|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **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 № 16**

|  |  |
| --- | --- |
| **Студент** Козлов М. А.  **Группа** ИУ7-65Б  **Оценка (баллы) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **Преподаватель** Толпинская Н. Б., Строганов Ю. В. |  |

Москва.

2021 г.

**Задание.**

**Создать базу знаний: «ПРЕДКИ»**, позволяющую **наиболее эффективным** способом (за меньшее количество шагов, что обеспечивается меньшим количеством предложений БЗ - правил), используя разные варианты (примеры) **одного вопроса**, определить (указать: какой вопрос для какого варианта):

1. по имени субъекта определить всех его бабушек (предки 2-го колена),
2. по имени субъекта определить всех его дедушек (предки 2-го колена),
3. по имени субъекта определить всех его бабушек и дедушек (предки 2-го колена),
4. по имени субъекта определить его бабушку по материнской линии (предки 2-го колена),
5. по имени субъекта определить его бабушку и дедушку по материнской линии (предки 2-го колена).

Минимизировать количество правил и количество вариантов вопросов. Использовать **конъюнктивные правила и простой вопрос.**

**Для одного** из вариантов **ВОПРОСА** и конкретной БЗ **составить таблицу**, отражающую конкретный порядок работы системы.

|  |
| --- |
| family(GrandmotherMomline,  GrandfatherMomline,  GrandmotherDadline,  GrandfatherDadline,  Mother, Father, Child):-  parents(Mother, Father, Child),  parents(GrandmotherMomline, GrandfatherMomline, Mother),  parents(GrandmotherDadline, GrandfatherDadline, Father).  famaly(1, Child, Mother, Father) :- parents(Mother, Father, Child), !.  famaly(N, Child, Mother, Father) :- parents(Mother1, \_, Child),  N2 is N - 1,  famaly(N2, Mother1, Mother, Father).  famaly(N, Child, Mother, Father) :- parents(\_, Father1, Child),  N2 is N - 1,  famaly(N2, Father1, Mother, Father).  parents("Lena", "Vasya", "Natasha").  parents("Ann", "Egor", "Lena").  parents("Eva", "Dmitry", "Vasya").  parents("Krina", "Sasha", "Ann").  parents("Dasha", "Petya", "Eva").  parents("Tanya", "Kostya", "Egor").  parents("Olga", "Oleg", "Dmitry"). |

1. по имени субъекта определить всех его бабушек (предки 2-го колена)

family(Grandmom\_MlineQ, \_, Grandmom\_FlineQ, \_, \_, \_, " Natasha ").

**Grandmom\_MlineQ=Ann, Grandmom\_FlineQ=Eva**

**1 Solution**

2. по имени субъекта определить всех его дедушек (предки 2-го колена)

family(\_, Grandfa\_MlineQ, \_, Grandfa\_FlineQ, \_, \_, " Natasha ").

**Grandfa\_MlineQ=Egor, Grandfa\_FlineQ=Dmitry**

**1 Solution**

3. по имени субъекта определить всех его бабушек и дедушек (предки 2-го колена)

family(Grandmom\_MlineQ, Grandfa\_MlineQ, Grandmom\_FlineQ, Grandfa\_FlineQ, \_, \_, " Natasha ").

**Grandmom\_MlineQ=Ann, Grandfa\_MlineQ=Egor, Grandmom\_FlineQ=Eva, Grandfa\_FlineQ=Dmitry**

**1 Solution**

4. по имени субъекта определить его бабушку по материнской линии (предки 2-го колена)

family(Grandmom\_MlineQ, \_, \_, \_, \_, \_, " Natasha ").

**Grandmom\_MlineQ=Ann**

**1 Solution**

5. по имени субъекта определить его бабушку и дедушку по материнской линии (предки 2-го колена)

family(Grandmom\_MlineQ, Grandfa\_MlineQ, \_, \_, \_, \_, "Natasha ").

