



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 №14 по курсу "Функциональное и логическое программирование"

Тема Использование правил в программе на Prolog

Студент Ковалец К. Э.

Группа ИУ7-63Б

Преподаватели Толпинская Н. Б., Строганов Ю. В.

Москва — 2022 г.

1 Условие задания

Создать базу знаний: «ПРЕДКИ», позволяющую наиболее эффективным способом (за меньшее количество шагов, что обеспечивается меньшим количеством предложений БЗ – правил), и 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.

2 Листинг программы

```
1 DOMAINS
2     name      = string.
3     gender    = symbol.
4     struct_parent = parent(name, gender).
5
6     child      = string.
7     grandma_mother = string.
8     grandpa_mother = string.
9     grandma_father = string.
10    grandpa_father = string.
11
12 PREDICATES
13     is_parent(struct_parent, name).
14     ancestors(child, grandma_mother, grandpa_mother, grandma_father,
15               grandpa_father).
16
17 CLAUSES
18     % дети
19     is_parent(parent("Sergey", m), "Anna").
20     is_parent(parent("Irina", w), "Anna").
21
22     % родители
23     is_parent(parent("Peter", m), "Sergey").
24     is_parent(parent("Maria", w), "Sergey").
25
26     is_parent(parent("Dmitry", m), "Irina").
27     is_parent(parent("Julia", w), "Irina").
28
29     % бабушки и дедушки
30     is_parent(parent("Vadimir", m), "Peter").
31     is_parent(parent("Daria", w), "Peter").
32
33     is_parent(parent("Anton", m), "Maria").
34     is_parent(parent("Natalia", w), "Maria").
35
36     is_parent(parent("Ivan", m), "Dmitry").
37     is_parent(parent("Elizabeth", w), "Dmitry").
38
39     is_parent(parent("Mikhail", m), "Julia").
40     is_parent(parent("Marina", w), "Julia").
41
42     % правило
43     ancestors(Child, GrandmaMother, GrandpaMother, GrandmaFather,
44               GrandpaFather) :-
45         is_parent(parent(Father, m), Child),
46         is_parent(parent(Mother, w), Child),
```

```

45     is_parent(parent(GrandpaMother, m), Mother),
46     is_parent(parent(GrandmaMother, w), Mother),
47     is_parent(parent(GrandpaFather, m), Father),
48     is_parent(parent(GrandmaFather, w), Father).
49
50 GOAL
51     % вопрос 1
52     % ancestors("Anna", GrandmaMother, _, GrandmaFather, _).
53
54     % вопрос 2
55     % ancestors("Anna", _, GrandpaMother, _, GrandpaFather).
56
57     % вопрос 3
58     ancestors("Anna", GrandmaMother, GrandpaMother, GrandmaFather,
59               GrandpaFather).
60
61     % вопрос 4
62     % ancestors("Anna", GrandmaMother, _, _, _).
63
64     % вопрос 5
65     % ancestors("Anna", GrandmaMother, GrandpaMother, _, _).

```

GrandmaMother	GrandpaMother	GrandmaFather	GrandpaFather	
"Julia"	"Dmitry"	"Maria"	"Peter"	1
Next 10 100 1,000 Stop				
?- ancestors("Anna", GrandmaMother, GrandpaMother, GrandmaFather, GrandpaFather).				

Рисунок 2.1 – Результат работы программы для 3 вопроса

3 Таблица для Лабораторной работы №14

Вопрос: `ancestors("Anna", GrandmaMother, _, _, _)`.

