



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 №14

Студент: Левушкин И. К.

Группа: ИУ7-62Б

Преподаватели: Толпинская Н. Б.,

Строганов Ю. В.

Москва, 2020 г.

Цель работы

Получить навыки построения модели предметной области, разработки и оформления программы на Prolog, изучить принципы, логику формирования программы и отдельные шаги выполнения программы на Prolog.

Задачи работы

Приобрести навыки декларативного описания предметной области с использованием фактов и правил. Изучить способы использования термов, переменных, фактов и правил в программе на Prolog, принципы и правила сопоставления и отождествления, порядок унификации.

Задание

Используя 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.

Владелец может иметь несколько телефонов, автомобилей, вкладов (Факты). В разных городах есть однофамильцы, в одном городе – фамилия уникальна.

Используя **конъюнктивное правило и простой вопрос**, обеспечить возможность поиска:

По Марке и Цвету автомобиля найти Фамилию, Город, Телефон и Банки, в которых владелец автомобиля имеет вклады. Лишней информации не находить и не передавать!!!

Владельцев может быть **несколько** (не более 3-х), **один** и **ни одного**.

1. Для каждого из трех вариантов **словесно подробно** описать порядок формирования ответа (в виде таблицы). При этом, указать – отметить моменты очередного запуска алгоритма унификации и полный результат его работы. Обосновать следующий шаг работы системы. Выписать унификаторы – подстановки. Указать моменты, причины и результат отката, если он есть.

2. Для случая нескольких владельцев (2-х): приведите примеры (таблицы) работы системы **при разных порядках** следования в БЗ процедур, и знаний в них: («Телефонный справочник», «Автомобили», «Вкладчики банков», или: «Автомобили», «Вкладчики банков», «Телефонный справочник»). Сделайте вывод: Одинаковы ли: множество работ и объем работ в разных случаях?
3. Оформите 2 таблицы, демонстрирующие **порядок работы алгоритма унификации** вопроса и подходящего заголовка правила (для двух случаев из пункта 2) и укажите результаты его работы: ответ и побочный эффект.

Реализация программы

domains

address = adr(symbol, symbol, integer, integer).

predicates

phone_list(symbol Surname, string Phone, address Address).

auto(symbol Surname, symbol Model, symbol Color, integer Cost,
integer Probeg).

bank_list(symbol Surname, symbol Bank, integer Account, integer Money).

get_info_by_model_color(symbol Model, symbol Color,
symbol Surname, string Phone, symbol Bank).

clauses

phone_list(levushkin, "89859771492", adr(moscow, kantemirovskaya, 5, 1)).

phone_list(samkov, "89899999", adr(chelyabinsk, pushkinskaya, 4, 2)).

phone_list(ryazanova, "8911911911", adr(moscow, baumanskaya, 9, 9)).

auto(levushkin, shkoda, orange, 600000, 10000).

auto(levushkin, volvo, grey, 3000000, 1000).

auto(samkov, volkswagen, pink, 1000000, 99999).

auto(samkov, bugatti, gold, 999999999, 1).

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).

get_info_by_model_color(Model, Color, Surname, Phone, Bank) :-

auto(Surname, Model, Color, _, _),

bank_list(Surname, Bank, _, _),

phone_list(Surname, Phone, _).

Тесты

Фамилия, город, телефонный номер, название банка по марке и цвету авто

1. Несколько вариантов ответа.

goal

```
get_info_by_model_color(bugatti, gold, Surname, Phone, Bank).
```

%Вывод:

```
Surname=samkov, Phone=898999999, Bank=sberbank
```

```
Surname=ryazanova, Phone=8911911911, Bank=tinkoff
```

```
Surname=ryazanova, Phone=8911911911, Bank=raiffeisen
```

```
3 Solutions
```

Порядок поиска ответа на вопрос:

Вопрос будет сопоставляться с каждым предложением сверху вниз, пока не найдется подходящий функтор. В таблице описаны ключевые шаги последующего поиска.