**Grandmom\_MlineQ=Ann, Grandfa\_MlineQ=Egor**

**1 Solution**

Вопрос family(Grandmom\_MlineQ, \_, \_, \_, \_, \_, "Natasha").

|  |  |  |  |
| --- | --- | --- | --- |
| № шага | Состояние резольвенты, и вывод: дальнейшие действия (почему?) | Для каких термов запускается алгоритм унификации: Т1=Т2 и каков результат (и подстановка) | Дальнейшие действия: прямой ход или откат (почему и к чему приводит?) |
| 1 | Состояние резольвенты  family(Grandmom\_MlineQ, \_, \_, \_, \_, \_, " Natasha")  Первое состояние резольвенты - вопрос. Далее запускается алгоритм унификации. | family(Grandmom\_MlineQ, \_, \_, \_, \_, \_, " Natasha") = family(Grandmom\_Mline, Grandfa\_Mline, Grandmom\_Fline, Grandfa\_Fline, Mother, Father, Child)  Унификация успешна.  Θ = {Grandmom\_MlineQ = Grandmom\_Mline, " Natasha" = Child} | Образование новой резольвенты:  1)В текущей резольвенте выполняется редукция для 1 цели.  2)К полученной конъюнкции целей применяется подстановка, полученная как наибольший общий унификатор цели и заголовка сопоставимого с ней правила. |
| 2 | Состояние резольвенты  parents(Mother, Father, " Natasha")  parents(Grandmom\_MlineQ, Grandfa\_Mline, Mother)  parents(Grandmom\_Fline, Grandfa\_Fline, Father) | parents(Mother, Father, " Natasha ") = family(Grandmom\_Mline, Grandfa\_Mline, Grandmom\_Fline, Grandfa\_Fline, Mother, Father, Child)  Разные главные функторы.  Унификация не успешна. | переход к следующему предложению. |
| 3 | Состояние резольвенты  parents(Mother, Father," Natasha")  parents(Grandmom\_MlineQ, Grandfa\_Mline, Mother)  parents(Grandmom\_Fline, Grandfa\_Fline, Father) | parents(Mother, Father, " Natasha")  = parents("Lena", "Vasya", " Natasha")  Унификация успешна.  Θ1 **= {**Mother = " Lena", Father = " Vasya"**}** | Образование новой резольвенты:  1)В текущей резольвенте выполняется редукция для 1 цели.  2)К полученной конъюнкции целей применяется подстановка, полученная как наибольший общий унификатор цели и заголовка сопоставимого с ней правила. |
| 4 | Состояние резольвенты  parents(Grandmom\_MlineQ, Grandfa\_Mline," Lena")  parents(Grandmom\_Fline, Grandfa\_Fline," Vasya") | parents(Grandmom\_MlineQ, Grandfa\_Mline," Lena") = family(Grandmom\_Mline, Grandfa\_Mline, Grandmom\_Fline, Grandfa\_Fline, Mother, Father, Child)  Разные главные функторы.  Унификация не успешна. | переход к следующему предложению. |
| 5 | Состояние резольвенты  parents(Grandmom\_MlineQ, Grandfa\_Mline," Lena")  parents(Grandmom\_Fline, Grandfa\_Fline," Vasya") | parents(Grandmom\_MlineQ, Grandfa\_Mline," Lena") = parents("Lena", " Vasya", " Natasha")  Разные константы типа name.  Унификация не успешна. | переход к следующему предложению. |
| 6 | Состояние резольвенты  parents(Grandmom\_MlineQ, Grandfa\_Mline, "Lena")  parents(Grandmom\_Fline, Grandfa\_Fline," Vasya") | parents(Grandmom\_MlineQ, Grandfa\_Mline," Lena ") = parents("Ann", "Egor", " Lena ")  Унификация успешна.  Θ2 = {Grandmom\_MlineQ = "Ann", Grandfa\_Mline = " Egor "} | Образование новой резольвенты:  1)В текущей резольвенте выполняется редукция для 1 цели.  2)К полученной конъюнкции целей применяется подстановка, полученная как наибольший общий унификатор цели и заголовка сопоставимого с ней правила. |
