2_otsek(A, B, A) :- A >= B, !.
max2_otsek(_, B, B).

```

```

% Нахождение наибольшего из трёх чисел
% Без отсечений:
max3(A, B, C, R) :-
(A >= B, A >= C -> R = A;
B >= C -> R = B;
R = C).

```

```

% С отсечениями:
max3_otsek(A, B, C, R) :-
A >= B, A >= C, !, R = A.
max3_otsek(_, B, C, R) :-
B >= C, !, R = B.
max3_otsek(_, _, C, R) :-
R = C.

```

```

% Нахождение наибольшего элемента в списке из целых чисел
% Самописное правило:
макс_список([X], X).
макс_список([H|T], Max) :-
макс_список(T, Tmp),
(H > Tmp -> Max = H ; Max = Tmp).

```

```

% Стандартное правило (max_list/2)
макс_список_стандартный(Список, Мах) :- max_list(Список, Мах).

% Нахождение наибольшего элемента в списке из термов
% Самописное правило:
макс_термы([Терм], Терм).
макс_термы([H|T], Мах) :-
макс_термы(T, Тmp),
(H @> Тmp -> Мах = H ; Мах = Тmp).

% Стандартное правило (max_list/2)
макс_термы_стандартный(Термы, Мах) :- max_list(Термы, Мах).

% Нахождение наибольшего элемента в списке из списков целых чисел
% Самописное правило:
макс_списков([Список], Макс) :- макс_список(Список, Макс).
макс_списков([Список|Остальные], Макс) :-
макс_список(Список, Макс1),
макс_списков(Остальные, Макс2),
(Макс1 > Макс2 -> Макс = Макс1 ; Макс = Макс2).

% Стандартное правило
макс_списков_стандартный(Списки, Макс) :-
findall(МаксВнутри, (member(Список, Списки), max_list(Список, МаксВнутри)), ВсеМакс)
max_list(ВсеМаксимумы, Макс).

% Нахождение наибольшего элемента в списке из списков термов
% Самописное правило
макс_списков_термы([Список], Макс) :- макс_термы(Список, Макс).
макс_списков_термы([Список|Остальные], Макс) :-
макс_термы(Список, Макс1),
макс_списков_термы(Остальные, Макс2),
(Макс1 @> Макс2 -> Макс = Макс1 ; Макс = Макс2).

% Стандартное правило
макс_списков_термы_стандартный(Списки, Макс) :-
findall(МаксВнутри, (member(Список, Списки), max_list(Список, МаксВнутри)), ВсеМакс)
max_list(ВсеМаксимумы, Макс).

```



```

% Нахождение среднего арифметического в списке из целых чисел
среднее_арифметическое(Список, Среднее) :-
sum_list(Список, Сумма),
length(Список, Длина),
Длина > 0, % защита от деления на 0
Среднее is Сумма / Длина.

% Нахождение общих элементов в двух списках
% Самописное правило
общие_элементы([], _, []).
общие_элементы([Н|Т], Список2, [Н|Общие]) :-
member(Н, Список2),
общие_элементы(Т, Список2, Общие).
общие_элементы([_|Т], Список2, Общие) :-
общие_элементы(Т, Список2, Общие).

% Стандартное правило
общие_элементы_стандартный(Список1, Список2, Общие) :-
intersection(Список1, Список2, Общие).

```

ЗАКЛЮЧЕНИЕ

В ходе работы была выполнена разработка базы знаний на языке Prolog, включающей информацию о различных продуктах с указанием их калорийности, содержания белков, жиров и углеводов. Также были разработаны запросы для получения:

- низкокалорийных продуктов,
- продуктов с высоким содержанием белков и являющихся веганскими,
- жирных продуктов с низким содержанием углеводов.

Были реализованы методы поиска максимальных значений в списках чисел и термов, как с использованием стандартных встроенных предикатов, так и с помощью собственных решений. Эти методы оказались полезными для поиска наибольших элементов в различных наборах данных.

Таким образом, все поставленные задачи были успешно выполнены, а база знаний протестирована с использованием различных запросов и правил.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. SWI-Prolog. Доступно на: <https://www.swi-prolog.org/>. [Дата обращения: сентябрь 2024].
2. Примеры использования библиотеки CLPFD. Доступно на: <https://www.youtube.com/watch?v=sHo6-hk21L8>. [Дата обращения: сентябрь 2024].