№ шага	Сравнение термы, результат, подстановка, если есть	Дальнейшие действия, прямой ход или откат (к чему приводит?)
0		Состояние резольвенты: <code>ancestors("Anna", GrandmaMother, _, _, _)</code>
1	Сравнение: <code>ancestors("Anna", GrandmaMother, _, _, _) == is_parent(parent("Sergey", m), "Anna")</code> Унификация: неуспешно (несовпадение функторов)	Прямой ход Переход к следующему предложению
2-14	—//—	—//—
15	Сравнение: <code>ancestors("Anna", GrandmaMother, _, _, _) == ancestors(Childe, GrandmaMother, GrandpaMother, GrandmaFather, GrandpaFather)</code> Унификация: успешно Подстановка: <code>{Childe = "Anna", GrandmaMother = GrandmaMother, GrandpaMother = _, GrandmaFather = _, GrandpaFather = _}</code>	Образование новой резольвенты: 1. Редукция верхней подцели: замена <code>ancestors("Anna", GrandmaMother, _, _, _)</code> телом найденного правила Получена конъюнкция целей: <code>is_parent(parent(Father, m), Child), is_parent(parent(Mother, w), Child), is_parent(parent(GrandpaMother, m), Mother), is_parent(parent(GrandmaMother, w), Mother), is_parent(parent(GrandpaFather, m), Father), is_parent(parent(GrandmaFather, w), Father).</code> 2. Применение подстановки к полученной конъюнкции целей. Новое состояние резольвенты: <code>is_parent(parent(Father, m), "Anna"), is_parent(parent(Mother, w), "Anna"), is_parent(parent(_, m), Mother), is_parent(parent(GrandmaMother, w), Mother), is_parent(parent(_, m), Father), is_parent(parent(_, w), Father).</code>
16	Сравнение: <code>is_parent(parent(Father, m), "Anna") == is_parent(parent("Sergey", m), "Anna")</code> Унификация: успешна	Образование новой резольвенты: 1. Редукция верхней подцели: удаление <code>is_parent(parent(Father, m), "Anna")</code> так как найденное правило – факт.

	<p>Подстановка: $\{Father = "Sergey"\}$</p>	<p>Получена конъюнкция целей: $is_parent(parent(Mother, w), "Anna"),$ $is_parent(parent(_, m), Mother),$ $is_parent(parent(GrandmaMother, w),$ $Mother),$ $is_parent(parent(_, m), "Sergey"),$ $is_parent(parent(_, w), "Sergey").$</p> <p>2. Применение подстановки к полученной конъюнкции целей.</p> <p>Новое состояние резольвенты: $is_parent(parent(Mother, w), "Anna"),$ $is_parent(parent(_, m), Mother),$ $is_parent(parent(GrandmaMother, w),$ $Mother),$ $is_parent(parent(_, m), Father),$ $is_parent(parent(_, w), Father).$</p>
17	<p>Сравнение: $is_parent(parent(Mother, w), "Anna") ==$ $is_parent(parent("Sergey", m), "Anna")$</p> <p>Унификация: неуспешно (несовпадение термов)</p>	<p>Прямой ход Переход к следующему предложению</p>
18	<p>Сравнение: $is_parent(parent(Mother, w), "Anna") ==$ $is_parent(parent("Irina", w), "Anna")$</p> <p>Унификация: успешна</p> <p>Подстановка: $\{Mother = "Irina"\}$</p>	<p>Образование новой резольвенты:</p> <p>1. Редукция верхней подцели: удаление $is_parent(parent(Mother, w), "Anna")$ так как найденное правило – факт.</p> <p>Получена конъюнкция целей: $is_parent(parent(_, m), Mother),$ $is_parent(parent(GrandmaMother, w),$ $Mother),$ $is_parent(parent(_, m), "Sergey"),$ $is_parent(parent(_, w), "Sergey").$</p> <p>2. Применение подстановки к полученной конъюнкции целей.</p> <p>Новое состояние резольвенты: $is_parent(parent(_, m), "Irina"),$ $is_parent(parent(GrandmaMother, w),$ $"Irina"),$ $is_parent(parent(_, m), "Sergey"),$ $is_parent(parent(_, w), "Sergey").$</p>
19	<p>Сравнение: $is_parent(parent(_, m), "Irina") ==$ $is_parent(parent("Sergey", m), "Anna")$</p>	<p>Прямой ход Переход к следующему предложению</p>

