

Основы языка Visual Prolog

Программирование в логике

Логика предикатов — это простейший способ объяснить, как "работает" мышление, и она проще, чем арифметика, которой вы давно пользуетесь.

Пролог включает механизм вывода, который основан на сопоставлении образцов, помощью подбора ответов на запросы он извлекает хранящуюся (известную) информацию.

Синтаксис логики предикатов

Предложения на естественном языке	Синтаксис логики предикатов
Машина красивая	fun (car)
Роза красная	red (rose)
Билл любит машину, если машина красивая	likes (bill, Car) if fun (Car)

Факты и правила

- **объекты** (objects) - Билл, собака
- **отношения** (relations) - любит
- **правила** (rules) - Билл любит собак, если собаки хорошие

- В Прологе отношение или свойства между объектами называется **фактом** (fact).
- Состоит из имени:
 - отношения
 - объекта или объектов, заключенных в круглые скобки
 - завершается точкой (.)

likes (bill, cindy). - Билл любит Синди.

likes (cindy, bill). - Синди любит Билла.

likes (bill, dogs). - Билл любит собак.

- **Правило** — это заключение, для которого известно, что оно истинно, если одно или несколько других найденных заключений или фактов являются истинными.

likes(cindy, Something):- likes(bill, Something). - Сунди любит все, что любит Билл.

likes (caitlin, Something):- green(Something). - Кейтлин любит все зеленое.

- **Символ** : - имеет смысл "если", и служит для разделения двух частей правила: заголовка и тела.

Запросы

- Однократно дав языку Пролог несколько фактов, **можно задавать вопросы**, касающиеся отношений между ними. Это называется **запросом (query) системы языка Пролог**.

likes(bill, cindy).

likes(bill, What).

likes(cindy, What).

Размещение фактов, правил и запросов

Предположим, что есть следующие факты и правила:

Быстрая машина - приятная. (A fast car is fun).

Большая машина — красивая. (A big car is nice).

Маленькая машина — практичная. (A little car is practical).

Биллу нравится машина, если она приятная. (Bill likes a car if the car is fun).

Загрузите программу ch02e01.pro в среду визуальной разработки Visual Prolog запустите ее утилитой Test Goal.

Переменные: общее представление

- В Прологе *переменные* позволяют вам записывать общие факты и правила и задавать общие вопросы.
- Начинаются с заглавной буква (What, Something, X и т.д.)

Предложения

2 типа фраз (*предложения (clause)*):

- **Факт** - представляет либо свойство объекта, либо отношение между объектами. Факт самодостаточен. Прологу не требуется дополнительных сведений для подтверждения факта, и факт может быть использован как основа для логического вывода.
- **Правило** - это конструкция Пролога, которая описывает, что можно логически вывести из других данных. *Правило* — это свойство или отношение, которое достоверно, когда известно, что ряд других отношений достоверен. Синтаксически эти **отношения разделены запятыми.**

Примеры работы с правилами

<p>Диана — вегетарианка и ест только то, что говорит ей ее доктор</p>	<pre>diane_can_eat(Food_on_menu):-v egetable(Food_on_menu), on_doctor_list(Food_on_menu).</pre>
<p>Вам хотелось бы записать факт, который истинен, если <i>Person1</i> является родителем <i>Person2</i>.</p>	<pre>Parent(Person1, Person2):- father(Person1, Person2). Parent(Person1, Person2):- mother(Person1, Person2) .</pre>
<p>Человек может купить машину, если машина ему нравится (<i>likes</i>), и если машина продается (<i>for sale</i>).</p>	<pre>can_buy(Name, Model):- person(Name), car(Model), likes(Name, Model), for_sale(Model).</pre>

Программа ch02e02.pro

- протестируйте ее.

Испытайте следующие цели в разделе

GOAL:

- Что может купить Келли?
- Что может купить Джуди?
- Кто может купить hot-rod?
- Кто может купить машину?

Предикаты

- Отношение в Прологе называется *предикатом*.
- *Аргументы* — это объекты, которые связываются ЭТИМ отношением;
в факте *likes (bill, cindy)* отношение *likes* — это предикат, а объекты *bill* и *cindy* — аргументы.

Переменные

В простом запросе, чтобы найти того, кто любит теннис, можно использовать переменные - *likes(X, tennis)*.

правильные имена переменных:

- *My_first_correct_variable_naitie*
- *Sales_10_11_86*

неправильные:

- *Istattempt*
- *second_attempt*
- *"disaster"*

Испытайте цель:
GOAL likes(Person, tennis).

Инициализация переменных

- Переменные в Прологе инициализируются при сопоставлении с константами в фактах или правилах.
- Нельзя сохранить информацию, присвоив значение переменной. Переменные используются как часть процесса поиска решения, а не как хранилище информации.

Загрузите программу ch02e03.pro

likes(Person, reading), likes(Person, swimming).

likes(Person, reading)

1. *Person* – свободная переменная
2. *reading* – факт известен
3. *Person* - *ellen*
7. *Person* – *john*
8. Совпадений нет
9. *Person* – свободная переменная
10. Продолжается пока не будет найден ответ

likes(Person, swimming)

4. *likes(ellen, swimming)*
5. Совпадений нет
6. *Person* – свободная переменная

Анонимные переменные

- Анонимная переменная может быть использована на месте любой другой переменной и ей никогда не присваивается значение.
- В Прологе анонимные переменные обозначаются символом подчеркивания (`_`).
- Анонимные переменные также можно использовать в фактах (`eats(_)`).
- Анонимные переменные сопоставляются с любыми данными.

Загрузите программу
ch02e04.pro в TestGoal-проект.

Цели (запросы)

- Когда вы даете Прологу запрос, в действительности вы даете ему цель для выполнения.
- **простые:** *likes (ellen, reading)*
- **сложные:** *likes(Person, reading), likes(Person, swimming)*.
- Загрузите программу *ch02e04.pro* и задайте следующую сложную цель:
Goal parent(Person, _), male(Person).
- **Составные цели:** конъюнкция (и - ,) и дизъюнкция (или - ;)

Программы ch02e05.pro

- *Есть ли в списке машина, стоящая меньше, чем \$25000?*
- *Есть ли в списке автомобиль, стоимостью меньше \$25 000, или грузовик стоимостью меньше \$20 000?*

Комментарии

- **Многострочные комментарии** должны начинаться с символов `/*` (косая черта, звездочка) и завершаться символами `*/` (звездочка, косая черта).
- Для установки **однострочных комментариев** можно использовать либо эти же символы, либо начинать комментарий символом процента (`%`).

Пример

- */* Это первый пример комментария */*
- *% Это второй пример комментария*
- */*******

*А эти три строчки — пример
многострочного комментария

*****/*

- */*Вы также можете поместить
комментарий Visual Prolog /*внутри
комментария */ как здесь*/*

Сопоставление

- Идентичные структуры сопоставимы (сравнимы) друг с другом: `parent (joe, tammy)` сопоставимо с `parent (joe, tammy)`.
- *Сопоставление (сравнение)* обычно использует одну или несколько свободных переменных.
- Две свободные переменные могут сопоставляться друг с другом.
- В случае "связывания" между собой нескольких свободных переменных все они называются *совмещенными свободными переменными* .
- В Прологе связывание переменных (со значениями) производится двумя способами: **входе и выходе**.

- Когда переменная передается в предложение, она считается *входным аргументом* и обозначается символом (i).
- Когда же переменная возвращается из предложения, она является *выходным аргументом* и обозначается символом (o).