



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»

Дисциплина: «Функциональное и логическое
программирование»

Лабораторная работа №15

Студент: Левушкин И. К.

Группа: ИУ7-62Б

Преподаватели: Толпинская Н. Б.,

Строганов Ю. В.

Москва, 2020 г.

Цель работы

Изучить структуру, особенности и принципы оформления программы, и способ выполнения программы на Prolog.

Задачи работы

Приобрести навыки декларативного описания предметной области с использованием фактов, правил и некоторых специальных разделов программы. Изучить порядок использования фактов и правил в программе на Prolog, принципы и особенности сопоставления и отождествления термов, на основе механизма унификации.

Задание

Создать базу знаний **«Собственники»**, дополнив базу знаний, хранящую знания (лаб. 13):

- **«Телефонный справочник»**: Фамилия, №тел, Адрес – структура (Город, Улица, №дома, №кв),
- **«Автомобили»**: Фамилия_владельца, Марка, Цвет, **Стоимость**, и др.,
- **«Вкладчики банков»**: Фамилия, Банк, счет, сумма, др.,

знаниями о дополнительной **собственности** владельца. **Преобразовать** знания об автомобиле к форме знаний о собственности. Вид собственности (кроме автомобиля):

- **Строение, стоимость** и другие его характеристики;
- **Участок, стоимость** и другие его характеристики;
- **Водный транспорт, стоимость** и другие его характеристики.

Описать и использовать вариантный домен: **Собственность**. Владелец может иметь, но **только один** объект **каждого вида собственности** (это касается и bf автомобиля), или не иметь некоторых видов собственности.

Используя **конъюнктивное правило и разные формы задания одного вопроса (пояснять для какого №задания – какой вопрос)**, обеспечить возможность поиска:

1. Названий всех объектов собственности заданного субъекта,

2. Названий и стоимости всех объектов собственности заданного субъекта,
3. Разработать правило, позволяющее найти суммарную стоимость всех объектов собственности заданного субъекта.

Для 2-го пункт и **одной** фамилии **составить таблицу**, отражающую конкретный порядок работы системы, с объяснениями порядка работы и особенностей использования доменов (указать конкретные T1 и T2 и полную подстановку на каждом шаге)

| № ша-га | Сравниваемые термы; результат; подстановка, если есть | Дальнейшие действия: прямой ход или откат (к чему приводит?) |
|---------|--|--|
| 1... | -попытка унификации: T1=T2 ... -результат: Успех и подстановка, или Нет | Комментарий, вывод... |
| 2 | ... | ... |
| | | |

При желании, можно усложнить свою базу знаний, введя варианты: **строение**: (Дом, офис, торговый центр), **участок**: (садовый, территория под застройку, территория под агро-работы), **Водный транспорт**: варианты названий.

Реализация программы

domains

```
town, street, sername, model,  
watercraft, color, bank, area, building, own = symbol  
phone = string  
house, case, cost, probeg, account, money = integer  
address = adr(town, street, house, case)  
ownership = building_owner(building, cost) ; area_owner(area, cost)  
; watercraft_owner(watercraft, cost)  
; car_owner(model, color, cost, probeg)  
predicates
```

```
phone_list(sername, phone, address).  
auto(sername, model, color, cost, probeg).  
bank_list(sername, bank, account, money).
```

```
owner(sername, ownership).
```

```
get_own_cost(sername, own, cost).
```

```
get_sum_cost_owner(sername, cost).  
clauses
```

```
get_own_cost(Sername, building, Cost) :-  
owner(Sername, building_owner(_, Cost)),  
!.
```

```
get_own_cost(Sername, area, Cost) :-  
owner(Sername, area_owner(_, Cost)),  
!.
```

```
get_own_cost(Sername, watercraft, Cost) :-  
owner(Sername, watercraft_owner(_, Cost)),  
!.
```

```
get_own_cost(Sername, car, Cost) :-  
owner(Sername, car_owner(_, _, Cost, _)),  
!.
```

```
get_own_cost(_, _, Cost):-  
Cost = 0.
```

```
get_sum_cost_owner(Sername, Cost) :-  
get_own_cost(Sername, building, Building_Cost),  
get_own_cost(Sername, area, Area_Cost),  
get_own_cost(Sername, watercraft, Watercraft_Cost),  
get_own_cost(Sername, car, Car_Cost),  
Cost = Building_Cost + Area_Cost + Watercraft_Cost + Car_Cost.
```

```
owner(Sername, car_owner(Model, Color, Cost, Probeg)) :-  
auto(Sername, Model, Color, Cost, Probeg).
```

```
owner(levushkin, building_owner(dacha, 90000)).  
owner(levushkin, area_owner(dacha_area, 9000)).  
owner(levushkin, watercraft_owner(boat, 90)).
```

```
owner(samkov, building_owner(fitness_club, 888888)).  
owner(samkov, area_owner(park, 89898989)).
```

```
owner(ryazanova, building_owner(moscow_city, 987654321)).
```

```
phone_list(levushkin, "89859771492", adr(moscow, kantemirovskaya, 5, 1)).  
phone_list(samkov, "898999999", adr(chelyabinsk, pushkinskaya, 4, 2)).  
phone_list(ryazanova, "8911911911", adr(moscow, baumanskaya, 9, 9)).
```

```
auto(levushkin, volvo, grey, 3000000, 1000).  
auto(samkov, volkswagen, pink, 1000000, 99999).  
auto(ryazanova, bugatti, gold, 999999999, 1).
```

```
bank_list(levushkin, sberbank, 1111, 900000).  
bank_list(samkov, sberbank, 2222, 100).  
bank_list(ryazanova, tinkoff, 3333, 99999999).  
bank_list(ryazanova, raiffeisen, 4444, 888888888).
```