| 7 | Состояние резольвенты  parents(Grandmom\_Fline, Grandfa\_Fline," Vasya") | parents(Grandmom\_Fline, Grandfa\_Fline," Vasya ") = family(Grandmom\_Mline, Grandfa\_Mline, Grandmom\_Fline, Grandfa\_Fline, Mother, Father, Child)  Разные главные функторы.  Унификация не успешна. | Откат, переход к следующему предложению. |
| 8 | Состояние резольвенты  parents(Grandmom\_Fline, Grandfa\_Fline," Vasya") | parents(Grandmom\_Fline, Grandfa\_Fline,"Vasya") =  parents("Lena", " Egor", "Natasha")  Разные константы типа name.  Унификация не успешна. | Откат, переход к следующему предложению. |
| ... | ... | ... | ... |
| 10 | Состояние резольвенты  parents(Grandmom\_Fline, Grandfa\_Fline," Vasya") | parents(Grandmom\_Fline, Grandfa\_Fline," Vasya ") = parents("Eva", "Dmitry", " Vasya ")  Унификация успешна.  Θ3 = {Grandmom\_Fline = "Eva", Grandfa\_Fline = "Dmitry"} | Образование новой резольвенты:  1)В текущей резольвенте выполняется редукция для 1 цели.  2)К полученной конъюнкции целей применяется подстановка, полученная как наибольший общий унификатор цели и заголовка сопоставимого с ней правила. |
| 11 | Состояние резольвенты:  Резольвента пуста |  | Найден ответ  Grandmom\_MlineQ = "Ann"  Поиск альтернативного решения. Откат к предыдущему состоянию резольвенты.  Освобождение переменных Grandmom\_Fline, Grandfa\_Fline |
| 12 | Состояние резольвенты  parents(Grandmom\_Fline, Grandfa\_Fline," Vasya") |  | Все знания в базе знаний просмотрены.  Поиск альтернативного решения. Откат к предыдущему состоянию резольвенты.  Освобождение переменных  Grandmom\_MlineQ, Grandfa\_Mline |
| 13 | Состояние резольвенты  parents(Grandmom\_MlineQ, Grandfa\_Mline, "Lena")  parents(Grandmom\_Fline, Grandfa\_Fline," Vasya") | parents(Grandmom\_MlineQ, Grandfa\_Mline," Lena")  parents("Eva", "Dmitry", " Vasya ")  Разные константы типа name.  Унификация не успешна. | Все знания в базе знаний просмотрены.  Поиск альтернативного решения. Откат к предыдущему состоянию резольвенты.  Освобождение переменных Mother, Father |
| 14 | Состояние резольвенты  parents(Mother, Father, " Natasha")  parents(Grandmom\_MlineQ, Grandfa\_Mline, Mother)  parents(Grandmom\_Fline, Grandfa\_Fline, Father) | parents(Mother, Father, " Natasha") = parents("Ann", "Egor", " Lena")  Разные константы типа name.  Унификация не успешна. | Откат, переход к следующему предложению. |
| 15 | Состояние резольвенты  parents(Mother, Father, " Natasha")  parents(Grandmom\_MlineQ, Grandfa\_Mline, Mother)  parents(Grandmom\_Fline, Grandfa\_Fline, Father) | parents(Mother, Father, " Natasha") =  parents("Eva", "Dmitry", " Vasya ")  Разные константы типа name.  Унификация не успешна. | Все знания в базе знаний просмотрены.  Поиск альтернативного решения. Откат к предыдущему состоянию резольвенты.  Освобождение переменных Grandmom\_MlineQ, Child |
| 16 | Состояние резольвенты:  family(Grandmom\_MlineQ, \_, \_, \_, \_, \_, "Vera") | family(Grandmom\_MlineQ, \_, \_, \_, \_, \_, " Natasha") = parents("Lena", "Vasya", " Natasha")  Разные главные функторы.  Унификация не успешна. | Откат, переход к следующему предложению. |
| ... | ... | ... | ... |
| 18 | Состояние резольвенты:  family(Grandmom\_MlineQ, \_, \_, \_, \_, \_, "Vera") | family(Grandmom\_MlineQ, \_, \_, \_, \_, \_, " Natasha") = ("Eva", "Dmitry", " Vasya ")  Разные главные функторы.  Унификация не успешна. | Все знания в базе знаний просмотрены.  Завершение работы программы. |