№ шага	Сравниваемые термы; результат; подстановка, если есть	Дальнейшие действия: прямой ход или откат (к чему приводит?)
13	Сравнение <code>get_info_by_model_color(bugatti, gold, Surname, Phone, Bank)</code> с <code>get_info_by_model_color(Model, Color, Surname, Phone, Bank)</code> . Успех. Подстановка <code>Model=bugatti, Color=gold</code> .	Прямой ход: <code>auto(Surname, bugatti, gold, _, _)</code> . Сравниваем сверху вниз - > первые 3 терма не подходят (разные названия функторов - <code>phone_list</code>).
17	Сравнение <code>auto(Surname, bugatti, gold, _, _)</code> с <code>auto(levushkin, shkoda, orange, _, _)</code> . Неудача (<code>bugatti ≠ shkoda</code>).	Откат к <code>auto(Surname, bugatti, gold, _, _)</code> .
18	Сравнение <code>auto(Surname, bugatti, gold, _, _)</code> с <code>auto(levushkin, volvo, grey, _, _)</code> . Неудача (<code>bugatti ≠ volvo</code>).	Откат к <code>auto(Surname, bugatti, gold, _, _)</code> .
19	Сравнение <code>auto(Surname, bugatti, gold, _, _)</code> с <code>auto(samkov, volkswagen, pink, _, _)</code> . Неудача (<code>bugatti ≠ volkswagen</code>).	Откат к <code>auto(Surname, bugatti, gold, _, _)</code> .

20	Сравнение <code>auto(Sername, bugatti, gold, _, _)</code> с <code>auto(samkov, bugatti, gold, _, _)</code> . Успех. Подстановка <code>Sername=samkov</code> .	Прямой ход: <code>bank_list(samkov, Bank, _, _)</code> . Сравнение сверху вниз. Первые 8 термов не подходят (<code>phone_list, auto</code>).
29	Сравнение <code>bank_list(samkov, Bank, _, _)</code> с <code>bank_list(levushkin, sberbank, _, _)</code> . Неудача (<code>samkov ≠ levushkin</code>).	Откат к <code>bank_list(samkov, Bank, _, _)</code> .
30	Сравнение <code>bank_list(samkov, Bank, _, _)</code> с <code>bank_list(samkov, sberbank, _, _)</code> . Успех. Подстановка <code>Bank=sberbank</code> .	Прямой ход: <code>phone_list(samkov, Phone, _)</code> .
31	Сравнение <code>phone_list(samkov, Phone, _)</code> и <code>phone_list(levushkin, «89859771492», _)</code> . Неудача (<code>samkov ≠ levushkin</code>).	Откат к <code>phone_list(samkov, Phone, _)</code> .
32	Сравнение <code>phone_list(samkov, Phone, _)</code> и <code>phone_list(samkov, «89899999», _)</code> . Успех. Подстановка <code>Phone=«89899999»</code> .	Вывод: <code>Sername=samkov</code> , <code>Phone=«89899999»</code> , <code>Bank=sberbank</code> . Откат к <code>phone_list(samkov, Phone, _)</code> .
33	Сравнение <code>phone_list(samkov, Phone, _)</code> и <code>phone_list(ryazanova, «8911911911», _)</code> . Неудача (<code>samkov ≠ ryanova</code>).	Откат к <code>phone_list(samkov, Phone, _)</code> . Следующие 10 термов не подходят (<code>auto, bank_list, get_info_by_model_color</code>). Откат к <code>bank_list(samkov, Bank, _, _)</code> . Следующие 3 термина не подходят (<code>bank_list, get_info_by_model_color</code>). Откат к <code>auto(Sername, bugatti, gold, _, _)</code> .
47	Сравнение <code>auto(Sername, bugatti, gold, _, _)</code> с <code>auto(ryazanova, bugatti, gold, _, _)</code> . Успех. Подстановка <code>Sername=ryazanova</code> .	Прямой ход: <code>bank_list(ryazanova, Bank, _, _)</code> . Следующие 8 термов не подходят (<code>phone_list, auto</code>).
56	Сравнение <code>bank_list(ryazanova, Bank, _, _)</code> с <code>bank_list(levushkin, sberbank, _, _)</code> . Неудача (<code>ryazanova ≠ levushkin</code>).	Откат к <code>bank_list(ryazanova, Bank, _, _)</code> .