Используя конъюнктивное правило и разные формы задания одного вопроса, обеспечить возможность поиска

1. Названия всех объектов собственности по заданному субъекту.

Пример 1:

goal

```
owner(levushkin, area_owner(Area, _));  
owner(levushkin, building_owner(Building, _));  
owner(levushkin, watercraft_owner(Watercraft, _));  
owner(levushkin, car_owner(Car, _, _, _)).
```

%Вывод:

```
Area=dacha_area  
Building=dacha  
Watercraft=boat  
Car=volvo  
4 Solutions
```

Пример 2:

goal

```
owner(samkov, area_owner(Area, _));  
owner(samkov, building_owner(Building, _));  
owner(samkov, watercraft_owner(Watercraft, _));  
owner(samkov, car_owner(Car, _, _, _)).
```

%Вывод:

```
Area=park  
Building=fitness_club  
Car=volkswagen  
3 Solutions
```

Пример 3:

goal

```
owner(ryazanova, area_owner(Area, _));  
owner(ryazanova, building_owner(Building, _));  
owner(ryazanova, watercraft_owner(Watercraft, _));  
owner(ryazanova, car_owner(Car, _, _, _)).
```

%Вывод:

```
Building=moscow_city  
Car=bugatti  
2 Solutions
```

2. Названия и стоимости всех объектов собственности по заданному субъекту.

Пример 1:

```
goal
owner(levushkin, area_owner(Area, Area_Cost));
owner(levushkin, building_owner(Building, Building_Cost));
owner(levushkin, watercraft_owner(Watercraft, Watercraft_Cost));
owner(levushkin, car_owner(Car, _, Car_Cost, _)).
%Вывод:
Area=dacha_area, Area_Cost=9000
Building=dacha, Building_Cost=90000
Watercraft=boat, Watercraft_Cost=90
Car=volvo, Car_Cost=3000000
4 Solutions
```

Пример 2:

```
goal
owner(samkov, area_owner(Area, Area_Cost));
owner(samkov, building_owner(Building, Building_Cost));
owner(samkov, watercraft_owner(Watercraft, Watercraft_Cost));
owner(samkov, car_owner(Car, _, Car_Cost, _)).
%Вывод:
Area=park, Area_Cost=89898989
Building=fitness_club, Building_Cost=888888
Car=volkswagen, Car_Cost=1000000
3 Solutions
```

Пример 3:

```
goal
owner(ryazanova, area_owner(Area, Area_Cost));
owner(ryazanova, building_owner(Building, Building_Cost));
owner(ryazanova, watercraft_owner(Watercraft, Watercraft_Cost));
owner(ryazanova, car_owner(Car, _, Car_Cost, _)).
%Вывод:
Building=moscow_city, Building_Cost=987654321
Car=bugatti, Car_Cost=999999999
2 Solutions
```