|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **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 № 17**

|  |  |
| --- | --- |
| **Студент** Козлов М. А.  **Группа** ИУ7-65Б  **Оценка (баллы) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **Преподаватель** Толпинская Н. Б., Строганов Ю. В. |  |

Москва.

2021 г.

**Задание.**

В одной программе написать правила, позволяющие найти

1. Максимум из двух чисел
   1. без использования отсечения,
   2. с использованием отсечения;
2. Максимум из трех чисел
   1. без использования отсечения,
   2. с использованием отсечения;

Убедиться в правильности результатов.

Для каждого случая пункта 2 обосновать необходимость всех условий тела.

Для одного из вариантов ВОПРОСА и каждого варианта задания 2 составить таблицу, отражающую конкретный порядок работы системы.

|  |
| --- |
| max(X, Y, X) :- X >= Y. max(X, Y, Y) :- X < Y.  max\_cat(X, Y, X) :- X >= Y, !. max\_cat(\_, Y, Y).  max(X, Y, Z, X) :- X >= Y, X >= Z. max(X, Y, Z, Y) :- Y >= X, Y >= Z. max(X, Y, Z, Z) :- Z >= X, Z >= Y.  max\_cat(X, Y, Z, X) :- X >= Y, X >= Z, !. max\_cat(\_, Y, Z, Y) :- Y >= Z, !. max\_cat(\_, \_, Z, Z). |

**Обоснование необходимости каждого условия:**

**Без использования отсечения**

* max(X, Y, Z, X) :- X >= Y, X >= Z. - Нужно проверить является ли Х максимальным.
* max(X, Y, Z, Y) :- Y >= X, Y >= Z. - Нужно проверить, истинность условия Y >= X, т. к. если предыдущий шаг выполнен успешно, программа продолжить искать альтернативные решения.
* max(X, Y, Z, Z) :- Z >= X, Z >= Y. Аналогичные рассуждения.

**С отсечением**

* max\_cat(X, Y, Z, X) :- X >= Y, X >= Z, !.- Нужно проверить является ли Х максимальным. Если да, то отсекаем.
* max\_cat(\_, Y, Z, Y) :- Y >= Z, !. - Уже проверено, что Х не максимальный, остается только проверить, является ли Y максимальным.
* max\_cat(\_, \_, Z, Z). - Х и Y не максимальные - ответ Z.

|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **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 № 18**

|  |  |
| --- | --- |
| **Студент** Козлов М. А.  **Группа** ИУ7-65Б  **Оценка (баллы) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **Преподаватель** Толпинская Н. Б., Строганов Ю. В. |  |

Москва.

2021 г.

**Задание:**

**Используя хвостовую рекурсию, разработать программу, позволяющую найти:**

1. n!,
2. n-е число Фибоначчи.

Убедиться в правильности результатов.

Для одного из вариантов вопроса и каждого задания составить таблицу, отражающую конкретный порядок работы системы:

|  |
| --- |
| factorial(N, Result):- fact(N, Result, 1).  fact(1, Result, Result) :- !.  fact(N, Result, TmpRes):-  NewRes = TmpRes \* N,  N\_1 = N - 1,  fact(N\_1, Result, NewRes).  fibonacci(N, Result):- fib(N, Result, 1, 0).  fib(1, Result, Result, \_):- !.  fib(N, Result, X1, X2):-  Sum = X1 + X2,  N\_1 = N - 1,  fib(N\_1, Result, Sum, X1). |

**Примеры работы программы:**

**Поиск факториала числа:**

factorial(3, Res).

**Res=6**

**1 Solution**

**Поиск n-ого числа Фибоначчи:**

fibonacci(3, Res).

