|  |  |
| --- | --- |
| 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 № 16, 17**

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

**Тема: «Среда Visual Prolog 5.2»**

|  |  |
| --- | --- |
| **Студент** Прохорова Л. А.  **Группа** ИУ7-63Б  **Оценка (баллы) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **Преподаватели Толпинская Н. Б., Строганов Ю. В.** |  |

Москва.

2021 г.

**Лабораторная работа № 16**

**Цель работы** –изучить использование правил в программе: структуру, особенности оформления, а также, способ и принципы выполнения таких программ на Prolog.

**Задачи работы**: приобрести навыки эффективного декларативного описания предметной области с использованием фактов и правил.

Изучить порядок использования фактов и правил в программе на Prolog, принципы и особенности сопоставления и отождествления термов, на основе механизма унификации. Способ формирования и изменения резольвенты. Порядок формирования ответа

**Ход работы**

**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, с объяснениями:

очередная проблема на каждом шаге и метод ее решения;

каково новое текущее состояние резольвенты, как получено;

какие дальнейшие действия? (Запускается ли алгоритм унификации? Каких термов? Почему этих?) ;

вывод по результатам очередного шага и дальнейшие действия.

Код программы представлен в листинге 1.

|  |
| --- |
| Листинг 1  domains  name = symbol.  predicates  parents(name, name, name).  family(name, name, name, name, name, name, name).  clauses  family(GrandmotherMomline, GrandfatherMomline, GrandmotherDadline, GrandfatherDadline, Mother, Father, Child):-  parents(Mother, Father, Child),  parents(GrandmotherMomline, GrandfatherMomline, Mother),  parents(GrandmotherDadline, GrandfatherDadline, Father).  parents("Lena", "Vasya", "Natasha").  parents("Ann", "Egor", "Lena").  parents("Eva", "Dmitry", "Vasya").    goal  %family(GrandmotherMomlineQ, \_, GrandmotherDadlineQ, \_, \_, \_, "Natasha").  %family(\_, GrandfatherMomlineQ, \_, GrandfatherDadlineQ, \_, \_, "Natasha").  %family(GrandmotherMomlineQ, GrandfatherMomlineQ, GrandmotherDadlineQ, GrandfatherDadlineQ, \_, \_, "Natasha").  %family(GrandmotherMomlineQ, \_, \_, \_, \_, \_, "Natasha").  family(GrandmotherMomlineQ, GrandfatherMomlineQ, \_, \_, \_, \_, "Natasha"). |

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

family(GrandmotherMomlineQ, \_, GrandmotherDadlineQ, \_, \_, \_, "Natasha").

**GrandmotherMomlineQ=Ann, GrandmotherDadlineQ=Eva**

**1 Solution**

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

family(\_, GrandfatherMomlineQ, \_, GrandfatherDadlineQ, \_, \_, "Natasha").

**GrandfatherMomlineQ=Egor, GrandfatherDadlineQ=Dmitry**

**1 Solution**

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

family(GrandmotherMomlineQ, GrandfatherMomlineQ, GrandmotherDadlineQ, GrandfatherDadlineQ, \_, \_, "Natasha").

**GrandmotherMomlineQ=Ann, GrandfatherMomlineQ=Egor, GrandmotherDadlineQ=Eva, GrandfatherDadlineQ=Dmitry**

**1 Solution**

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

family(GrandmotherMomlineQ, \_, \_, \_, \_, \_, "Natasha").

**GrandmotherMomlineQ=Ann**

**1 Solution**

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

family(GrandmotherMomlineQ, GrandfatherMomlineQ, \_, \_, \_, \_, "Natasha").

