

Министерство образования и науки Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
Высшего профессионального образования  
Уфимский Государственный Нефтяной Технический Университет  
Кафедра «Вычислительная техника и инженерная кибернетика»

Отчет

По практическому заданию №1  
«Ознакомление с принципами организации и эксплуатации  
базовой Prolog-системы»

По дисциплине «Системы искусственного интеллекта»

Выполнил: студент гр. БГРС-21-03

Минхаеров.Б.Р

Проверил: доцент кафедры информатики

Дмитриев В.Г

**Тема работы:** Ознакомление с принципами организации и эксплуатации базовой Prolog-системы.

**Задание:** ознакомиться с основными возможностями системы SWI-Prolog, правилами заполнения информационной базы, составления элементарных и составных вопросов со стандартными системными предикатами и с составлением простейших правил для их накопления в Prolog-системе.

Выписать ключевые термины и понятия, дать пояснения. Привести свои примеры фрагментов кода. Составить отчет.

## ***Теоретические сведения***

### **а. Общая информация**

Язык Пролог, самый известный из представителей семейства языков логического программирования, вырос из работ Алана Колмерауэра (A. Colmerauer) по обработке естественного языка и независимых работ Роберта Ковальского (R. Kowalski) по приложениям логики к программированию. Дэвиду Уоррену (D. Warren) и его коллегам из Эдинбургского университета удалось осуществить достаточно эффективную реализацию Пролога. Имя Уоррена вошло в историю логического программирования. В его честь названа базовая техника реализации Пролога, получившая название абстрактной машины Уоррена. Программа на языке Пролог представляет собой набор фактов и (возможно) правил. Если программа содержит только факты, то ее называют база данных. Если она содержит еще и правила, то часто используют термин база знаний.

SWI-Prolog распространяется под лицензией GPL, что обеспечивает возможность его использования без нарушений чьих-либо коммерческих интересов. Эта версия языка Пролог доступна как пользователям ОС Linux, так и пользователям Windows.

### **б. Атомы, числа, переменные и составные термы**

Программа на языке Пролог обычно описывает некую действительность. Объекты (элементы) описываемого мира представляются с помощью термов. Терм интуитивно означает объект. Существует 4 вида термов: атомы, числа, переменные и составные термы. Атомы и числа иногда группируют вместе и называют простейшими термами.

**Атом** - это отдельный объект, считающийся элементарным. В Прологе атом представляется последовательностью букв нижнего и верхнего регистра, цифр и символа подчеркивания '\_', начинающейся со строчной буквы. Кроме того, любой набор допустимых символов, заключенный в апострофы, также является атомом. Наконец, комбинации специальных символов + - \* = < > : & также являются атомами (следует отметить, что набор этих символов может отличаться в различных версиях Пролога).

## Пример

Представленные далее последовательности являются корректными атомами:

```
b, abcXYZ, x_123, efg_hij, коля, слесарь,  
'Это также атом Пролога',  
+, ::, <---->, ***
```

**Числа** в Прологе бывают целыми (Integer) и вещественными (Float). Синтаксис целых чисел прост, как это видно из следующих примеров: 1, 1313, 0, -97. Не все целые числа могут быть представлены в машине, их диапазон ограничен интервалом между некоторыми минимальным и максимальным значениями, определенными конкретной реализацией Пролога. SWI-Prolog допускает использование целых чисел в диапазоне от -2147483648 (-231) до 2147483647 (231-1). Синтаксис вещественных чисел также зависит от конкретной реализации. Мы будем придерживаться простых правил, понятных из следующих примеров: 3.14, -0.0035, 100.2. При обычном программировании на Прологе вещественные числа используются редко. Причина этого кроется в том, что Пролог - язык, предназначенный в первую очередь для обработки символьной, а не числовой информации. При символьной обработке часто используются целые числа, нужда же в вещественных числах невелика. Везде, где можно, Пролог старается привести число к целому виду.

**Переменными** в Прологе являются строки символов, цифр и символа подчеркивания, начинающиеся с заглавной буквы или символа подчеркивания:

```
X, _4711, X_1_2, Результат, _x23, Объект2, _
```

Последний пример (единственный символ подчеркивания) является особым случаем - анонимной переменной (переменной без имени). Анонимная переменная применяется, когда ее значение не используется в программе. Возможное неоднократное употребление безымянной переменной в одном выражении применяется для того, чтобы подчеркнуть наличие переменных при отсутствии их специфической значимости.

**Составные термы** (функции) состоят из имени функции (нечислового атома) и списка аргументов (термов Пролога, то есть атомов, чисел, переменных или других составных термов), заключенных в круглые скобки и разделенных запятыми. Группы составных термов используют для составления фраз Пролога. Нельзя помещать символ пробела между функтором (именем функции) и открывающей круглой скобкой. В других позициях, однако, пробелы могут быть полезны для создания более читаемых программ. Ниже приведены два составных терма:

```
итого(клиент(X,23,_), 71)  
'Что случилось?'(ничего)
```

При задании имен термов предпочтительнее использовать мнемонические ("говорящие") имена, так как терм а(ж), например, гораздо менее информативен, чем терм автор(жюль\_верн).

Еще одной важной структурой данных в Прологе является список. Мы познакомимся с ним позднее. Сейчас отметим только один из видов списков -

список символов. Такие списки могут быть представлены в виде строк, например, первый аргумент составного термина возраст("Борис",10) - строка. При записи строки заключаются в кавычки.

