|  |  |
| --- | --- |
| 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»

**ОТЧЕТ**

*к лабораторной работе №16*

*По курсу: «Функциональное и логическое программирование»*

Студент ИУ7-65Б

Юмаев А.Р.

Преподаватель

Строганов Ю.В.

Толпинская Н.Б

*Москва, 2020 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 вопрпосов. Использовать конъюнктивные правила и простой вопрос.**

Для одного из вариантов ВОПРОСА и конкретной БЗ составить таблицу, отражающую конкретный порядок работы системы, с объяснениями:

1. очередная проблема на каждом шаге и метод ее решения;
2. каково новое текущее состояние резольвенты, как получено;
3. какие дальнейшие действия? (Запускается ли алгоритм унификации? Каких термов? Почему этих?);
4. вывод по результатам очередного шага и дальнейшие действия.

**Ответы на вопросы**

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

В случае, если есть что доказывать (задана цель), программа автоматически запускает алгоритм унификации, также он может быть запущен вручную. Формально - если в резольвенте есть цель - для нее запускается алгоритм унификации (унифицируется текущая цель, выбранная из резольвенты, со знаниями в БЗ).

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

Для поиска ответа на вопрос системе необходимо найти подходящее знание в БЗ, для поиска такого знания используется алгоритм унификации. Формально, он помогает системе понять, что заголовок подошел: алгоритм попарно пытается сопоставить термы (текущую цель и термы из БЗ) и построить для них общий пример (для этого используется подстановка).

1. **Какое первое состояние резольвенты?**

На первом шаге в резольвенте находится заданный вопрос (цель).

1. **Как меняется резольвента?**

В резольвенте хранятся цели, истинность которых необходимо доказать. Для хранения целей в резольвенте используется стек. Она (резольвента) меняется в ходе доказательства с помощью алгоритма редукции (замена текущей цели на тело правила, найденного в БЗ с помощью алгоритма унификации). Преобразование происходит следующим образом:

1. берется верхняя цель и заменяется на тело правила (если речь идет о факте, так как факт - частный случай правила, не имеющий тела, цель просто убирается), найденного путем сопоставления алгоритмом унификации из БЗ.
2. к полученной конъюнкции целей применяется подстановка, полученная алгоритмом унификации
3. **В каких пределах программы уникальны переменные?**

Именованная переменная уникальна в рамках предложения, в котором она используются. Любая анонимная переменная является уникальной.

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

Применение подстановки заключается в замене каждого вхождения переменной на соответствующий терм (). В результате применения подстановки переменные конкретизируются значениями, которые будут далее использованы при доказательстве истинности тела выбранного правила то есть значения переменных переходят на следующих шаг доказательства.

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

Механизм отката запускается при возникновении тупиковой ситуации.

|  |
| --- |
| domains name = symbol  predicates  find\_mother(name, name)  find\_father(name, name)  find\_grandparents(name, name, name, name, name)  clauses  find\_mother("Olga", "Vladislav").  find\_mother("Olga", "Kirill").  find\_mother("Katya", "Victoria").  find\_mother("Sasha", "Anna").  find\_mother("Victoria", "Nikita").  find\_mother("Anna", "Jenny").    find\_father("Dima", "Vladislav").  find\_father("Dima", "Kirill").  find\_father("Kolya", "Victoria").  find\_father("Sergey", "Anna").  find\_father("Vladislav", "Nikita").  find\_father("Kirill", "Jenny").    find\_grandparents(Child, MothersGranny, MothersGrandDad, FathersGranny, FathersGrandDad) :-   find\_mother(MothersGranny, X),  find\_mother(X, Child),  find\_mother(FathersGranny, Y),  find\_father(Y, Child),  find\_father(FathersGrandDad, Z),  find\_father(Z, Child),  find\_father(MothersGrandDad, K),  find\_mother(K, Child). |

**Примеры целей и результатов работы программы**

Задание: по имени субъекта определить всех его бабушек (предки 2-го колена)

Цель: find\_grandparents("Jenny", MothersGranny, \_, FathersGranny, \_).

Результат: **MothersGranny**=Sasha, **FathersGranny**=Olga

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

Цель: find\_grandparents("Jenny", \_, MothersGrandDad, \_, FathersGrandDad).

