|  |  |
| --- | --- |
| 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»\_

**Отчёт**

**к лабораторной работе № 18**

**По курсу: «Функциональное и логическое программирование»**

**Тема: «Среда Visual Prolog 5.2»**

|  |  |
| --- | --- |
| **Студент** Унтилова А.О.  **Группа** ИУ7-66Б  **Оценка (баллы) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **Преподаватели 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:

|  |
| --- |
| predicates  factorial(integer, integer).  fact(integer, integer, integer).    fibonacci(integer, integer).  fib(integer, integer, integer, integer).    clauses  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).    goal  %factorial(3, Res).  fibonacci(3, Res). |

**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(4, 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 | 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 | 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 | fact(2, Res, 3). | fact(2, Res, 3). = factorial(N, Result)  Разные главные функторы.  Унификация не успешна. | Откат, переход к следующему предложению. |
| … | … | … | … |
| 8 | 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 | 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 | 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 | fact(1, Res, 3). | fact(1, Res, 6). = factorial(1, Result)  Разные главные функторы.  Унификация не успешна. | Откат, переход к следующему предложению. |
| … | … | … | … |
| 12 | 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 | 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 | 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 | 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 | 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 | Sum = 1 + 1,  N\_1 = 2 - 1, fib(N\_1, Res, Sum, 2). | Sum = 1 + 1 | Значение утверждения true  Sum = 2, переход к следующему определению |
| 11 | N\_1 = 2 - 1,  fib(N\_1, Res, 2, 2). | N\_1 = 2 - 1 | Значение утверждения true  N\_1 = 1, переход к следующему определению |
| 12 | fib(1, Res, 2, 2). | fib(1, Res, 2, 2). = factorial(1, Result)  Разные главные функторы.  Унификация не успешна. | Откат, переход к следующему предложению. |
| … | … | … | … |
| 13 | 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»\_

**Отчёт**

**к лабораторной работе № 19**

**По курсу: «Функциональное и логическое программирование»**

**Тема: «Среда Visual Prolog 5.2»**

|  |  |
| --- | --- |
| **Студент** Унтилова А.О.  **Группа** ИУ7-66Б  **Оценка (баллы) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **Преподаватели 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:**

|  |
| --- |
| domains  list = integer\*.    predicates  find\_len(list, integer).  func\_len(list, integer, integer).    find\_sum(list, integer).  func\_sum(list, integer, integer).    find\_sum\_odd(list, integer).  func\_sum\_odd(list, integer, integer).    clauses  find\_len(List, Len) :- func\_len(List, 0, Len).  func\_len([], Len, Len) :- !.  func\_len([\_|T], CurrLen, Len) :- NewLen = CurrLen + 1, func\_len(T, NewLen, Len).    find\_sum(List, Sum) :- func\_sum(List, 0, Sum).  func\_sum([], Sum, Sum) :- !.  func\_sum([H|T], CurS, Sum) :- NewS = CurS + H, func\_sum(T, NewS, Sum).    find\_sum\_odd(List, Sum) :- func\_sum\_odd(List, 0, Sum).  func\_sum\_odd([], Sum, Sum) :- !.  func\_sum\_odd([\_,H|T], CurS, Sum) :- NewS = CurS + H, func\_sum\_odd(T, NewS, Sum).    goal:  %find\_len([1, 2, 3], Len\_t).  %find\_sum([1, 3, 5, 7], Sum\_t).  find\_sum\_odd([1, 3, 5, 7], Sum\_t). |

**Примеры работы программы:**

find\_len([1, 2, 3], Len\_t).

**Len\_t=3**

**1 Solution**

find\_sum([1, 3, 5, 7], Sum\_t).

**Sum\_t=16**

**1 Solution**

find\_sum\_odd([1, 3, 5, 7], Sum\_t).