Порядок работы системы на 3 примере:

| № ша-га | Сравниваемые термы; результат; подстановка, если есть | Дальнейшие действия: прямой ход или откат (к чему приводит?) |
|---------|---|---|
| 1 | T1 = owner(ryazanova, area_owner(Area, Area_Cost)); T2 = get_own_cost(Sername, building, Cost). Неудача (функторы owner и get_own_cost не равны). | Прямой ход к следующему предложению. Аналогичная ситуация в следующих 5 предложениях. Прямой ход к следующему предложению |
| 7 | T1 = owner(ryazanova, area_owner(Area, Area_Cost)); T2 = owner(Sername, car_owner(Model, Color, Cost, Probeg)). Неудача (функторы area_owner и car_owner не равны). | Прямой ход к следующему предложению. |
| 8 | T1 = owner(ryazanova, area_owner(Area, Area_Cost)); T2 = owner(levushkin, building_owner(dacha, 90000)). Неудача (ryazanova != levushkin). | Прямой ход к следующему предложению. |
| 9 | T1 = owner(ryazanova, area_owner(Area, Area_Cost)); T2 = owner(levushkin, area_owner(dacha_area, 9000)). Неудача (ryazanova != levushkin) | Прямой ход к следующему предложению. |
| 10 | T1 = owner(ryazanova, area_owner(Area, Area_Cost)); T2 = owner(levushkin, warercraft_owner(boat, 90)). Неудача (ryazanova != levushkin) | Прямой ход к следующему предложению. |
| 11 | T1 = owner(ryazanova, area_owner(Area, Area_Cost)); T2 = owner(samkov, building_owner(fitness_club, 888888)). Неудача (ryazanova != samkov). | Прямой ход к следующему предложению. |
| 12 | T1 = owner(ryazanova, area_owner(Area, Area_Cost)); T2 = owner(samkov, area_owner(park, 89898989)). Неудача (ryazanova != samkov). | Прямой ход к следующему предложению. |

| | | |
|----|--|---|
| 13 | <p>T1 = owner(ryazanova, area_owner(Area, Area_Cost));</p> <p>T2 = owner(ryazanova, building_owner(moscow_city, 987654321)).</p> <p>Неудача (функторы area_owner и building_owner не равны).</p> | Прямой ход к следующему предложению. |
| 14 | <p>T1 = owner(ryazanova, area_owner(Area, Area_Cost));</p> <p>T2 = phone_list(levushkin, «89859771492», adr(moscow, kantemirovskaya, 5, 1)).</p> <p>Неудача (функторы owner и phone_list не равны).</p> | Прямой ход к следующему предложению. Аналогичная ситуация в следующих 9 предложениях (phone_list, auto, bank_list). Откат, переход к предыдущему состоянию резольвенты. Поиск с начала предложений. |
| 24 | <p>T1 = owner(ryazanova, building_owner(Building, Building_Cost));</p> <p>T2 = get_own_cost(Sername, building, Cost).</p> <p>Неудача (функторы owner и get_own_cost не равны).</p> | Прямой ход к следующему предложению. Аналогичная ситуация в следующих 5 предложениях. Прямой ход к следующему предложению |
| 30 | <p>T1 = owner(ryazanova, building_owner(Building, Building_Cost));</p> <p>T2 = owner(Sername, car_owner(Model, Color, Cost, Probeg)).</p> <p>Неудача (функторы building_owner и car_owner не равны).</p> | Прямой ход к следующему предложению. |
| 31 | <p>T1 = owner(ryazanova, building_owner(Building, Building_Cost));</p> <p>T2 = owner(levushkin, building_owner(dacha, 90000)).</p> <p>Неудача (ryazanova != levushkin).</p> | Прямой ход к следующему предложению. |
| 32 | <p>T1 = owner(ryazanova, building_owner(Building, Building_Cost));</p> <p>T2 = owner(levushkin, area_owner(dacha_area, 9000)).</p> <p>Неудача (ryazanova != levushkin)</p> | Прямой ход к следующему предложению. |