Результат: **MothersGrandDad**=Sergey, **FathersGrandDad**=Dima

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

Цель: find\_grandparents("Jenny", MothersGranny, MothersGrandDad, FgrandMa, FgrandPa).

Результат: **MothersGranny**=Sasha, **MothersGrandDad**=Sergey,   
**FgrandMa**=Olga, **FgrandPa**=Dima

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

Цель: find\_grandparents("Jenny", MothersGranny, \_, \_, \_).

Результат: **MothersGranny**=Sasha

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

Цель: find\_grandparents("Jenny", \_, MothersGrandDad, \_, \_).

Результат: **MothersGrandDad**=Sergey

**Описание порядка работы системы**

**Цель:** find\_grandparents("Jenny", MothersGranny, \_, \_, \_).

|  |  |  |  |
| --- | --- | --- | --- |
| № 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 | Цель заносится в резольвенту.  find\_grandparents("Jenny", MothersGranny, \_, \_, \_).  Запуск процесса редукции для find\_grandparents("Jenny", MothersGranny, \_, \_, \_) | Попытка унификации:  ТЦ: find\_grandparents("Jenny", MothersGranny, \_, \_, \_)  И знаний БЗ, пока не будет найден подходящее знание или не будет «просмотрена» вся БЗ. | Прямой ход, переход к следующему предложению. |
|  | **find\_grandparents("Jenny", MothersGranny, \_, \_, \_).** | Попытка унификации:  ТЦ: find\_grandparents("Jenny", MothersGranny, \_, \_, \_)  find\_grandparents(Child, MothersGranny, MothersGrandDad, FathersGranny, FathersGrandDad) - знание в БЗ  Результат: успех  Подстановка: {Child="Jenny", MothersGranny= MothersGranny}. | Преобразование резольвенты (замена текущей цели на тело найденного с помощью унификации правила, применение полученной подстановки),  Прямой ход |
| 2 | **find\_mother(MothersGranny, X)**,  find\_mother(X, "Jenny"),  find\_mother(\_, Y),  find\_father(Y, "Jenny"),  find\_father(\_, Z),  find\_father(Z, "Jenny"),  find\_father(\_,  find\_mother(K, "Jenny").  Цель была заменена на подцели правила, найденного в БЗ, после чего к подцелям резольвенты применяется подстановка.  запуск редукции для **find\_mother(MothersGranny, X)** | Попытка унификации:  ТЦ:find\_mother(MothersGranny, X)  find\_mother("Olga", "Vladislav") - знание в БЗ  результат: успех  подстановка: {MothersGranny="Olga", X="Vladislav"} | Преобразование резольвенты (замена текущей цели на **пустое** тело факта применение полученной подстановки),  Прямой ход |
| 3 | **find\_mother("Vladislav", "Jenny")**,  find\_mother(\_, Y),  find\_father(Y, "Jenny"),  find\_father(\_, Z),  find\_father(Z, "Jenny"),  find\_father(\_,  find\_mother(K, "Jenny").  Выбор верхней цели, запуск редукции | Попытка унификации:  ТЦ:find\_mother("Vladislav", "Jenny"),  И знаний БЗ, пока не будет найден подходящее знание или не будет «просмотрена» вся БЗ. | Прямой ход, переход к следующему предложению, по окончании был достигнут конец БЗ, термов, при которых унификации завершилась успехом найдено не было, резольвента не пуста. Откат. |
| 4 | **find\_mother(MothersGranny, X)**,  find\_mother(X, "Jenny"),  find\_mother(\_, Y),  find\_father(Y, "Jenny"),  find\_father(\_, Z),  find\_father(Z, "Jenny"),  find\_father(\_,  find\_mother(K, "Jenny").  запуск редукции **find\_mother(MothersGranny, X)** | Попытка унификации:  ТЦ:find\_mother(MothersGranny, X)  find\_mother("Olga", "Kirill") - знание в БЗ  результат: успех  подстановка: {MothersGranny="Olga", X="Kirill"} | Преобразование резольвенты (замена текущей цели на **пустое** тело факта применение полученной подстановки),  Прямой ход |