57	Сравнение <code>bank_list(ryazanova, Bank, _, _)</code> с <code>bank_list(samkov, sberbank, _, _)</code> . Неудача (<code>ryazanova ≠ samkov</code>)	Откат к <code>bank_list(ryazanova, Bank, _, _)</code> .
58	Сравнение <code>bank_list(ryazanova, Bank, _, _)</code> с <code>bank_list(ryazanova, tinkoff, _, _)</code> . Успех. Подстановка <code>Bank=tinkoff</code> .	Прямой ход: <code>phone_list(ryazanova, Phone, _)</code> .
59	Сравнение <code>phone_list(ryazanova, Phone, _)</code> с <code>phone_list(levushkin, «89859771492», _)</code> . Неудача (<code>ryazanova ≠ levushkin</code>).	Откат к <code>phone_list(ryazanova, Phone, _)</code> .
60	Сравнение <code>phone_list(ryazanova, Phone, _)</code> с <code>phone_list(samkov, «89899999», _)</code> . Неудача (<code>ryazanova ≠ samkov</code>)	Откат к <code>phone_list(ryazanova, Phone, _)</code> .
61	Сравнение <code>phone_list(ryazanova, Phone, _)</code> с <code>phone_list(ryazanova, «8911911911», _)</code> . Успех. Подстановка <code>Phone=«8911911911»</code> .	Вывод: <code>Sername=ryazanova</code> , <code>Phone=89899999</code> , <code>Bank=tinkoff</code> . Откат к <code>phone_list(ryazanova, Phone, _)</code> . Следующие 10 термов не подходят (<code>auto</code> , <code>bank_list</code> , <code>get_info_by_model_color</code>). Откат к <code>bank_list(ryazanova, Bank, _, _)</code> .
72	Сравнение <code>bank_list(ryazanova, Bank, _, _)</code> с <code>bank_list(ryazanova, raiffeisen, _, _)</code> . Успех. Подстановка <code>Bank=raiffeisen</code> .	Прямой ход: <code>phone_list(ryazanova, Phone, _)</code> .
73	Сравнение <code>phone_list(ryazanova, Phone, _)</code> с <code>phone_list(levushkin, «89859771492», _)</code> . Неудача (<code>ryazanova ≠ levushkin</code>).	Откат к <code>phone_list(ryazanova, Phone, _)</code> .
74	Сравнение <code>phone_list(ryazanova, Phone, _)</code> с <code>phone_list(samkov, «89899999», _)</code> . Неудача (<code>ryazanova ≠ samkov</code>)	Откат к <code>phone_list(ryazanova, Phone, _)</code> .

75	<p>Сравнение phone_list(ryazanova, Phone, _) с phone_list(ryazanova, «8911911911», _). Успех. Подстановка Phone=«8911911911».</p>	<p>Вывод: Sername=ryazanova, Phone=89899999, Bank=raiffeisen. Откат к phone_list(ryazanova, Phone, _). Следующие 10 термов не подходят (auto, bank_list, get_info_by_model_color). Откат к bank_list(ryazanova, Bank, _, _). Следующий 1 терм не подходит (get_info_by_model_color). Откат к auto(Sername, bugatti, gold, _, _). Следующие 5 термов не подходят (bank_list, get_info_by_model_color). Откат к get_info_by_model_color. Завершение работы (91 шаг)</p>

2. Один ответ.

goal

```
get_info_by_model_color(shkoda, orange, Surname, Phone, Bank).
```

%Вывод:

```
Surname=levushkin, Phone=89859771492, Bank=sberbank
```

```
1 Solution
```

Порядок поиска ответа на вопрос:

Вопрос будет сопоставляться с каждым предложением сверху вниз, пока не найдется подходящий функтор. В таблице описаны ключевые шаги последующего поиска.

