Практическое занятие № 5

Цель: построение программы с использованием стандартных доменов. Получение навыков формирования правил для программ на языке Пролог и составления запросов для выявления конечной цели.

Ход работы:

Проект находится в том же репозитории что и проект

Контрольные вопросы:

1. Факты и правила в Прологе.

**Факт** записывается как предикат с одним или более объектами. Но факт может быть предикатом, не содержащим объектов, то есть *предикатом с нулевой арностью*.

**Правило** используется Прологом (Prolog) для проверки условия вывода нового факта. **Правила** в Прологе (Prolog) записываются в разделе утверждений (clauses). Программа может содержать достаточно много правил, вырабатывающих различные заключения. Эффект введения правил точно такой же, как если бы программа содержала большое число утверждений-фактов.

1. Типы данных в Прологе

Термины подразделяются на

Переменные. Заполнитель, не привязанный к какому-либо конкретному термину. Переменные идентифицируются символами, соответствующими регулярному выражению [A-Z\_][A-Za-z\_0-9]\*. Переменная \_ является специальной: это анонимная переменная. Каждое вхождение \_ обозначает отдельную переменную. Например, учитывая факт,foo(1,2,3). Тест, подобный foo(\_,\_,\_). будет успешным, тогда как тест, подобный foo(A,A,A). завершится неудачей.Однако, как только переменная объединена со значением (привязана к нему), она перестает быть переменной: если не отменена путем обратного отслеживания, она навсегда остается тем, с чем она объединена.

Числа могут быть либо плавающими, либо целыми. Применяются обычные правила (например, -321 - это целое число, -321.0 или что-то вроде -3.21e+02 - это число с плавающей точкой.

Атомы - это имена, обычно обозначаемые словом, начинающимся со строчной буквы (например, atom), соответствующим обычному повторному выражению [a-z][A-Za-z0-9\_]\*. В качестве альтернативы атомы могут быть разделены апострофами (например, 'atom'), которые удобно позволяют использовать символы, недопустимые в противном случае. Синтаксис для atom несколько сложнее этого: по сути, все, что не попадает в другую категорию, образует атом (например, специальный атом [], обозначающий пустой список, и запятая (,), обозначающая соединение, - все это атомы.

Все остальное, по сути, является структурой, т.е. кортежами терминов, помеченных функтором (именем, соответствующим правилам для атома) с арностью (количеством аргументов). Можно даже рассматривать атомы как структуру с арностью 0.

Поверх других "типов данных" prolog насыпан синтаксический сахар:

Списки обозначаются structure ./2, причем левый аргумент является началом списка, а правый - его концом. Пустой список обозначается атомом []. Например,

список [a] внутренне представлен как .(a,[]),

список [a,b] как .(a,.(b,[])) и

список [a,b|[c]] как .(a,.(b,.(c,[]))).

Следует отметить, что можно написать список, используя любую нотацию: они будут соответствующим образом унифицированы. Однако вы можете увидеть привлекательность использования записи списка в квадратных скобках.

Аналогичный синтаксический сахар применяется к строкам. Строка может быть записана как текстовая строка, разделенная двойными кавычками: "Кот и шляпа". Однако внутренне строки представлены в виде списков целых чисел, представляющих кодовые точки для каждого из символов во внутренней кодировке реализаций.