**Res=2**

**1 Solution**

**Вопрос factorial(3, Res).**

|  |  |  |  |
| --- | --- | --- | --- |
| № шага | Состояние резольвенты, и вывод: дальнейшие действия (почему?) | Для каких термов запускается алгоритм унификации: Т1=Т2 и каков результат (и подстановка) | Дальнейшие действия: прямой ход или откат (почему и к чему приводит?) |
| 1 | Состояние резольвенты:  factorial(3, Res). | factorial(3, Res) = factorial(N, Result)  Унификация успешна.  **Θ = {3 = N, Res = Result }** | Прямой ход  Смена состояния резольвенты |
| 2 | Состояние резольвенты:  fact(3, Res, 1). | fact(3, Res, 1). = factorial(N, Result)  Разные главные функторы.  Унификация не успешна. | Откат, переход к следующему предложению. |
| 3 | Состояние резольвенты:  fact(3, Res, 1). | fact(3, Res, 1). = fact(1, Result, Result)  Разные числовые константы (3 ≠ 1).  Унификация не успешна. | Прямой ход  Смена состояния резольвенты |
| 4 | Состояние резольвенты:  fact(3, Res, 1). | fact(3, Res, 1). = fact(N, Result, TmpRes)  Унификация успешна.  Θ1 = {3 = N, Res = Result, 1 = TmpRes}  **Θ = {3 = N, Res = Result }** | Прямой ход  Смена состояния резольвенты |
| 5 | Состояние резольвенты:  NewRes = TmpRes \* 3,  N\_1 = 3 - 1,  fact(N\_1, Res, NewRes). | NewRes = TmpRes \* 3 | Значение утверждения true  NewrRes = 3,  Прямой ход  Смена состояния резольвенты |
| 6 | Состояние резольвенты:  N\_1 = 3 - 1,  fact(N\_1, Res, 3). | N\_1 = 3 - 1 | Значение утверждения true  N\_1 = 2,  Прямой ход  Смена состояния резольвенты |
| 7 | Состояние резольвенты:  fact(2, Res, 3). | fact(2, Res, 3). = factorial(N, Result)  Разные главные функторы.  Унификация не успешна. | Откат, переход к следующему предложению. |
| … | … | … | … |
| 8 | Состояние резольвенты:  fact(2, Res, 3). | fact(2, Res, 3). = fact(N, Result, TmpRes)  Унификация успешна.  Θ2 = {2 = N, Res = Result, 3 = TmpRes}  **Θ = {2 = N, Res = Result }** | Прямой ход  Смена состояния резольвенты |
| 9 | Состояние резольвенты:  NewRes = 3 \* 2,  N\_1 = 2 - 1,  fact(N\_1, Res, NewRes). | NewRes = 3 \* 2 | Значение утверждения true  NewrRes = 6,  Прямой ход  Смена состояния резольвенты |
| 10 | Состояние резольвенты:  N\_1 = 2 - 1,  fact(N\_1, Res, 3). | N\_1 = 2 - 1 | Значение утверждения true  N\_1 = 1,  Прямой ход  Смена состояния резольвенты |
| 11 | Состояние резольвенты:  fact(1, Res, 3). | fact(1, Res, 6). = factorial(1, Result)  Разные главные функторы.  Унификация не успешна. | Откат, переход к следующему предложению. |
| … | … | … | … |
| 12 | Состояние резольвенты:  fact(1, Res, 6). | fact(1, Res, 6). = fact(1, Result, Result)    Унификация успешна.  Θ3 = {1 = 1, Res = 6, 6 = Result}  **Θ = {1 = N, Res = 6 }** | Прямой ход  Смена состояния резольвенты |
| 13 | Состояние резольвенты:  ! | ! - указывает системе отменить поиск альтернативных решений для целей до него | Прямой ход  Смена состояния резольвенты |
| 14 | Состояние резольвенты:  Резольвента пуста. |  | **Вывод Res = 6**  Завершение работы программы. |