**GrandmotherMomlineQ=Ann, GrandfatherMomlineQ=Egor**

**1 Solution**

Вопрос family(GrandmotherMomlineQ, \_, \_, \_, \_, \_, "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(GrandmotherMomlineQ, \_, \_, \_, \_, \_, "Natasha")  Первое состояние резольвенты - вопрос. Далее запускается алгоритм унификации. | family(GrandmotherMomlineQ, \_, \_, \_, \_, \_, "Natasha") = family(GrandmotherMomline, GrandfatherMomline, GrandmotherDadline, GrandfatherDadline, Mother, Father, Child)  Унификация успешна.  **Θ = {GrandmotherMomlineQ = GrandmotherMomline, "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(GrandmotherMomlineQ, GrandfatherMomline, Mother)  parents(GrandmotherDadline, GrandfatherDadline, Father) | parents(Mother, Father, "Natasha") = family(GrandmotherMomline, GrandfatherMomline, GrandmotherDadline, GrandfatherDadline, Mother, Father, Child)  Разные главные функторы.  Унификация не успешна. | Откат, переход к следующему предложению. |
| 3 | Состояние резольвенты  parents(Mother, Father,"Natasha")  parents(GrandmotherMomlineQ, GrandfatherMomline, Mother)  parents(GrandmotherDadline, GrandfatherDadline, Father) | parents(Mother, Father, "Natasha")  = parents("Lena", "Vasya", "Natasha")  Унификация успешна.  Θ1 **= {**Mother = "Lena", Father = "Vasya"**}**  **Θ = {GrandmotherMomlineQ = GrandmotherMomline, "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. |
| 4 | Состояние резольвенты  parents(GrandmotherMomlineQ, GrandfatherMomline,"Lena")  parents(GrandmotherDadline, GrandfatherDadline,"Vasya") | parents(GrandmotherMomlineQ, GrandfatherMomline,"Lena") = family(GrandmotherMomline, GrandfatherMomline, GrandmotherDadline, GrandfatherDadline, Mother, Father, Child)  Разные главные функторы.  Унификация не успешна. | Откат, переход к следующему предложению. |
| 5 | Состояние резольвенты  parents(GrandmotherMomlineQ, GrandfatherMomline,"Lena")  parents(GrandmotherDadline, GrandfatherDadline,"Vasya") | parents(GrandmotherMomlineQ, GrandfatherMomline,"Lena") = parents("Lena", "Vasya", "Natasha")  Разные константы типа name.  Унификация не успешна. | Откат, переход к следующему предложению. |
| 6 | Состояние резольвенты  parents(GrandmotherMomlineQ, GrandfatherMomline,"Lena")  parents(GrandmotherDadline, GrandfatherDadline,"Vasya") | parents(GrandmotherMomlineQ, GrandfatherMomline,"Lena") = parents("Ann", "Egor", "Lena")  Унификация успешна.  Θ2 **= {**GrandmotherMomlineQ = "Ann", GrandfatherMomline = "Egor"**}**  **Θ = {GrandmotherMomlineQ = "Ann", "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. |
| 7 | Состояние резольвенты  parents(GrandmotherDadline, GrandfatherDadline,"Vasya") | parents(GrandmotherDadline, GrandfatherDadline,"Vasya") = family(GrandmotherMomline, GrandfatherMomline, GrandmotherDadline, GrandfatherDadline, Mother, Father, Child)  Разные главные функторы.  Унификация не успешна. | Откат, переход к следующему предложению. |
| 8 | Состояние резольвенты  parents(GrandmotherDadline, GrandfatherDadline,"Vasya") | parents(GrandmotherDadline, GrandfatherDadline,"Vasya") = parents("Lena", "Vasya", "Natasha")  Разные константы типа name.  Унификация не успешна. | Откат, переход к следующему предложению. |
| ... | ... | ... | ... |
| 10 | Состояние резольвенты  parents(GrandmotherDadline, GrandfatherDadline,"Vasya") | parents(GrandmotherDadline, GrandfatherDadline,"Vasya") = parents("Eva", "Dmitry", "Vasya")  Унификация успешна.  Θ3 **= {**GrandmotherDadline = "Eva", GrandfatherDadline = "Vasya"**}**  **Θ = {GrandmotherMomlineQ = "Ann", "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. |
| 11 | Состояние резольвенты:  Резольвента пуста |  | Найден ответ  **GrandmotherMomlineQ = "Ann"**  Поиск альтернативного решения. Откат к предыдущему состоянию резольвенты.  Освобождение переменных GrandmotherDadline, GrandfatherDadline |
| 12 | Состояние резольвенты  parents(GrandmotherDadline, GrandfatherDadline,"Vasya") |  | Все знания в базе знаний просмотрены.  Поиск альтернативного решения. Откат к предыдущему состоянию резольвенты.  Освобождение переменных  GrandmotherMomlineQ, GrandfatherMomline |
| 13 | Состояние резольвенты  parents(GrandmotherMomlineQ, GrandfatherMomline,"Lena")  parents(GrandmotherDadline, GrandfatherDadline,"Vasya") | parents(GrandmotherMomlineQ, GrandfatherMomline,"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(GrandmotherMomlineQ, GrandfatherMomline, Mother)  parents(GrandmotherDadline, GrandfatherDadline, 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(GrandmotherMomlineQ, GrandfatherMomline, Mother)  parents(GrandmotherDadline, GrandfatherDadline, 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 GrandmotherMomlineQ, Child |
| 16 | Состояние резольвенты:  family(GrandmotherMomlineQ, \_, \_, \_, \_, \_, "Natasha") | family(GrandmotherMomlineQ, \_, \_, \_, \_, \_, "Natasha") = parents("Lena", "Vasya", "Natasha")  Разные главные функторы.  Унификация не успешна. | Откат, переход к следующему предложению. |
| ... | ... | ... | ... |
| 18 | Состояние резольвенты:  family(GrandmotherMomlineQ, \_, \_, \_, \_, \_, "Natasha") | family(GrandmotherMomlineQ, \_, \_, \_, \_, \_, "Natasha") = parents("Eva", "Dmitry", "Vasya")  Разные главные функторы.  Унификация не успешна. | Все знания в базе знаний просмотрены.  Вывод **GrandmotherMomlineQ = Ann**  Завершение работы программы. |