**Sum\_t=10**

**1 Solution**

**Вопрос find\_len([1, 2, 3], Len\_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:  find\_len([1, 2, 3], Len\_t). | find\_len([1, 2, 3], Len\_t).= find\_len(List, Len)  Унификация успешна.  **Θ = {[1, 2, 3] = List, Len\_t = Len }** | Прямой ход  Смена состояния резольвенты |
| 2 | Состояние резольвенты:  func\_len([1, 2, 3], 0, Len\_t). | func\_len([1, 2, 3], 0, Len\_t). = find\_len(List, Len)  Разные главные функторы.  Унификация не успешна. | Откат, переход к следующему предложению. |
| 3 | Состояние резольвенты:  func\_len([1, 2, 3], 0, Len\_t). | func\_len([1, 2, 3], 0, Len\_t). = func\_len([], Len, Len)  Разные структуры списков [1,2,3]≠[]  Унификация не успешна. | Откат, переход к следующему предложению. |
| 4 | Состояние резольвенты:  func\_len([1, 2, 3], 0, Len\_t). | func\_len([1, 2, 3], 0, Len) = func\_len([\_|T], CurrLen, Len)  Унификация успешна.  Θ1 = {[2, 3] = T, 0 = CurrLen, Len\_t = Len}  **Θ = {[1, 2, 3] = List, Len\_t = Len }** | Прямой ход  Смена состояния резольвенты |
| 5 | Состояние резольвенты:  NewLen = 0 + 1,  func\_len([2, 3], NewLen, Len\_t). | NewLen = 0 + 1 | Значение утверждения true  NewLen = 1,  переход к следующему определению |
| 6 | func\_len([2, 3], 1, Len\_t). | func\_len([2, 3], 1, Len\_t). = find\_len(List, Len)  Разные главные функторы.  Унификация не успешна | Откат, переход к следующему предложению. |
| … | … | … | … |
| 7 | func\_len([2, 3], 1, Len\_t). | func\_len([2, 3], 1, Len\_t). = func\_len([\_|T], CurrLen, Len)  Унификация успешна.  Θ2 = {[3] = T, 1 = CurrLen, Len\_t = Len}  **Θ = {[1, 2, 3] = List, Len\_t = Len }** | Прямой ход  Смена состояния резольвенты |
| 8 | Состояние резольвенты:  NewLen = 1 + 1,  func\_len([3], NewLen, Len\_t). | NewLen = 1 + 1 | Значение утверждения true  NewLen = 2,  переход к следующему определению |
| 9 | func\_len([3], 2, Len\_t). | func\_len([3], 2, Len\_t). = find\_len(List, Len)  Разные главные функторы.  Унификация не успешна | Откат, переход к следующему предложению. |
| … | … | … | … |
| 10 | func\_len([3], 2, Len\_t). | func\_len([3], 2, Len\_t). = func\_len([\_|T], CurrLen, Len)  Унификация успешна.  Θ3 = {[] = T, 2 = CurrLen, Len\_t = Len}  **Θ = {[1, 2, 3] = List, Len\_t = Len }** | Прямой ход  Смена состояния резольвенты |
| 11 | Состояние резольвенты:  NewLen = 2 + 1,  func\_len([], NewLen, Len\_t). | NewLen = 2 + 1 | Значение утверждения true  NewLen = 3,  переход к следующему определению |
| 12 | func\_len([], 3, Len\_t). | func\_len([3], 2, Len\_t). = find\_len(List, Len)  Разные главные функторы.  Унификация не успешна | Откат, переход к следующему предложению. |
| 13 | func\_len([], 3, Len\_t). | func\_len([], 3, Len\_t) = func\_len([], Len, Len)  Унификация успешна.  Θ4 = {[] = [], 3 = Len, Len\_t = 3}  **Θ = {[1, 2, 3] = List, Len\_t = 3 }** | 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 Len\_t = 3**  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»\_

**Отчёт**

**к лабораторной работе № 20**

**По курсу: «Функциональное и логическое программирование»**

**Тема: «Среда Visual Prolog 5.2»**

|  |  |
| --- | --- |
| **Студент** Унтилова А.О.  **Группа** ИУ7-66Б  **Оценка (баллы) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **Преподаватели 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:**

|  |
| --- |
| domains  list = integer\*.    predicates  bigger\_els(list, integer, list).    odd\_index\_els(list, list).  one\_remove\_elem(list, integer, list).  all\_remove\_elem(list, integer, list).  to\_set(list, list).  put(integer, list, list).  permutation(list, list).    clauses  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  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  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).    % lst -> set  to\_set([], []) :- !.  to\_set([H | T], [H | R]) :- all\_remove\_elem(T, H, T1), to\_set(T1, R).    goal  bigger\_els([1, 3, 4], 2, Res\_t).  %odd\_index\_els([1, 2, 3, 4, 5], Res\_t).  %one\_remove\_elem([1, 2, 3, 4, 2, 5, 2], 2, Res\_t).  %all\_remove\_elem([1, 2, 3, 4, 2, 5, 2], 2, Res\_t).  %to\_set([1, 2, 2, 3, 4, 4, 4, 5, 5, 6], Res\_t). |

**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}  **Θ = {1 = H, [3,4] = 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\_t = [3|Res]}  **Θ = {1 = H, [3,4] = T, 2 = Num, Res\_t = 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}  **Θ = {1 = H, [3,4] = 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.  Θ2 = {4 = H, [] = T, 2 = Num, Res\_t = [4|Res]}  **Θ = {1 = H, [3,4] = T, 2 = Num, Res\_t = 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.  Θ3 = {[] = [], Res\_t = [] }  **Θ = {1 = H, [3,4] = T, 2 = Num, Res\_t = Res}** | 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. |  | **Вывод Res\_t = [3, 4]**  Завершение работы программы. |