**Вопрос fibonacci(3, Res).**

|  |  |  |  |
| --- | --- | --- | --- |
| № шага | Состояние резольвенты, и вывод: дальнейшие действия (почему?) | Для каких термов запускается алгоритм унификации: Т1=Т2 и каков результат (и подстановка) | Дальнейшие действия: прямой ход или откат (почему и к чему приводит?) |
| 1 | Состояние резольвенты:  fibonacci(3, Res). | fibonacci(3, Res).= factorial(N, Result)  Разные главные функторы.  Унификация не успешна. | Откат, переход к следующему предложению. |
| … | … | … | … |
| 2 | Состояние резольвенты:  fibonacci(3, Res). | fibonacci(3, Res). = fibonacci(N, Result)  Унификация успешна.  **Θ = {3 = N, Res = Result }** | Прямой ход  Смена состояния резольвенты |
| 3 | Состояние резольвенты:  fib(3, Res, 1, 0). | fib(N, Result, 1, 0). = factorial(N, Result)  Разные главные функторы.  Унификация не успешна. | Откат, переход к следующему предложению. |
| … | … | … | … |
| 4 | Состояние резольвенты:  fib(3, Res, 1, 0). | fib(3, Res, 1, 0).= fib(1, Result, Result, \_)  Разные числовые константы (3 ≠ 1).  Унификация не успешна. | Откат, переход к следующему предложению. |
| 5 | Состояние резольвенты:  fib(3, Res, 1, 0). | fib(3, Res, 1, 0).= fib(N, Result, X1, X2)  Унификация успешна.  Θ1 = {3 = N, Res = Result, 1 = X1, 0 = X2}  **Θ = {3 = N, Res = Result }** | Прямой ход  Смена состояния резольвенты |
| 6 | Состояние резольвенты:  Sum = 1 + 0,  N\_1 = 3 - 1, fib(N\_1, Res, Sum, 1). | Sum = 1 + 0 | Значение утверждения true  Sum = 1, переход к следующему определению |
| 7 | Состояние резольвенты:  N\_1 = 3 - 1,  fib(N\_1, Res, 1, 1). | N\_1 = 3 - 1 | Значение утверждения true  N\_1 = 2, переход к следующему определению |
| 8 | Состояние резольвенты:  fib(2, Res, 1, 1). | fib(N\_1, Res, 1, 1). = factorial(N, Result)  Разные главные функторы.  Унификация не успешна. | Откат, переход к следующему предложению. |
| … | … | … | … |
| 9 | Состояние резольвенты:  fib(2, Res, 1, 1). | fib(2, Res, 1, 1).= fib(N, Result, X1, X2)  Унификация успешна.  Θ1 = {2 = N, Res = Result, 1 = X1, 1 = X2}  **Θ = {2 = N, Res = Result }** | Прямой ход  Смена состояния резольвенты |
| 10 | Состояние резольвенты:  Sum = 1 + 1,  N\_1 = 2 - 1, fib(N\_1, Res, Sum, 1). | Sum = 1 + 1 | Значение утверждения true  Sum = 2, переход к следующему определению |
| 11 | Состояние резольвенты:  N\_1 = 2 - 1,  fib(N\_1, Res, 2,1). | N\_1 = 2 - 1 | Значение утверждения true  N\_1 = 1, переход к следующему определению |
| 12 | Состояние резольвенты:  fib(1, Res, 2, 2). | fib(1, Res, 2, 2). = factorial(N, Result)  Разные главные функторы.  Унификация не успешна. | Откат, переход к следующему предложению. |
| … | … | … | … |
| 13 | Состояние резольвенты:  fib(1, Res, 2, 2). | fib(1, Res, 2, 2)..= fib(1, Result, Result, \_)    Унификация успешна.  Θ3 = {1 = 1, Res = 2, 2 = Result}  **Θ = {1 = N, Res = 2 }** | Прямой ход  Смена состояния резольвенты |
| 14 | Состояние резольвенты:  ! | ! - указывает системе отменить поиск альтернативных решений для целей до него | Прямой ход  Смена состояния резольвенты |
| 15 | Состояние резольвенты:  Резольвента пуста. |  | **Вывод Res = 2**  Завершение работы программы. |

|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **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 № 19**

|  |  |
| --- | --- |
| **Студент** Козлов М. А.  **Группа** ИУ7-65Б  **Оценка (баллы) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **Преподаватель** Толпинская Н. Б., Строганов Ю. В. |  |

Москва.

2021 г.

**Задание**

Используя хвостовую рекурсию, разработать эффективную программу, (комментируя назначение аргументов), позволяющую:

1. Найти длину списка (по верхнему уровню);
2. Найти сумму элементов числового списка
3. Найти сумму элементов числового списка, стоящих на нечетных позициях исходного списка (нумерация от 0)

Убедиться в правильности результатов.