**Лабораторная работа № 17**

Цель работы – изучить способы организации эффективных программ на Prolog, особенности использования системных предикатов и порядок выполнения программ с их использованием.

Задачи работы: приобрести навыки эффективного описания предметной области с использованием фактов и правил.

Изучить возможность использования системных предикатов в программе на Prolog, принципы и особенности порядка работы в этом случае. Способ формирования и изменения резольвенты в этом случае и порядок формирования ответа.

Задание

В одной программе написать правила, позволяющие найти

1. Максимум из двух чисел

а) без использования отсечения,

в) с использованием отсечения;

2. Максимум из трех чисел

а) без использования отсечения,

в) 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.

Код программы представлен в листинге 2.

|  |
| --- |
| predicates  max(real, real, real)  max\_cat(real, real, real)  max(real, real, real, real)  max\_cat(real, real, real, real)    clauses  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).    goal  %max(3, 4, Max).  %max(2, 1, Max).    %max\_cat(2, 1, Max).  %max\_cat(1, 3, Max).  %max(5, 3, 2, Max).  %max(3, 5, 2, Max).  %max(1, 2, 8, Max).    %max\_cat(5, 3, 2, Max).  %max\_cat(3, 5, 2, Max).  max\_cat(1, 2, 8, Max). |

1. Максимум из двух чисел

а) без использования отсечения

Максимум стоит на второй позиции

*max(3, 4, Max).*

**Max=4**

**1 Solution**

Максимум стоит на первой позиции

*max(2, 1, Max).*

**Max=2**

**1 Solution**

в) с использованием отсечения;

*max\_cat(2, 1, Max).*

Максимум стоит на первой позиции

**Max=2**

**1 Solution**

Максимум стоит на второй позиции

*max\_cat(1, 3, Max).*

**Max=3**

**1 Solution**

2. Максимум из трех чисел

а) без использования отсечения

Максимум стоит на первой позиции

*max(5, 3, 2, Max).*

**Max=5**

**1 Solution**

Максимум стоит на второй позиции

*max(3, 5, 2, Max).*

**Max=5**

**1 Solution**

Максимум стоит на третьей позиции

*max(1, 2, 8, Max).*

**Max=8**

**1 Solution**

в) с использованием отсечения;