| | | |
|----|--|--|
| 33 | <p>T1 = owner(ryazanova, building_owner(Building, Building_Cost));</p> <p>T2 = owner(levushkin, warercraft_owner(boat, 90)).</p> <p>Неудача (ryazanova != levushkin)</p> | Прямой ход к следующему предложению. |
| 34 | <p>T1 = owner(ryazanova, building_owner(Building, Building_Cost));</p> <p>T2 = owner(samkov, building_owner(fitness_club, 888888)).</p> <p>Неудача (ryazanova != samkov).</p> | Прямой ход к следующему предложению. |
| 35 | <p>T1 = owner(ryazanova, building_owner(Building, Building_Cost));</p> <p>T2 = owner(samkov, area_owner(park, 89898989)).</p> <p>Неудача (ryazanova != samkov).</p> | Прямой ход к следующему предложению. |
| 36 | <p>T1 = owner(ryazanova, building_owner(Building, Building_Cost));</p> <p>T2 = owner(ryazanova, building_owner(moscow_city, 987654321)).</p> <p>Успех. Подсановка Building = moscow_city, Building_Cost = 987654321.</p> | <p>Вывод:</p> <p>Building = moscow_city, Building_Cost = 987654321,</p> <p>Прямой ход к следующему предложению. Реконкретизация Building, Building_Cost.</p> |
| 37 | <p>T1 = owner(ryazanova, building_owner(Building, Building_Cost));</p> <p>T2 = phone_list(levushkin, «89859771492», adr(moscow, kantemirovskaya, 5, 1)).</p> <p>Неудача (функторы owner и phone_list не равны).</p> | <p>Прямой ход к следующему предложению. Аналогичная ситуация в следующих 9 предложениях (phone_list, auto, bank_list). Откат, переход к предыдущему состоянию резольвенты. Поиск с начала предложений.</p> |

| | | |
|----|--|--|
| 47 | <p>T1 = owner(ryazanova, watercraft_owner(Watercraft, Watercraft_Cost));</p> <p>T2 = get_own_cost(Sername, building, Cost).</p> <p>Неудача (функторы owner и get_own_cost не равны).</p> | <p>Прямой ход к следующему предложению. Аналогичная ситуация в следующих 5 предложениях. Прямой ход к следующему предложению</p> |
| 53 | <p>T1 = owner(ryazanova, watercraft_owner(Watercraft, Watercraft_Cost));</p> <p>T2 = owner(Sername, car_owner(Model, Color, Cost, Probeg)).</p> <p>Неудача (функторы watercraft_owner и car_owner не равны).</p> | <p>Прямой ход к следующему предложению.</p> |
| 54 | <p>T1 = owner(ryazanova, watercraft_owner(Watercraft, Watercraft_Cost));</p> <p>T2 = owner(levushkin, building_owner(dacha, 90000)).</p> <p>Неудача (ryazanova != levushkin).</p> | <p>Прямой ход к следующему предложению.</p> |
| 55 | <p>T1 = owner(ryazanova, watercraft_owner(Watercraft, Watercraft_Cost));</p> <p>T2 = owner(levushkin, area_owner(dacha_area, 9000)).</p> <p>Неудача (ryazanova != levushkin)</p> | <p>Прямой ход к следующему предложению.</p> |
| 56 | <p>T1 = owner(ryazanova, watercraft_owner(Watercraft, Watercraft_Cost));</p> <p>T2 = owner(levushkin, warercraft_owner(boat, 90)).</p> <p>Неудача (ryazanova != levushkin)</p> | <p>Прямой ход к следующему предложению.</p> |
| 57 | <p>T1 = owner(ryazanova, watercraft_owner(Watercraft, Watercraft_Cost));</p> <p>T2 = owner(samkov, building_owner(fitness_club, 888888)).</p> <p>Неудача (ryazanova != samkov).</p> | <p>Прямой ход к следующему предложению.</p> |

| | | |
|----|---|---|
| 58 | <p>T1 = owner(ryazanova, watercraft_owner(Watercraft, Watercraft_Cost));</p> <p>T2 = owner(samkov, area_owner(park, 89898989)).</p> <p>Неудача (ryazanova != samkov).</p> | Прямой ход к следующему предложению. |
| 59 | <p>T1 = owner(ryazanova, watercraft_owner(Watercraft, Watercraft_Cost));</p> <p>T2 = owner(ryazanova, building_owner(moscow_city, 987654321)).</p> <p>Неудача (функторы watercraft_owner и building_owner не равны).</p> | Прямой ход к следующему предложению. |
| 60 | <p>T1 = owner(ryazanova, watercraft_owner(Watercraft, Watercraft_Cost));</p> <p>T2 = phone_list(levushkin, «89859771492», adr(moscow, kantemirovskaya, 5, 1)).</p> <p>Неудача (функторы owner и phone_list не равны).</p> | Прямой ход к следующему предложению. Аналогичная ситуация в следующих 9 предложениях (phone_list, auto, bank_list). Откат, переход к предыдущему состоянию резольвенты. Поиск с начала предложений. |
| 70 | <p>T1 = owner(ryazanova, car_owner(Car, _, Car_Cost, _));</p> <p>T2 = get_own_cost(Sername, building, Cost).</p> <p>Неудача (функторы owner и get_own_cost не равны).</p> | Прямой ход к следующему предложению. Аналогичная ситуация в следующих 5 предложениях. Прямой ход к следующему предложению |
| 76 | <p>T1 = owner(ryazanova, car_owner(Car, _, Car_Cost, _));</p> <p>T2 = owner(Sername, car_owner(Model, Color, Cost, Probeg)).</p> <p>Успех. Подстановка ryazanova = Sername, Car = Model, Car_Cost = Cost.</p> | Прямой ход к auto(ryazanova, Car, _, Car_Cost, _). Поиск с начала предложений. |

