|  |  |
| --- | --- |
| 1. 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*

*По предмету: «Функциональное и логическое программирование»*

Преподаватель: Толпинская Н.Б.

Студент: Турсунов Ж.Р.

Группа: ИУ7-66Б

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, 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.

Т.к. резольвента хранится в виде стека, то состояние резольвенты требуется отображать в столбик: вершина – сверху! Новый шаг надо начинать с нового состояния резольвенты!

|  |  |  |  |
| --- | --- | --- | --- |
| № 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… | … | … | Комментарий, вывод… |
| … | … | … | … |

**Текст программы**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **predicates**  **family(symbol Child, symbol Father, symbol Mother).**  **grand(symbol Child, symbol F\_f, symbol M\_f, symbol F\_m, symbol M\_m).**  **clauses**  **family(alexey, vladimir, elena).**  **family(inna, egor, kseniya).**  **family(vladimir, alexey, inna).**  **family(oleg, kirill, nadezhda).**  **family(olga, andrey, anna).**  **family(ivan, oleg, olga).**  **grand(Child, F\_f, M\_f, F\_m, M\_m) :- family(Child, Father, Mother),**  **family(Father, F\_f, M\_f), family(Mother, F\_m, M\_m).**  **goal**  **grand(ivan, \_, M\_f, \_, M\_m). %task\_1**  **%** **grand(ivan, F\_f, \_, F\_m, \_). %task\_2**  **%** **grand(ivan, F\_f, M\_f, F\_m, M\_m). %task\_3**  **%** **grand(ivan, \_, \_, \_, M\_m). %task\_4**  **%** **grand(ivan, \_, \_, F\_m, M\_m). %task\_5**   1. **По имени субъекта определить всех его бабушек (предки 2-го колена)**  |  | | --- | | **goal**  **grand(ivan, \_, Grandma\_f, \_, Grandma\_m).**  **Output:**  Grandma\_f=Nadezhda, Grandma\_m=Anna  1 Solution |  1. **По имени субъекта определить всех его дедушек (предки 2-го колена)**  |  | | --- | | **goal**  **grand(ivan, Grandpa\_f, \_, Grandpa\_m, \_).**  **Output:**  Grandpa\_f=kirill, Grandpa\_m=andrey   1. Solution |  1. **По имени субъекта определить всех его бабушек и дедушек (предки 2-го колена)**  |  | | --- | | **goal**  **grand(ivan, Grandpa\_f, Grandma\_f, Grandpa\_m, Grandpa\_m).**    **Output:**  F\_f=kirill, M\_f=nadezhda, F\_m=andrey, M\_m=anna  1 Solution |  1. **По имени субъекта определить всех его бабушек и дедушек (предки 2-го колена)**  |  | | --- | | **goal**  **grand(ivan, \_, \_, \_, Grandma\_m).**    **Output:**  Grandma\_m=anna   1. Solution |  1. **По имени субъекта определить его бабушку и дедушку по материнской линии (предки 2-го колена).**  |  | | --- | | **goal**  **grand(ivan, \_, \_, Grandpa\_m, Grandma\_m).**    **Output:**  Grandpa\_m=andrey, Grandma\_m=anna  1 Solution | |

**Описание порядка работы системы**

1. **По имени субъекта определить всех его дедушек (предки 2-го колена)**

|  |
| --- |
| **goal**  **grand(ivan, Grandpa\_f, \_, Grandpa\_m, \_).**  **Output:**  Grandpa\_f=kirill, Grandpa\_m=andrey   1. Solution |