Максимум стоит на первой позиции

*max\_cat(5, 3, 2, Max).*

**Max=5**

**1 Solution**

Максимум стоит на второй позиции

*max\_cat(3, 5, 2, Max).*

**Max=5**

**1 Solution**

Максимум стоит на третьей позиции

*max\_cat(1, 2, 8, Max).*

**Max=8**

**1 Solution**

**Обоснование необходимости каждого условия:**

Без использования отсечения

max(X, Y, Z, X):- X >= Y, X >= Z. → Необходимо проверить является ли Х наибольшим.

max(X, Y, Z, Y):- Y >= X, Y >= Z. → Необходимо проверить, истинность Y >= X, т. к. даже если предыдущий шаг выполнен успешно, программа продолжит работу. Без этого условия, У будет ответом, например в случае Х=3 У = 2 Z= 1, что неверно.

max(X, Y, Z, Z):- Z >= X, Z >= Y. Необходимо проверить что Z больше двух других переменных, так как программа продолжит выполнение даже если предыдущие шаги были выполнены успешно, что может привести к неверному результату.

С отсечением

max\_cat(X, Y, Z, X):- X >= Y, X >= Z, !. → Проверка что Х наибольший. Если да, то отсекаем.

max\_cat(\_, Y, Z, Y):- Y >= Z, !. → Уже проверено, что Х не максимальный, остается только проверить, что У максимальный.

max\_cat(\_, \_, Z, Z). → Х и У не максимальные → ответ Z.

Вопрос max(5, 3, 2, Max).

|  |  |  |  |
| --- | --- | --- | --- |
| № 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:  max(5, 3, 2, Max) | max(5, 3, 2, Max) = max(X, Y, X)  Разная арность.  Унификация не успешна. | Откат, переход к следующему предложению. |
| ... | ... | ... | ... |
| 3 | Состояние резольвенты:  max(5, 3, 2, Max) | max(5, 3, 2, Max) = max\_cat(X, Y, X)  Разные главные функторы.  Унификация не успешна. | Откат, переход к следующему предложению. |
| ... | ... | ... | ... |
| 5 | Состояние резольвенты:  max(5, 3, 2, Max) | max(5, 3, 2, Max) = max(X, Y, Z, X)  Унификация успешна  **Θ = {5 = X, 3 = Y, 2 = Z, Max = X}** | Прямой ход  Смена состояния резольвенты |
| 6 | Состояние резольвенты:  5 >= 3  5 >= 2 | 5 >= 3 yes | Прямой ход  Смена состояния резольвенты |
| 7 | Состояние резольвенты:  5 >= 2 | 5 >= 2 yes | Прямой ход  Смена состояния резольвенты |
| 8 | Состояние резольвенты:  Резольвента пуста |  | Найден ответ **Max = 5**  **Откат**  Поиск альтернативного решения. Обратный ход. |
| 9 | Состояние резольвенты:  max(5, 3, 2, Max) | max(5, 3, 2, Max) = max(X, Y, Z, Y)  Унификация успешна  **Θ = {5 = X, 3 = Y, 2 = Z, Max = Y}** | Прямой ход  Смена состояния резольвенты |
| 10 | Состояние резольвенты:  3 >= 5  3 >= 2 | 3 >= 5 no | **Откат**  Смена состояния резольвенты |
| 11 | Состояние резольвенты:  max(5, 3, 2, Max) | max(5, 3, 2, Max) = max(X, Y, Z, Z)  Унификация успешна  **Θ = {5 = X, 3 = Y, 2 = Z, Max = Z}** | Прямой ход  Смена состояния резольвенты |
| 12 | 2 >= 5  2 >= 3 | 2 >= 5 no | **Откат**  Смена состояния резольвенты |
| 13 | Состояние резольвенты:  max(5, 3, 2, Max) | max(5, 3, 2, Max) = max\_cat(X, Y, Z, X)  Разные главные функторы.  Унификация не успешна. | Откат, переход к следующему предложению. |
| ... | ... | ... | ... |
| 15 | Состояние резольвенты:  max(5, 3, 2, Max) | max(5, 3, 2, Max) = max\_cat(\_, \_, Z, Z)  Разные главные функторы.  Унификация не успешна. | Все знания в базе знаний проверены.  **Вывод Max = 5**  Завершение работы программы. |