№ шага	Сравниваемые термы; результат; подстановка, если есть	Дальнейшие действия: прямой ход или откат (к чему приводит?)
13	Сравнение <code>get_info_by_model_color(shkoda, orange, Surname, Phone, Bank)</code> с <code>get_info_by_model_color(Model, Color, Surname, Phone, Bank)</code> . Успех. Подстановка <code>Model=shkoda, Color=orange</code> .	Прямой ход: <code>auto(Surname, shkoda, orange, _, _)</code> . Сравниваем сверху вниз -> первые 3 терма не подходят (разные названия функторов - <code>phone_list</code>).
17	Сравнение <code>auto(Surname, shkoda, orange, _, _)</code> с <code>auto(levushkin, shkoda, orange, _, _)</code> . Успех. Подстановка <code>Surname=levushkin</code> .	Прямой ход: <code>bank_list(levushkin, Bank, _, _)</code> . Сравнение сверху вниз. Первые 8 термов не подходят (<code>phone_list, auto</code>).
26	Сравнение <code>bank_list(levushkin, Bank, _, _)</code> с <code>bank_list(levushkin, sberbank, _, _)</code> . Успех. Подстановка <code>Bank=sberbank</code> .	Прямой ход: <code>phone_list(levushkin, Phone, _)</code> .
27	Сравнение <code>phone_list(levushkin, Phone, _)</code> с <code>phone_list(levushkin, «89859771492», _)</code> . Успех. Подстановка (<code>Phone=«89859771492»</code>).	Вывод: <code>Surname=levushkin,</code> <code>Phone=«89859771492»,</code> <code>Bank=sberbank.</code> Откат к <code>phone_list(levushkin, Phone, _)</code> .
28	Сравнение <code>phone_list(levushkin, Phone, _)</code> с <code>phone_list(samkov, «89899999», _)</code> . Неудача (<code>levushkin ≠ samkov</code>).	Откат к <code>phone_list(levushkin, Phone, _)</code> .

29	Сравнение phone_list(levushkin, Phone, _) с phone_list(ryazanova, «89899999», _). Неудача (levushkin \neq ryazanova).	Откат к phone_list(levushkin, Phone, _). Следующие 10 термов не подходят (auto, bank_list, get_info_by_model_color). Откат к bank_list(levushkin, Bank, _, _).
40	Сравнение bank_list(levushkin, Bank, _) с bank_list(samkov, sberbank, _, _). Неудача (levushkin \neq samkov).	Откат к bank_list(levushkin, Bank, _, _).
41	Сравнение bank_list(levushkin, Bank, _) с bank_list(ryazanova, tinkoff, _, _). Неудача (levushkin \neq ryazanova).	Откат к bank_list(levushkin, Bank, _, _).
42	Сравнение bank_list(levushkin, Bank, _) с bank_list(ryazanova, raiffeisen, _, _). Неудача (levushkin \neq ryazanova).	Откат к bank_list(levushkin, Bank, _, _). Следующий 1 терм не подходит (get_info_by_model_color). Откат к auto(Sername, shkode, orange, _, _).
44	Сравнение auto(Sername, shkode, orange, _, _) с auto(levushkin, volvo, grey, _, _). Неудача (shkode \neq volvo).	Откат к auto(Sername, shkode, orange, _, _).
45	Сравнение auto(Sername, shkode, orange, _, _) с auto(samkov, volkswagen, pink, _, _). Неудача (shkode \neq volkswagen).	Откат к auto(Sername, shkode, orange, _, _).
46	Сравнение auto(Sername, shkode, orange, _, _) с auto(samkov, bugatti, gold, _, _). Неудача (shkode \neq bugatti).	Откат к auto(Sername, shkode, orange, _, _).
47	Сравнение auto(Sername, shkode, orange, _, _) с auto(ryazanova, bugatti, gold, _, _). Неудача (shkode \neq bugatti).	Откат к auto(Sername, shkode, orange, _, _). Следующие 5 термов не подходят (bank_list, get_info_by_model_color). Откат к get_info_by_model_color. Завершение работы (52 шага).

3. Ни одного ответа.

goal

```
get_info_by_model_color(volvo, dark, Surname, Phone, Bank).
```

%Вывод:

```
No Solution
```

Порядок поиска ответа на вопрос:

Вопрос будет сопоставляться с каждым предложением сверху вниз, пока не найдется подходящий функтор. В таблице описаны ключевые шаги последующего поиска.