|  |  |  |  |
| --- | --- | --- | --- |
| №  шага | Состояния резольвенты, и вывод: дальнейшие дейстия (почему?) | Для каких термов запускается алгоритм унификации: T1=T2 и каков результат (и подстановка) | Дальнейшие действия: прямой ход или откат (почему и к чему приводит?) |
| 1 | **grand(ivan, Grandpa\_f, \_, Grandpa\_m, \_)** | **Попытка унификации:**  **grand(ivan, Grandpa\_f, \_, Grandpa\_m, \_)**  **=**  **family(alexey, vladimir, elena)**  **Результат:**  Неудача (разные функторы) | Прямой ход, переход к следующему предложению. |
| 2 | **grand(ivan, Grandpa\_f, \_, Grandpa\_m, \_)** | **Попытка унификации:**  **grand(ivan, Grandpa\_f, \_, Grandpa**\_**m,** \_**)**  **=**  **family(inna, egor, kseniya)**  **Результат:**  Неудача (разные функторы) | Прямой ход, переход к следующему предложению. |
| 3 | **grand(ivan, Grandpa\_f, \_, Grandpa\_m, \_)** | **Попытка унификации:**  **grand(ivan, F\_f, \_, F\_m,** \_**)**  **=**  **family(vladimir, alexey, inna)**  **Результат:**  Неудача (разные функторы) | Прямой ход, переход к следующему предложению |
| 4 | **grand(ivan, Grandpa\_f, \_, Grandpa\_m, \_)** | **Попытка унификации:**  **grand(ivan, F\_f, \_, F**\_**m,** \_**)**  **=**  **family(oleg, kirill, nadezhda)**  **Результат:**  Неудача (разные функторы) | Прямой ход, переход к следующему предложению |
| 5 | **grand(ivan, Grandpa\_f, \_, Grandpa\_m, \_)** | **Попытка унификации:**  **grand(ivan, F\_f, \_, F**\_**m,** \_**)**  **=**  **family(olga, andrey, anna)**  **Результат:**  Неудача (разные функторы) | Прямой ход, переход к следующему предложению |
| 6 | **grand(ivan, Grandpa\_f, \_, Grandpa\_m, \_)** | **Попытка унификации:**  **grand(ivan, F\_f, \_, F**\_**m,** \_**)**  **=**  **family(ivan, oleg, olga)**  **Результат:**  Неудача (разные функторы) | Прямой ход, переход к следующему предложению |
| 7 | **grand(ivan, Grandpa\_f, \_, Grandpa\_m, \_)** | **Попытка унификации:**  **grand(ivan, F\_f, \_, F**\_**m,** \_**)**  **=**  **grand(Child, F**\_**f, M**\_**f, F**\_**m, M**\_**m)**  **Результат:**  Успех (подстановка)  **{Child = ivan, F\_f = F\_f, F\_m = F\_m}** | Редукция цели, прямой ход к сопоставлению **family(Child, Father, Mother),** поиск с начала предложений. |
| 8 | **family(ivan, Father, Mother),**  **family(Father, F\_f, \_),**  **family(Mother, F\_m, \_)** | **Попытка унификации:**  **family(ivan, Father, Mother)**  **=**  **family(alexey, vladimir, elena)**  **Результат**:  Неудача **(ivan ≠ alexey)** | Прямой ход, переход к следующему предложению |
| 9 | **family(ivan, Father, Mother),**  **family(Father, F\_f, \_),**  **family(Mother, F\_m, \_)** | **Попытка унификации:**  **family(ivan, Father, Mother)**  **=**  **family(inna, egor, kseniya)**  **Результат**:  Неудача **(ivan ≠ inna)** | Прямой ход, переход к следующему предложению |
| 10 | **family(ivan, Father, Mother),**  **family(Father, F\_f, \_),**  **family(Mother, F\_m, \_)** | **Попытка унификации:**  **family(ivan, Father, Mother)**  **=**  **family(vladimir, alexey, inna)**  **Результат:**  Неудача **(ivan ≠ vladimir)** | Прямой ход, переход к следующему предложению |
| 11 | **family(ivan, Father, Mother),**  **family(Father, F\_f, \_),**  **family(Mother, F\_m, \_)** | **Попытка унификации:**  **family(ivan, Father, Mother)**  **=**  **family(oleg, kirill, nadezhda)**  **Результат:**  Неудача **(ivan ≠ oleg)** | Прямой ход, переход к следующему предложению |
| 12 | **family(ivan, Father, Mother),**  **family(Father, F\_f, \_),**  **family(Mother, F\_m, \_)** | **Попытка унификации:**  **family(ivan, Father, Mother)**  **=**  **family(olga, andrey, anna)**  **Результат:**  Неудача **(ivan ≠ olga)** | Прямой ход, переход к следующему предложению |
| 13 | **family(ivan, Father, Mother),**  **family(Father, F\_f, \_),**  **family(Mother, F\_m, \_)** | **Попытка унификации:**  **family(ivan, Father, Mother)**  **=**  **family(ivan, oleg, olga)**  **Результат:**  Успех (подстановка)  **{Father = oleg, Mother = olga}** | Прямой ход к сопоставлению **family(oleg, F\_f, \_),** поиск с начала предложений. |