Вопрос max\_cat(5, 3, 2, Max).

|  |  |  |  |
| --- | --- | --- | --- |
| № 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:  max\_cat(5, 3, 2, Max) | max\_cat(5, 3, 2, Max) = max(X, Y, X)  Разные главные функторы.  Унификация не успешна. | Откат, переход к следующему предложению. |
| ... | ... | ... | ... |
| 3 | Состояние резольвенты:  max\_cat(5, 3, 2, Max) | max\_cat(5, 3, 2, Max) = max\_cat(X, Y, X)  Разная арность.  Унификация не успешна. | Откат, переход к следующему предложению. |
| ... | ... | ... | ... |
| 5 | Состояние резольвенты:  max\_cat(5, 3, 2, Max) | max\_cat(5, 3, 2, Max) = max(X, Y, Z, X)  Разные главные функторы.  Унификация не успешна. | Откат, переход к следующему предложению. |
| ... | ... | ... | ... |
| 8 | Состояние резольвенты:  max\_cat(5, 3, 2, Max) | max\_cat(5, 3, 2, Max) = max\_cat(X, Y, Z, X)  Унификация успешна.  **Θ = {5 = X, 3 = Y, 2 = Z, Max =X }** | Прямой ход  Смена состояния резольвенты |
| 9 | Состояние резольвенты:  5 >= 3  5 >= 2  ! | 5 >= 3 yes | Прямой ход  Смена состояния резольвенты |
| 10 | Состояние резольвенты:  5 >= 2  ! | 5 >= 2 yes | Прямой ход  Смена состояния резольвенты |
| 11 | Состояние резольвенты:  ! | ! - указывает прологу отменить поиск альтернатив для целей до него | Прямой ход  Смена состояния резольвенты |
| 12 | Состояние резольвенты:  Резольвента пуста. |  | **Вывод Max = 5**  Завершение работы программы. |

**Ответы на вопросы:**

**1. В каком случае система запускает алгоритм унификации? (Как эту необходимость на формальном уровне распознает система?)**

Процесс унификации в случае, когда необходимо доказать какое-то утверждение. Если резольвента не пуста, есть, что доказывать. Также пользователь может запустить этот процесс вручную с помощью утверждения Т1=Т2.

**2. Каковы назначение и результат использования алгоритма унификации?**

Унификация позволяет формализовать процесс логического вывода. Назначение - поиск знания, которое является ответом на конкретный вопрос. Результат унификации — ответ «да» или «нет»(подходит знание или нет).

**3. Какое первое состояние резольвенты?**

Первое состояние резольвенты - заданный вопрос.

**4. Как меняется резольвента?**

Резольвента меняется в два этапа.

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.

Успешное завершение работы программы достигается, когда резольвента пуста.

**5. В каких пределах программы уникальны переменные?**

Именованные переменные уникальны в рамках одного предложения, т. е. в разных предложениях одно и то же имя переменной может использоваться для обозначения разных объектов. Анонимные переменные уникальны везде.

**6. Как применяется подстановка, полученная с помощью алгоритма унификации?**

Если алгоритм унификации завершился успешно и найдена подстановка, соответствующие переменные конкретизируются полученными значениями.

**7. В каких случаях запускается механизм отката?**

Механизм отката к предыдущему шагу выполняется в случае, когда унификация завершается

тупиковой ситуацией(неудачей) или для того, чтобы получить другое решение.