### с. Запросы, факты и правила

**Запрос** - это последовательность предикатов, разделенных запятыми и завершающаяся точкой. На естественном языке запятая соответствует союзу "и", а на языке математической логики обозначает конъюнкцию. С помощью запросов можно "спрашивать" базу данных о том, какие утверждения являются истинными. Предикат запроса называется целью. Простые вопросы, не содержащие никаких переменных, называют да-нет-вопросами. Они допускают лишь два возможных ответа: "Yes" означает наличие соответствующего факта в базе данных (первый запрос примера, приведенного ниже), "No" - его отсутствие (второй запрос). В случае ответа "Yes" говорят, что запрос завершился успехом, цель достигнута.

Пример

?- больше(слон, лошадь), больше(лошадь, осел).

Yes

?- больше(слон, собака).

No

Использование переменных в запросах позволяет задавать более сложные вопросы. Предположим, например, что мы хотим определить, какие животные больше осла? В следующем запросе переменная X обозначает искомый ответ:

?- больше(X, осел).

X = лошадь

Yes

При обработке запроса переменная X приняла значение "лошадь". Просматривая базу данных, интерпретатор обнаружил факт, утверждающий, что лошадь больше осла, и запрос был успешно выполнен.

Запросы с переменными могут иметь более одного решения. Первым всегда выводится то из решений, которое находится ближе к началу базы данных. Если нам достаточно только одного ответа, то можно нажать Enter и закончить поиск. В случае, если мы захотим получить очередной ответ, нужно нажать клавишу ; (точка с запятой), и Пролог начнет поиск других вариантов ответа на запрос. Сообщение "No" говорит об отсутствии очередного решения.

**Факт** - это утверждение о том, что соблюдается некоторое конкретное отношение. Он является безусловно верным. В разговорной речи под фактом понимается нечто вроде "Сегодня солнечно" или "Васе 10 лет". На Прологе это запишется в виде

'Сегодня солнечно'.

'Васе 10 лет'.

Если вы сохраните эти факты в файле и затем загрузите его, то можно задавать вопросы интерпретатору Пролога (напомним, что запрос вводится после приглашения Пролога, которое в большинстве версий имеет вид ?-), например,

?- 'Сегодня солнечно'.

Yes

?- 'Васе 10 лет'.

Yes

?- 'Сегодня солнечно', 'Васе 10 лет'.

Yes

Запятая между фактами в последнем запросе означает операцию логического и (конъюнкцию).

Такая форма записи соответствует логике высказываний, возможности которой, как уже говорилось, достаточно ограничены. Мы не можем задать, например, вопрос о том, сколько лет Васе. Гораздо удобнее использовать параметризованные факты, работу с которыми поддерживает логика предикатов. На Прологе факт может быть записан в виде предиката, аргументы которого являются символьными или числовыми константами.

**Правила.** Пролог могут содержать правила, позволяющие получать дополнительные знания о том мире, который описывает программа. Правило задает новый предикат через определенные ранее. Правило состоит из головы (предиката) и тела (последовательности предикатов, разделенных запятыми). Голова и тело разделены знаком :- и, подобно каждой фразе Пролога, правило должно заканчиваться точкой. Запятая в теле правила означает конъюнкцию (&&, логическое и).

Знак :- есть схематическая запись стрелки (<-) и показывает, что из правой части следует левая. Этот знак читается как "если". Интуитивный смысл правила состоит в том, что цель, являющаяся головой, будет истинной, если Пролог сможет показать, что все выражения (подцели) в теле правила являются истинными.

Пример

Правило, определяющее отношение ребенок/2 через отношение отец/2, запишется следующим образом:

ребенок(X, Y) :- отец(Y, X).

Это означает, что если человек Y является для человека X отцом, то X является ребенком Y. Здесь X и Y - переменные. Напомним, что запись ребенок/2 показывает, что предикат ребенок является функцией от двух аргументов.

#### d. Предикаты

**Предикат** - это логическая функция от одного или нескольких аргументов, то есть функция, действующая в множество из двух значений: истина и ложь. Предикат Пролога записывается в виде составного терма:

имя\_предиката(аргументы).

Аргументы перечисляются через запятую и представляют собой какие-то объекты или свойства объектов, а имя предиката обозначает связь или отношение между аргументами. Предикат однозначно определяется парой: имя и количество аргументов. Два предиката с одинаковым именем, но различным количеством аргументов, считаются различными. Количество параметров предиката называется его арностью (arity). При описании предиката арность указывают после его имени, разделяя их символом '/' (слэш).

Как правило, имена предикатов и аргументов записываются в именительном падеже. Пробелы в них не допускаются, поэтому в качестве разделителей в символьных константах используется символ подчеркивания.

Пример

Факт "Коля работает слесарем" на Прологе запишется следующим образом:  
профессия(коля, слесарь).

Здесь предикат профессия/2 имеет два аргумента: первый означает имя человека, а второй - профессию. Факт "Борису 10 лет" можно представить в виде:

возраст("Борис", 10).

Порядок аргументов предиката связан со смыслом факта и поэтому не изменяем. При записи фактов надо помнить, что:

имя факта начинается со строчной буквы;

запись каждого факта заканчивается точкой.

В приведенных выше примерах профессия/2 и возраст/2 - предикаты (составные термы), коля и слесарь - атомы, 10 - число, "Борис" - строка. Подробнее о видах термов Пролога рассказывается в следующем разделе.

**База данных** на Прологе - это совокупность фактов. В процессе работы в базу данных можно добавлять новые факты, удалять или изменять старые.

Пример

Составим базу данных из следующих фактов: "слон больше, чем лошадь", "лошадь больше, чем осел", "осел больше, чем собака" и "осел больше, чем обезьяна":

больше(слон, лошадь).

больше(лошадь, осел).

больше(осел, собака).

больше(осел, обезьяна).

Мы использовали предикат больше/2, имеющий два параметра.

После сохранения этой базы данных в текстовом файле можно формулировать запросы к интерпретатору Пролога:

?- больше(слон, лошадь).

Yes

?- больше(лошадь, слон).

No