	Унификация: неуспешно (несовпадение термов)	
20-23	——//——	——//——
24	<p>Сравнение: $is_parent(parent(_, m), "Irina") == is_parent(parent("Dmitry", m), "Irina")$</p> <p>Унификация: успешна</p> <p>Подстановка: $\{_ = "Dmitry"\}$</p>	<p>Образование новой резольвенты:</p> <p>1. Редукция верхней подцели: удаление $is_parent(parent(_, m), "Irina")$ так как найденное правило – факт.</p> <p>Получена конъюнкция целей: $is_parent(parent(GrandmaMother, w), Mother),$ $is_parent(parent(_, m), "Sergey"),$ $is_parent(parent(_, w), "Sergey").$</p> <p>2. Применение подстановки к полученной конъюнкции целей.</p> <p>Новое состояние резольвенты: $is_parent(parent(GrandmaMother, w), "Irina"),$ $is_parent(parent(_, m), "Sergey"),$ $is_parent(parent(_, w), "Sergey").$</p>
25	<p>Сравнение: $is_parent(parent(GrandmaMother, w), "Irina") == is_parent(parent("Sergey", m), "Anna")$</p> <p>Унификация: неуспешно (несовпадение термов)</p>	<p>Прямой ход Переход к следующему предложению</p>
26-29	——//——	——//——
30	<p>Сравнение: $is_parent(parent(GrandmaMother, w), "Irina") == is_parent(parent("Julia", w), "Irina")$</p> <p>Унификация: успешна</p> <p>Подстановка: $\{GrandmaMother = "Julia"\}$</p>	<p>Образование новой резольвенты:</p> <p>1. Редукция верхней подцели: удаление $is_parent(parent(GrandmaMother, w), "Irina")$ так как найденное правило – факт.</p> <p>Получена конъюнкция целей: $is_parent(parent(_, m), "Sergey"),$ $is_parent(parent(_, w), "Sergey").$</p> <p>2. Применение подстановки к полученной конъюнкции целей.</p> <p>Новое состояние резольвенты: $is_parent(parent(_, m), "Sergey"),$ $is_parent(parent(_, w), "Sergey").$</p>

31	<p>Сравнение: $is_parent(parent(_, m), "Sergey") == is_parent(parent("Sergey", m), "Anna")$</p> <p>Унификация: неуспешно (несовпадение термов)</p>	<p>Прямой ход Переход к следующему предложению</p>
32	—//—	—//—
33	<p>Сравнение: $is_parent(parent(_, m), "Sergey") == is_parent(parent("Peter", m), "Sergey")$</p> <p>Унификация: успешна</p> <p>Подстановка: $\{_ = "Peter"\}$</p>	<p>Образование новой резольвенты:</p> <p>1. Редукция верхней подцели: удаление $is_parent(parent(_, m), "Sergey")$ так как найденное правило – факт.</p> <p>Получена конъюнкция целей: $is_parent(parent(_, w), "Sergey")$.</p> <p>2. Применение подстановки к полученной конъюнкции целей.</p> <p>Новое состояние резольвенты: $is_parent(parent(_, w), "Sergey")$.</p>
34	<p>Сравнение: $is_parent(parent(_, w), "Sergey") == is_parent(parent("Sergey", m), "Anna")$</p> <p>Унификация: неуспешно (несовпадение термов)</p>	<p>Прямой ход Переход к следующему предложению</p>
35-36	—//—	—//—
37	<p>Сравнение: $is_parent(parent(_, w), "Sergey") == is_parent(parent("Maria", w), "Sergey")$</p> <p>Унификация: успешна</p> <p>Подстановка: $\{_ = "Maria"\}$</p>	<p>Образование новой резольвенты:</p> <p>1. Редукция верхней подцели: удаление $is_parent(parent(_, w), "Sergey")$ так как найденное правило – факт.</p> <p>Новое состояние резольвенты: пуста</p> <p>Решение найдено: форматирование подстановки в качестве побочного эффекта: $\{GrandmaMother = "Julia"\}$ Система должна найти все возможные ответы.</p>
		<p>Обратная трассировка:</p> <p>1) Отмена крайней редукции (шаг 37) 2) Восстановление предыдущего состояния резольвенты (шаг 33): $is_parent(parent(_, w), "Sergey")$</p>

		<p>3) Реконкретизация переменных с шага 37: $\{ _ = "Maria" \}$</p> <p>Переход к следующему предложению относительно шага 37.</p>
38-48	——//——	——//——
49		<p>Конец БЗ</p> <p>Обратная трассировка:</p> <p>1) Отмена крайней редукции (шаг 33)</p> <p>2) Восстановление предыдущего состояния резольвенты (шаг 30): $is_parent(parent(_, m), "Sergey"),$ $is_parent(parent(_, w), "Sergey").$</p> <p>3) Реконкретизация переменных с шага 33: $\{ _ = "Peter" \}$</p> <p>Переход к следующему предложению относительно шага 33.</p>
Решения далее найдены не будут, в итоге система через несколько восстановлений восстановит резольвенту до шага 0		
103		<p>Конец БЗ</p> <p>Обратная трассировка:</p> <p>1) Отмена крайней редукции (шаг 0)</p> <p>2) Восстановление предыдущего состояния резольвенты: резольвента пуста.</p> <p>Завершение работы. На вопрос удалось ответить утвердительно, поэтому в качестве побочного эффекта была возвращена 1 подстановка.</p>