| | | |
|-----|---|--|
| 77 | <p>T1 = auto(ryazanova, Car, _, Car_Cost, _);</p> <p>T2 = get_own_cost(Sername, building, Cost).</p> <p>Неудача (функторы auto и get_own_cost не равны).</p> | <p>Прямой ход к следующему предложению. Аналогичная ситуация в следующих 15 предложениях (get_own_cost, get_sum_cost_owner, owner, phone_list). Прямой ход к следующему предложению</p> |
| 93 | <p>T1 = auto(ryazanova, Car, _, Car_Cost, _);</p> <p>T2 = auto(levushkin, volvo, grey, 3000000, 1000).</p> <p>Неудача (ryazanova != levushkin).</p> | <p>Прямой ход к следующему предложению.</p> |
| 94 | <p>T1 = auto(ryazanova, Car, _, Car_Cost, _);</p> <p>T2 = auto(samkov, volkswagen, pink, 1000000, 99999).</p> <p>Неудача (ryazanova != samkov).</p> | <p>Прямой ход к следующему предложению.</p> |
| 95 | <p>T1 = auto(ryazanova, Car, _, Car_Cost, _);</p> <p>T2 = auto(ryazanova, bugatti, gold, 999999999, 1).</p> <p>Успех. Подстановка ryazanova = ryazanova, Car = bugatti, Car_Cost = 999999999.</p> | <p>Вывод: Car = bugatti, Car_Cost = 999999999. Прямой ход к следующему предложению, реконкретизация Car, Car_Cost.</p> |
| 96 | <p>T1 = auto(ryazanova, Car, _, Car_Cost, _);</p> <p>T2 = bank_list(levushkin, sberbank, 1111, 900000).</p> <p>Неудача (функторы auto и bank_list не равны).</p> | <p>Прямой ход к следующему предложению. Аналогичная ситуация в следующих 3 предложениях. Откат, переход к предыдущему состоянию резольвенты - owner(ryazanova, car_owner(Car, _, Car_Cost, _)), реконкретизация Car, Car_Cost.</p> |
| 100 | <p>T1 = owner(ryazanova, car_owner(Car, _, Car_Cost, _));</p> <p>T2 = owner(levushkin, building_owner(dacha, 90000).</p> <p>Неудача (ryazanova != levushkin).)</p> | <p>Прямой ход к следующему предложению. Аналогичная ситуация в следующих 2 предложениях. Прямой ход к следующему предложению.</p> |

| | | |
|-----|--|---|
| 103 | <p>T1 = owner(ryazanova, car_owner(Car, _, Car_Cost, _));</p> <p>T2 = owner(samkov, building_owner(fitness_club, 888888)).</p> <p>Неудача (ryazanova != samkov).</p> | <p>Прямой ход к следующему предложению. Аналогичная ситуация в следующем предложении. Прямой ход к следующему предложению.</p> |
| 105 | <p>T1 = owner(ryazanova, car_owner(Car, _, Car_Cost, _));</p> <p>T2 = owner(ryazanova, building_owner(moscow_city, 987654321)).</p> <p>Неудача (функторы car_owner и building_owner не равны).</p> | <p>Прямой ход к следующему предложению.</p> |
| 106 | <p>T1 = owner(ryazanova, car_owner(Car, _, Car_Cost, _));</p> <p>T2 = phone_list(levushkin, «89859771492», adr(moscow, kantemirovskaya, 5, 1)).</p> <p>Неудача (функторы owner и phone_list не равны).</p> | <p>Прямой ход к следующему предложению. Аналогичная ситуация в следующих 9 предложениях (phone_list, auto, bank_list). Откат, переход к предыдущему состоянию резольвенты. Конец clauses. Опустошение резольвенты. Завершение работы (115 шагов).</p> |
| | | |

3. Разработать правило, позволяющее найти суммарную стоимость всех объектов собственности заданного субъекта.