№ шага	Сравниваемые термы; результат; подстановка, если есть	Дальнейшие действия: прямой ход или откат (к чему приводит?)
13	Сравнение <code>get_info_by_model_color(volvo, dark, Surname, Phone, Bank)</code> с <code>get_info_by_model_color(Model, Color, Surname, Phone, Bank)</code> . Успех. Подстановка <code>Model=volvo, Color=dark</code> .	Прямой ход: <code>auto(Surname, volvo, dark, _, _)</code> . Сравниваем сверху вниз - > первые 3 терма не подходят (разные названия функторов - <code>phone_list</code>).
17	Сравнение <code>auto(Surname, volvo, dark, _, _)</code> с <code>auto(levushkin, shkoda, orange, _, _)</code> . Неудача (<code>volvo ≠ shkoda</code>).	Откат к <code>auto(Surname, volvo, dark, _, _)</code> .
18	Сравнение <code>auto(Surname, volvo, dark, _, _)</code> с <code>auto(levushkin, volvo, grey, _, _)</code> . Неудача (<code>dark ≠ grey</code>).	Откат к <code>auto(Surname, volvo, dark, _, _)</code> .
19	Сравнение <code>auto(Surname, volvo, dark, _, _)</code> с <code>auto(samkov, volkswagen, pink, _, _)</code> . Неудача (<code>volvo ≠ volkswagen</code>).	Откат к <code>auto(Surname, volvo, dark, _, _)</code> .
20	Сравнение <code>auto(Surname, volvo, dark, _, _)</code> с <code>auto(samkov, bugatti, gold, _, _)</code> . Неудача (<code>volvo ≠ bugatti</code>).	Откат к <code>auto(Surname, volvo, dark, _, _)</code> .
21	Сравнение <code>auto(Surname, volvo, dark, _, _)</code> с <code>auto(ryazanova, bugatti, gold, _, _)</code> . Неудача (<code>volvo ≠ ryananova</code>).	Откат к <code>auto(Surname, volvo, dark, _, _)</code> . Следующие 5 термов не подходят (<code>bank_list, get_info_by_model_color</code>). Откат к <code>get_info_by_model_color</code> . Завершение работы (26 шагов).

Порядок работы алгоритма унификации при изменении порядка следования в Базе Знаний процедур и знаний в них.

Prolog обрабатывает правило в порядке следования предикатов в его теле, а не в базе знаний, следовательно, их порядок в БЗ не влияет ни на ход работы, ни, тем более, на результат. Таблицы, соответственно, полностью совпадут.

Ниже приведена таблица, демонстрирующая порядок работы алгоритма унификации:

№ шага	Результирующая ячейка	Рабочее поле	П. алг.	Стек
0			1	get_info_by_model_color(bugatti, gold, Surname, Phone, Bank)=get_info_by_model_color(Model, Color, Surname, Phone, Bank):- auto(Surname, Model, Color, _, _), bank_list(Surname, Bank, _, _), phone_list(Surname, Phone, _).
1		get_info_by_model_color(bugatti, gold, Surname, Phone, Bank)=get_info_by_model_color(Model, Color, Surname, Phone, Bank):- auto(Surname, Model, Color, _, _), bank_list(Surname, Bank, _, _), phone_list(Surname, Phone, _).	e	auto(Surname, bugatti, gold, _, _) = auto(samkov, bugatti, gold, _, _), auto(Surname, bugatti, gold, _, _) = auto(ryazanova, bugatti, gold, _, _), bank_list(Surname, Bank, _, _), phone_list(Surname, Phone, _, _).