Для одного из вариантов ВОПРОСА и одного из заданий составить таблицу, отражающую конкретный порядок работы системы.

**Код программы:**

|  |
| --- |
| len(List, L) :- len(List, 0, L).  len([], L, L) :- !.  len([\_ | T], A, L) :- L2 is A + 1, len(T, L2, L).  sum(List, R) :- sum(List, 0, R).  sum([], R, R) :- !.  sum([H | T], A, R) :- R2 is A + H, sum(T, R2, R).  sum\_odd([\_, H | T], [H | R2]) :- sum\_odd(T, R2).  sum\_odd([], []) :- !.  sum\_odd([\_ | []], []) :- !. |

**Примеры работы программы:**

len([1, 2, 3], R).

**R=3**

**1 Solution**

sum ([1, 3, 5, 7], R).

**R=16**

**1 Solution**

sum\_odd [1, 3, 5, 7], R).

**R=10**

**1 Solution**

|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **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 № 20**

|  |  |
| --- | --- |
| **Студент** Козлов М. А.  **Группа** ИУ7-65Б  **Оценка (баллы) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **Преподаватель** Толпинская Н. Б., Строганов Ю. В. |  |

Москва.

2021 г.

**Задание**

Используя хвостовую рекурсию, разработать, комментируя аргументы, эффективную программу, позволяющую:

1. Сформировать список из элементов числового списка, больших заданного значения;
2. Сформировать список из элементов, стоящих на нечетных позициях исходного списка (нумерация от 0);
3. Удалить заданный элемент из списка (один или все вхождения);
4. Преобразовать список в множество (можно использовать ранее разработанные процедуры).

Убедиться в правильности результатов

Для одного из вариантов ВОПРОСА и 1-ого задания составить таблицу, отражающую конкретный порядок работы систем

**Код программы:**

|  |
| --- |
| bigger\_els([], \_, []) :- !.  bigger\_els([H|T], Num, Res) :- H <= Num, bigger\_els(T, Num, Res).  bigger\_els([H|T], Num, [H | Res]) :- H > Num, bigger\_els(T, Num, Res).  odd\_index\_els([\_, H | T], [H | R2]) :- odd\_index\_els(T, R2).  odd\_index\_els([], []) :- !.  odd\_index\_els([\_ | []], []) :- !.  one\_remove\_elem([], \_, []) :- !.  one\_remove\_elem([El | T], El, T) :- !.  one\_remove\_elem([H | T], El, [H | R]) :- one\_remove\_elem(T, El, R).  all\_remove\_elem([], \_, []) :- !.  all\_remove\_elem([El | T], El, R) :- all\_remove\_elem(T, El, R), !.  all\_remove\_elem([H | T], El, [H | R]) :- all\_remove\_elem(T, El, R).  to\_set([], []) :- !.  to\_set([H | T], [H | R]) :- all\_remove\_elem(T, H, T1), to\_set(T1, R). |

**Примеры работы программы:**

bigger\_els([1, 3, 4], 2, Res\_t).

Res\_t=[3,4]

1 Solution

odd\_index\_els([1, 2, 3, 4, 5], Res\_t).

Res\_t=[2,4]

1 Solution

one\_remove\_elem([1, 2, 3, 4, 2, 5, 2], 2, Res\_t).

Res\_t=[1,3,4,2,5,2]

1 Solution

all\_remove\_elem([1, 2, 3, 4, 2, 5, 2], 2, Res\_t).

Res\_t=[1,3,4,5]

1 Solution

to\_set([1, 2, 2, 3, 4, 4, 4, 5, 5, 6], Res\_t).