Правило:

```

get_sum_cost_owner(Sername, Cost) :-
  get_own_cost(Sername, building, Building_Cost),
  get_own_cost(Sername, area, Area_Cost),
  get_own_cost(Sername, watercraft, Watercraft_Cost),
  get_own_cost(Sername, car, Car_Cost),
  Cost = Building_Cost + Area_Cost + Watercraft_Cost + Car_Cost.

```

Правила для предиката get_own_cost:

```

get_own_cost(Sername, building, Cost) :-
  owner(Sername, building_owner(_, Cost)),

```

!.

```
get_own_cost(Sername, area, Cost) :-  
owner(Sername, area_owner(_, Cost)),  
!.
```

```
get_own_cost(Sername, watercraft, Cost) :-  
owner(Sername, watercraft_owner(_, Cost)),  
!.
```

```
get_own_cost(Sername, car, Cost) :-  
owner(Sername, car_owner(_, _, Cost, _)),  
!.
```

```
get_own_cost(_, _, Cost) :-  
Cost = 0.
```

Тестирование правила:

Пример 1:

```
goal  
    get_sum_cost_owner(levushkin, Cost).  
%Вывод:  
Cost=3099090  
1 Solution
```

Пример 2:

```
goal  
    get_sum_cost_owner(samkov, Cost).  
%Вывод:  
Cost=91787877  
1 Solution
```

Пример 3:

```
goal  
    get_sum_cost_owner(ryazanova, Cost).  
%Вывод:  
Cost=1987654320  
1 Solution
```

Ответы на вопросы

В каком фрагменте программы сформулировано знание? Это знание о чем на формальном уровне?

Знания размещены в CLAUSES в виде предложений.

Они представляют собой знания о некоторой предметной области, формально – отношения между различными объектами.

Что содержит тело правила?

Тело правила содержит условие истинности заголовка правила.

Что дает использование переменных при формулировании знаний? В чем отличие формулировки знания с помощью термов с одинаковой арностью при использовании одной переменной и при использовании нескольких переменных?

Использование переменных в формулировании знаний позволяют уточнять значения и переносить их «в пространстве и времени».

Формулировка знаний с использованием переменных носит более общий характер по отношению к знанию, состоящему только лишь из констант.

Использование знаний с одинаковой арностью при использовании одной переменной носит менее общий характер по отношению к знанию с использованием нескольких переменных.

С каким квантором переменные входят в правило, в каких пределах переменная уникальна?

Переменные входят в правило с квантором всеобщности («для любого»).

Переменные уникальны в пределах предложения.

Исключение – анонимные переменные – каждая такая переменная является отдельной сущностью и применяется, когда ее значение неважно для данного предложения.

Какова семантика (смысл) предложений раздела DOMAINS? Когда, где и с какой целью используется это описание?

Предложения в разделе DOMAINS используются для объявления используемых доменов, не являющимися стандартными доменами в Prolog.

Раздел доменов используется для описания структур (вариантных доменов).

Какова семантика (смысл) предложений раздела PREDICATES? Когда, и где используется это описание? С какой целью?

В разделе PREDICATES описываются предикаты, их аридность (местность) и домены (типы и природа аргументов).

С помощью описанных предикатов, можно создавать предложения в базе знаний.

Предикаты используются для представления, как фактов, так и правил.

Унификация каких термов запускается на самом первом шаге работы системы? Каковы назначение и результат использования алгоритма унификации?

На первом шаге работы происходит унификация вопроса и первого предложения базы знаний.

Назначение алгоритма унификации заключается в попарном сопоставлении термов и попытке построить для них общий пример.

Результатом использования алгоритма унификации может быть успех или тупиковая ситуация (неудача).

В каком случае 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.

Исправление ошибок 13-ой лабораторной работы.

В таблице: `get_info_by_phone(«89859771492», Surname, Model, Cost)` и `get_info_by_phone(Phone, Surname, Model, Cost)`. Подстановка `Phone= «89859771492»` Не полная!!!