| 14 | **family(Father, F\_f, \_),**  **family(Mother, F\_m, \_)** | **Попытка унификации:**  **family(oleg, F\_f, \_)**  **=**  **family(alexey, vladimir, elena)**  **Результат:**  Неудача **(oleg ≠ alexey)** | Прямой ход, переход к следующему предложению |
| 15 | **family(Father, F\_f, \_),**  **family(Mother, F\_m, \_)** | **Попытка унификации:**  **family(oleg, F\_f, \_)**  **=**  **family(inna, egor, kseniya)**  **Результат:**  Неудача **(oleg ≠ inna)** | Прямой ход, переход к следующему предложению |
| 16 | **family(Father, F\_f, \_),**  **family(Mother, F\_m, \_)** | **Попытка унификации:**  **family(oleg, F\_f, \_)**  **=**  **family(vladimir, alexey, inna)**  **Результат:**  Неудача **(oleg ≠ vladimir)** | Прямой ход, переход к следующему предложению |
| 17 | **family(Father, F\_f, \_),**  **family(Mother, F\_m, \_)** | **Попытка унификации:**  **family(oleg, F\_f, \_)**  **=**  **family(oleg, kirill, nadezhda)**  **Результат:**  Успех (подстановка)  **{F\_f = kirill}** | Прямой ход к сопоставлению **family(olga, F\_m, \_),** поиск с начала предложений. |
| 18 | **family(olga, F\_m, \_)** | **Попытка унификации:**  **family(olga, F\_m, \_)**  **=**  **family(alexey, vladimir, elena)**  **Результат:**  Неудача **(olga ≠ alexey)** | Прямой ход, переход к следующему предложению |
| 19 | **family(olga, F\_m, \_)** | **Попытка унификации:**  **family(olga, F\_m, \_)**  **=**  **family(inna, egor, kseniya)**  **Результат:**  Неудача **(olga ≠ inna)** | Прямой ход, переход к следующему предложению |
| 20 | **family(olga, F\_m, \_)** | **Попытка унификации:**  **family(olga, F\_m, \_)**  **=**  **family(vladimir, alexey, inna)**  **Результат:**  Неудача **(olga ≠ vladimir)** | Прямой ход, переход к следующему предложению |
| 21 | **family(olga, F\_m, \_)** | **Попытка унификации:**  **family(olga, F\_m, \_)**  **=**  **family(oleg, kirill, nadezhda)**  **Результат:**  Неудача **(olga ≠ oleg)** | Прямой ход, переход к следующему предложению |
| 22 | **family(olga, F\_m, \_)** | **Попытка унификации:**  **family(olga, F\_m, \_)**  **=**  **family(olga, andrey, anna)**  **Результат:**  Успех (подстановка)  **{F\_m = andrey}** | Вывод:  **F\_f = kirill,**  **F\_m = andrey.**  Опустошение резольвенты вследствие успешной унификации.  Откат, переход к предыдущему состоянию резольвенты. Реконкретизация **F\_m**. |
| 23 | **family(olga, F\_m, \_)** | **Попытка унификации:**  **family(olga, F\_m, \_)**  **=**  **family(ivan, oleg, olga)**  **Результат:**  Неудача **(olga ≠ ivan)** | Прямой ход к следующему предложению. |
| 24 | **family(olga, F\_m, \_)** | **Попытка унификации:**  **family(olga, F\_m, \_)**  **=**  **grand(Child, F\_f, M\_f, F\_m, M\_m)**  **Результат:**  Неудача (разные функторы) | Конец БЗ, откат, переход к предыдущему состоянию резольвенты (шаг 17), реконкретизация F\_f. |
| 25 | **family(Father, F\_f, \_),**  **family(Mother, F\_m, \_)** | **Попытка унификации:**  **family(oleg, F\_m, \_)**  **=**  **family(olga, andrey, anna)**  **Результат:**  Неудача **(oleg ≠ olga)** | Прямой ход к следующему предложению. |
| 26 | **family(Father, F\_f, \_),**  **family(Mother, F\_m, \_)** | **Попытка унификации:**  **family(oleg, F\_m, \_)**  **=**  **family(ivan, oleg, olga)**  **Результат:**  Неудача **(oleg ≠ ivan)** | Прямой ход к следующему предложению. |
| 27 | **family(Father, F\_f, \_),**  **family(Mother, F\_m, \_)** | **Попытка унификации:**  **family(oleg, F\_m, \_)**  **=**  **grand(Child, F\_f, M\_f, F\_m, M\_m)**  **Результат**:  Неудача (разные функторы) | Конец БЗ, откат, переход к предыдущему состоянию резольвенты  (шаг 13), реконкретизация Father и Mother. |
| 28 | **family(ivan, Father, Mother),**  **family(Father, F\_f, \_),**  **family(Mother, F\_m, \_)** | **Попытка унификации:**  **family(ivan, Father, Mother)**  **=**  **grand(Child, F\_f, M\_f, F\_m, M\_m)**  **Результат:**  Неудача (разные функторы) | Откат, переход к предыдущему состоянию резольвенты (шаг 7). |
| 29 | **grand(ivan, Grandpa\_f, \_, Grandpa\_m, \_)**  конец clauses;  опустошение резольвенты;  завершение работы. |  |  |