Res\_t=[1,2,3,4,5,6]

1 Solution

**Вопрос bigger\_els([1, 3, 4], 2, Res\_t).**

|  |  |  |  |
| --- | --- | --- | --- |
| № шага | Состояние резольвенты, и вывод: дальнейшие действия (почему?) | Для каких термов запускается алгоритм унификации: Т1=Т2 и каков результат (и подстановка) | Дальнейшие действия: прямой ход или откат (почему и к чему приводит?) |
| 1 | Состояние резольвенты:  bigger\_els([1, 3, 4], 2, Res\_t). | bigger\_els([1, 3, 4], 2, Res\_t). =  bigger\_els([], \_, [])  Разные структуры списков [1,3,4]≠[]  Унификация не успешна. | Откат, переход к следующему предложению. |
| 2 | Состояние резольвенты:  bigger\_els([1, 3, 4], 2, Res\_t). | bigger\_els([1, 3, 4], 2, Res\_t). = bigger\_els([H|T], Num, Res)  Унификация успешна.  Θ = {1 = H, [3,4] = T, 2 = Num, Res\_t = Res} | Прямой ход  Смена состояния резольвенты |
| 3 | Состояние резольвенты:  1 <= 2,  bigger\_els([3,4], 2, Res). | 1 <= 2 | Значение утверждения true  переход к следующему определению |
| 4 | Состояние резольвенты:  bigger\_els([3,4], 2, Res). | bigger\_els([3,4], 2, Res). = bigger\_els([], \_, [])  Разные структуры списков [3,4]≠[]  Унификация не успешна. | Прямой ход  Смена состояния резольвенты |
| 5 | Состояние резольвенты:  bigger\_els([3, 4], 2, Res\_t). | bigger\_els([3, 4], 2, Res\_t). = bigger\_els([H|T], Num, Res)  Унификация успешна.  Θ1 = {3 = H, [4] = T, 2 = Num, Res\_t = Res} | Прямой ход  Смена состояния резольвенты |
| 6 | Состояние резольвенты:  3 <= 2,  bigger\_els([4], 2, Res\_t). | 3 <= 2 | Значение утверждения false  Откат  Смена состояния резольвенты |
| 7 | Состояние резольвенты:  bigger\_els([3, 4], 2, Res\_t). | bigger\_els([3, 4], 2, Res\_t). = bigger\_els([H|T], Num, [H|Res])  Унификация успешна.  Θ2 = {3 = H, [4] = T, 2 = Num, Res\_t1 = [3|Res]} | Прямой ход  Смена состояния резольвенты |
| 8 | Состояние резольвенты:  3 > 2,  bigger\_els([4], 2, Res\_t). | 3 > 2 | Значение утверждения true  переход к следующему определению |
| 9 | Состояние резольвенты:  bigger\_els([4], 2, Res). | bigger\_els([4], 2, Res). = bigger\_els([], \_, [])  Разные структуры списков [4]≠[]  Унификация не успешна. | Прямой ход  Смена состояния резольвенты |
| 10 | bigger\_els([4], 2, Res\_t). | bigger\_els([4], 2, Res\_t). = bigger\_els([H|T], Num, Res)  Унификация успешна.  Θ3 = {4 = H, [] = T, 2 = Num, Res\_t = Res} | Прямой ход  Смена состояния резольвенты |
| 11 | Состояние резольвенты:  4 <= 2,  bigger\_els([], 2, Res\_t). | 4 <= 2 | Значение утверждения false  Откат  Смена состояния резольвенты |
| 12 | Состояние резольвенты:  bigger\_els([4], 2, Res\_t). | bigger\_els([4], 2, Res\_t). = bigger\_els([H|T], Num, [H|Res])  Унификация успешна.  Θ4 = {4 = H, [] = T, 2 = Num, Res\_t2 = [4|Res]} | Прямой ход  Смена состояния резольвенты |
| 13 | Состояние резольвенты:  4 > 2,  bigger\_els([], 2, Res\_t). | 4 > 2 | Значение утверждения true  переход к следующему определению |
| 14 | bigger\_els([], 2, Res\_t). | bigger\_els([], 2, Res\_t). = bigger\_els([], \_, [])  Унификация успешна.  Θ5 = {[] = [], Res\_t3 = [] } | Прямой ход  Смена состояния резольвенты |
| 15 | Состояние резольвенты:  ! | ! - указывает системе отменить поиск альтернативных решений для целей до него | Прямой ход  Смена состояния резольвенты |
| 16 | Состояние резольвенты:  Резольвента пуста. |  | Завершение работы программы. |