Ниже приведены все три исправленные таблицы порядка поиска ответа на вопрос, так как данная ошибка затрагивает их всех:

Таблица для *Фамилия, марка авто и его цена по номеру телефона владельца* примера №1:

| № ша-га | Сравниваемые термы; результат; подстановка, если есть | Дальнейшие действия: прямой ход или откат (к чему приводит?) |
|---------|--|--|
| 1 | Сравнение <code>get_info_by_phone(«89859771492», Surname, Model, Cost)</code> и <code>get_info_by_phone(Phone, Surname, Model, Cost)</code> . Успех. Подстановка <code>Phone = «89859771492», Surname = Surname, Model = Model, Cost = Cost</code> | Прямой ход: <code>phone_list(Surname, «89859771492», _)</code> |
| 2 | Сравнение <code>phone_list(Surname, «89859771492», _)</code> и <code>phone_list(levushkin, «89859771492», _)</code> . Успех. Подстановка <code>Surname = levushkin, «89859771492» = «89859771492»</code> | Прямой ход: <code>auto(levushkin, Model, _, Cost, _)</code> |
| 3 | Сравнение <code>auto(levushkin, Model, _, Cost, _)</code> и <code>auto(levushkin, shkoda, _, 600000, _)</code> . Успех. Подстановка <code>levushkin = levushkin, Model = shkoda, Cost = 600000</code> | Вывод: <code>Surname=levushkin, Model=shkoda, Cost=600000.</code> Откат к <code>auto(levushkin, Model, _, Cost, _)</code> |
| 4 | Сравнение <code>auto(levushkin, Model, _, Cost, _)</code> и <code>auto(levushkin, volvo, _, 3000000, _)</code> . Успех. Подстановка <code>levushkin = levushkin, Model = volvo, Cost = 3000000</code> | Вывод: <code>Surname=levushkin, Model=volvo, Cost=3000000.</code> Откат к <code>auto(levushkin, Model, _, Cost, _)</code> |
| 5 | Сравнение <code>auto(levushkin, Model, _, Cost, _)</code> и <code>auto(samkov, volkswagen, _, 1000000, _)</code> . Неудача (<code>levushkin ≠ samkov</code>) | Откат к <code>auto(levushkin, Model, _, Cost, _)</code> |
| 6 | Сравнение <code>auto(levushkin, Model, _, Cost, _)</code> и <code>auto(ryazanova, bugatti, _, 999999999, _)</code> . Неудача (<code>levushkin ≠ ryananova</code>) | Откат к <code>auto(levushkin, Model, _, Cost, _)</code> , откат к <code>phone_list(Surname, «89859771492», _)</code> , неудача в поиске дальнейших подстановок «89859771492», откат к <code>get_info_by_phone</code> , неудача в поиске дальнейщих подстановок <code>get_info_by_phone</code> , завершение унификации. |

Таблица для *Марка авто по номеру телефона владельца* примера №1:

| № ша-га | Сравниваемые термины; результат; подстановка, если есть | Дальнейшие действия: прямой ход или откат (к чему приводит?) |
|---------|---|---|
| 1 | Сравнение <code>get_info_by_phone(«8911911911», _, Model, _)</code> и <code>get_info_by_phone(Phone, Surname, Model, Cost)</code> . Успех. Подстановка <code>Phone = «8911911911», Model = Model</code> . | Прямой ход: <code>phone_list(Surname, «8911911911», _)</code> . |
| 2 | Сравнение <code>phone_list(Surname, «8911911911», _)</code> и <code>phone_list(levushkin, «89859771492», _)</code> . Неудача (<code>«8911911911» ≠ «89859771492»</code>). | Откат к <code>phone_list(Surname, «8911911911», _)</code> . |
| 3 | Сравнение <code>phone_list(Surname, «8911911911», _)</code> и <code>phone_list(levushkin, «89859771493», _)</code> . Неудача (<code>«8911911911» ≠ «89859771493»</code>). | Откат к <code>phone_list(Surname, «8911911911», _)</code> . |
| 4 | Сравнение <code>phone_list(Surname, «8911911911», _)</code> и <code>phone_list(samkov, «89899999», _)</code> . Неудача (<code>«8911911911» ≠ «89899999»</code>). | Откат к <code>phone_list(Surname, «8911911911», _)</code> . |
| 5 | Сравнение <code>phone_list(Surname, «8911911911», _)</code> и <code>phone_list(ryazanova, «8911911911», _)</code> . Успех. Подстановка <code>Surname=ryazanova, «8911911911» = «8911911911»</code> . | Прямой ход: <code>auto(ryazanova, Model, _, _, _)</code> . |
| 6 | Сравнение <code>auto(ryazanova, Model, _, _, _)</code> и <code>auto(levushkin, shkode, _, _, _)</code> . Неудача (<code>ryazanova ≠ levushkin</code>). | Откат к <code>auto(ryazanova, Model, _, _, _)</code> . |
| 7 | Сравнение <code>auto(ryazanova, Model, _, _, _)</code> и <code>auto(levushkin, volvo, _, _, _)</code> . Неудача (<code>ryazanova ≠ levushkin</code>). | Откат к <code>auto(ryazanova, Model, _, _, _)</code> . |
| 8 | Сравнение <code>auto(ryazanova, Model, _, _, _)</code> и <code>auto(samkov, volkswagen, _, _, _)</code> . Неудача (<code>ryazanova ≠ samkov</code>). | Откат к <code>auto(ryazanova, Model, _, _, _)</code> . |
| 9 | Сравнение <code>auto(ryazanova, Model, _, _, _)</code> и <code>auto(ryazanova, bugatti, _, _, _)</code> . Успех. Подстановка <code>ryazanova = ryazanova, Model=bugatti</code> . | Вывод: <code>Model=bugatti</code> . откат к <code>auto(ryazanova, Model, _, _, _)</code> , откат к <code>phone_list</code> , откат к <code>get_info_by_phone</code> , завершение унификации. |