**Теоретические вопросы.**

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

Если есть что доказывать (цель), то процесс унификации запускается автоматически. Формально: если резольвента не пуста – запускается алгоритм унификации.

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

Назначение алгоритма унификации заключается в попарном сопоставлении термов и попытке построить для них общий пример. Унификация может завершаться успехом или тупиковой ситуацией (неудачей).

1. **Какое первое состояние резольвенты?**

Если задан простой вопрос, то сначала он попадает в резольвенту.

1. **Как меняется резольвента?**

Изменение резольвенты происходит в 2 этапа:

1) из стека выбирается подцель (верхняя, т.к. стек) и для нее выполняется редукция, т.е. замена подцели на тело найденного правила;

2) к полученной конъюнкции целей применяется подстановка (наибольший общий унификатор выбранной цели и заголовка сопоставленного с этой целью правила).

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

Переменные уникальны в пределах предложения, т.е. в рамках предложения одно и то же имя принадлежит одной и той же переменной. Исключение – анонимные переменные (обозначаются символом нижнего подчеркивания «\_») – каждая такая переменная является отдельной сущностью и применяется, когда ее значение неважно для данного предложения.

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

Применение подстановки {*x1=t1, …, xn=tn*} заключается в замене каждого вхождения переменной *xi* на соответствующий терм *ti*.

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

Механизм отката запускается, если возникла тупиковая ситуация (достигнут конец БЗ) либо резольвента пуста. В таких случаях происходит откат к предыдущему состоянию резольвенты.