2	Sername = samkov, Sername = ryazanova	auto(Sername, bugatti, gold, _, _) = auto(samkov, bugatti, gold, _, _), auto(Sername, bugatti, gold, _, _) = auto(ryazanova, bugatti, gold, _, _).	e	bank_list(samkov, Bank, _, _) = bank_list(samkov, sberbank, _, _), bank_list(ryazanova, Bank, _, _) = bank_list(ryazanova, tinkoff, _, _), bank_list(ryazanova, Bank, _, _) = bank_list(ryazanova, raiffeisen, _, _), phone_list(samkov, Phone, _, _) = phone_list(ryazanova, Phone, _, _).
3	Sername = samkov, Bank = sberbank, Sername = ryazanova, Bank = tinkoff, Bank = raiffeisen.	bank_list(samkov, Bank, _, _) = bank_list(samkov, sberbank, _, _), bank_list(ryazanova, Bank, _, _) = bank_list(ryazanova, tinkoff, _, _), bank_list(ryazanova, Bank, _, _) = bank_list(ryazanova, raiffeisen, _, _).	e	phone_list(samkov, Phone, _, _) = phone_list(samkov, «89899999», _, _), phone_list(ryazanova, Phone, _, _) = phone_list(ryazanova, «8911911911», _, _).
4	Sername = samkov, Bank = sberbank, Sername = ryazanova, Bank = tinkoff, Bank = raiffeisen.	phone_list(samkov, Phone, _, _) = phone_list(samkov, «89899999», _, _), phone_list(ryazanova, Phone, _, _) = phone_list(ryazanova, «8911911911», _, _).	e	
Вы-подстановка вод:		Т. к. стек пуст - успех и в рез. ячейнке подстановка		

Выводы

При разном следовании знаний и процедур количество операций не меняется. Поменяется лишь порядок сравнения. Это связано с тем, что (если не считать оптимизацию) алгоритм унификации реализует полный перебор.

Ответы на вопросы

1. В какой части правила сформулировано знание? Это знание о чем, с формальной точки зрения?

Заголовок содержит отдельное знание о предметной области (составной терм), а тело содержит условия истинности этого знания. Заголовок, как составной терм $f(t_1, t_2, \dots, t_m)$, содержит знание о том, что между аргументами: t_1, t_2, \dots, t_m существует отношение (взаимосвязь, взаимозависимость). А имя этого отношения – это f .

2. Что такое процедура?

Процедура – совокупность правил, заголовки которых имеют одно и то же имя и одну и ту же арность, т.е. это совокупность правил, описывающих одно определенное отношение.

3. Сколько в БЗ текущего задания процедур?

В текущей БЗ 4 процедуры. Они описаны в predicates.

4. Что такое пример терма, это частный случай терма, пример? Как строится пример?

Пример терма – результат подстановки конкретных значений в предикат, частный случай предиката. Строится после того, как задан вопрос. Хранится до окончания работы программы.

5. Что такое наиболее общий пример?

Наиболее общий унификатор двух термов – унификатор, соответствующий наиболее общему их примеру.

6. Назначение и результат работы алгоритма унификации. Что значит двунаправленная передача параметров при работе алгоритма унификации, поясните на примере одного из случаев пункта 3.

Работа алгоритма унификации заключается в попарном сопоставлении термов и попытке построить для них общий пример. Алгоритм унификации производит двунаправленную передачу параметров процедурам. Двунаправленная передача параметров при работе алгоритма унификации – передача этих самых

параметров извне в программу для дальнейшего использования или из программы во внешний мир (например, значение параметра, который нас интересует).

7. В каком случае запускается механизм отката?

Механизм отката запускается в случае, когда унификация завершается тупиковой ситуацией или неудачей. При этом происходит откат к предыдущему шагу.

8. Виды и назначение переменных в Prolog. Примеры из задания. Почему использованы те или другие переменные (примеры из задания)?

Назначение переменных

Переменные предназначены для передачи значений в программе. Они являются частью процесса сопоставления и не являются «хранилищем» информации.

Во время вычисления именованные переменные могут конкретизироваться (связываться с различными объектами), причем она может быть переопределена, путем отката вычислительного процесса и отмены ранее проведенной конкретизации для нахождения новых решений.

Анонимные переменные не могут быть связаны со значениями.

Виды переменных

- Именованные – обозначается комбинацией символов латинского алфавита, цифр и символа подчеркивания, начинающейся с прописной буквы или символа подчеркивания. Уникальны в рамках одного предложения.
- Анонимные – обозначаются символом подчеркивания. Любая анонимная переменная уникальна.

Пример из задания

```
get_info_by_model_color(Model, Color, Surname, Phone, Bank) :-  
    auto(Surname, Model, Color, _, _),  
    bank_list(Surname, Bank, _, _),  
    phone_list(Surname, Phone, _).
```

Анонимные переменные используются, чтобы лишней информации не находить и не передавать. Именованные – нужны для достижения цели и конкретизации.