Таблица для *Улица проживания, банки и номера телефона по фамилии и городу* примера №1:

| № ша-га | Сравниваемые термины; результат; подстановка, если есть | Дальнейшие действия: прямой ход или откат (к чему приводит?) |
|---------|--|--|
| 1 | Сравнение <code>get_info_surname_town(levushkin, moscow, Street, Bank, Phone)</code> и <code>get_info_surname_town(Surname, Town, Street, Bank, Phone)</code> . Успех. Подстановка <code>Surname=levushkin, Town=moscow, Street = Street, Bank = Bank, Phone = Phone</code> . | Прямой ход: <code>phone_list(levushkin, Phone, adr(moscow, Street, _, _))</code> . |
| 2 | Сравнение <code>phone_list(levushkin, Phone, adr(moscow, Street, _, _))</code> и <code>phone_list(levushkin, «89859771492», adr(moscow, kantemirovskaya, _, _))</code> . Успех. Подстановка <code>levushkin = levushkin, Phone=«89859771492», moscow = moscow, Street=kantemirovskaya</code> . | Прямой ход: <code>bank_list(levushkin, Bank, _, _)</code> . |
| 3 | Сравнение <code>bank_list(levushkin, Bank, _, _)</code> и <code>bank_list(levushkin, sberbank, _, _)</code> . Успех. Подстановка <code>levushkin = levushkin, Bank=sberbank</code> . | Вывод: <code>Street=kantemirovskaya, Bank=sberbank, Phone=«89859771492»</code> . Откат к <code>bank_list(levushkin, Bank, _, _)</code> , неудача в поиске дальнейших подстановок <code>levushkin</code> , откат к <code>phone_list(levushkin, Phone, adr(moscow, Street, _, _))</code> . |
| 4 | Сравнение <code>phone_list(levushkin, Phone, adr(moscow, Street, _, _))</code> и <code>phone_list(levushkin, «89859771493», adr(moscow, kantemirovskaya, _, _))</code> . Успех. Подстановка <code>levushkin = levushkin, Phone=«89859771493», moscow = moscow, Street = kantemirovskaya</code> . | Прямой ход: <code>bank_list(levushkin, Bank, _, _)</code> . |
| 5 | Сравнение <code>bank_list(levushkin, Bank, _, _)</code> и <code>bank_list(levushkin, sberbank, _, _)</code> . Успех. Подстановка <code>levushkin = levushkin, Bank=sberbank</code> . | Вывод: <code>Street=kantemirovskaya, Bank=sberbank, Phone=«89859771493»</code> . Откат к <code>bank_list(levushkin, Bank, _, _)</code> , неудача в поиске дальнейших подстановок <code>levushkin</code> , откат к <code>phone_list(levushkin, Phone, adr(moscow, Street, _, _))</code> , неудача в поиске дальнейших подстановок <code>levushkin, moscow</code> . Откат к <code>get_info_surname_town</code> , завершение унификации. |