| 5 | **find\_mother("Kirill", "Jenny")**,  find\_mother(\_, Y),  find\_father(Y, "Jenny"),  find\_father(\_, Z),  find\_father(Z, "Jenny"),  find\_father(\_,  find\_mother(K, "Jenny").  запуск редукции **find\_mother("Kirill", "Jenny")** | Попытка унификации:  ТЦ:find\_mother("Kirill", "Jenny"),  И знаний БЗ, пока не будет найден подходящее знание или не будет «просмотрена» вся БЗ. | Прямой ход, переход к следующему предложению, по окончании был достигнут конец БЗ, термов, при которых унификации завершилась успехом найдено не было, резольвента не пуста. Откат. |
| 6 | **find\_mother(MothersGranny, X)**,  find\_mother(X, "Jenny"),  find\_mother(\_, Y),  find\_father(Y, "Jenny"),  find\_father(\_, Z),  find\_father(Z, "Jenny"),  find\_father(\_,  find\_mother(K, "Jenny").  запуск редукции **find\_mother(MothersGranny, X)** | Попытка унификации:  ТЦ:find\_mother(MothersGranny, X)  find\_mother("Katya", "Victoria") - знание в БЗ  результат: успех  подстановка: {MothersGranny="Katya", X="Victoria"} | Преобразование резольвенты (замена текущей цели на **пустое** тело факта применение полученной подстановки),  Прямой ход |
| 7 | **find\_mother("Victoria", "Jenny"),**  find\_mother(\_, Y),  find\_father(Y, "Jenny"),  find\_father(\_, Z),  find\_father(Z, "Jenny"),  find\_father(\_,  find\_mother(K, "Jenny").  запуск редукции **find\_mother("Anna", "Jenny")** | Попытка унификации:  ТЦ:find\_mother("Victoria", "Jenny"),  И знаний БЗ, пока не будет найден подходящее знание или не будет «просмотрена» вся БЗ. | Прямой ход, переход к следующему предложению, по окончании был достигнут конец БЗ, термов, при которых унификации завершилась успехом найдено не было, резольвента не пуста. Откат. |
| 8 | **find\_mother(MothersGranny, X)**,  find\_mother(X, "Jenny"),  find\_mother(\_, Y),  find\_father(Y, "Jenny"),  find\_father(\_, Z),  find\_father(Z, "Jenny"),  find\_father(\_,  find\_mother(K, "Jenny").  Выбор верхней цели, запуск редукции | Попытка унификации:  ТЦ:find\_mother(MothersGranny, X)  find\_mother ("Sasha", "Anna") - знание в БЗ  результат: успех  подстановка: {MothersGranny="Sasha", X="Anna"} | Преобразование резольвенты (замена текущей цели на **пустое** тело факта применение полученной подстановки),  Прямой ход |
| 9 | **find\_mother("Anna", "Jenny")**,  find\_mother(\_, Y),  find\_father(Y, "Jenny"),  find\_father(\_, Z),  find\_father(Z, "Jenny"),  find\_father(\_,  find\_mother(K, "Jenny").  запуск редукции для **find\_mother("Anna", "Jenny")** | Попытка унификации:  ТЦ:find\_mother ("Anna", "Jenny"),  И знаний БЗ, пока не будет найден подходящее знание или не будет «просмотрена» вся БЗ. | Прямой ход, переход к следующему предложению. |
|  | **find\_mother("Anna", "Jenny")**,  find\_mother(\_, Y),  find\_father(Y, "Jenny"),  find\_father(\_, Z),  find\_father(Z, "Jenny"),  find\_father(\_,  find\_mother(K, "Jenny"). | Попытка унификации:  ТЦ:find\_mother ("Anna", "Jenny"),  find\_mother("Anna", "Jenny") - знание в БЗ  результат: успех | Прямой ход, переход к следующему предложению |
|  | Далее аналогичным способом будут просматриваться следующие цели, поскольку из всего набора нас интересовала переменная MothersGranny, таблица расписана до нахождения искомого значения. Далее программа будет выполнять последовательно цели, поа резольвента не станет пустой, после этого будет откат и проверка на наличие альтернативных решений. | | |