

## Лабораторная работа № 11

### Методические указания

#### Рекурсия на Prolog

**Цель работы** – изучить рекурсивные способы организации программ на Prolog, методы формирования эффективных рекурсивных программ обработки списков и порядок их реализации.

**Задачи работы:** приобрести навыки использования списков на Prolog, эффективного способа их обработки, организации и порядка работы соответствующих программ.

Изучить особенность использования переменных при обработке списков. Способ формирования и изменения резольвенты в этом случае и порядок формирования ответа.

#### Краткие теоретические сведения

Язык Prolog, позволяющий решать задачи искусственного интеллекта, часто использует списки – рекурсивно описанные структуры данных. Рекурсивно описанные структуры удобно обрабатывать рекурсивным способом.

**Рекурсия** – это один из способов организации повторных вычислений. Т.к. логическое программирование – не операторное, то рекурсия – это способ заставить систему использовать многократно одну и ту же процедуру (знание). Но этот процесс рано или поздно надо остановить. Поэтому в рекурсивных процедурах должна быть предусмотрена возможность выхода из рекурсии – специальное предложение процедуры. Напомним, что эффективный способ организации рекурсии – это хвостовая рекурсия. В логическом программировании это особенно важно в силу не детерминированного поиска ответа на вопрос! Кроме этого, повысить эффективность рекурсивной процедуры можно отсекая неперспективные пути поиска решения. В этих целях используется предикат отсечения, который, при необходимости, включают в тело некоторых правил.

Система работает формально, используя резольвенту (см. лекцию), алгоритм унификации и механизм отката. При работе со списками, в алгоритме унификации участвуют списки и переменные, обозначающие списки целиком или их части. Синтаксическая форма обозначения списков в тексте программы упрощена, для облегчения читабельности текста. Однако это ни как не меняет логику и порядок работы основных механизмов, встроенных в систему! (см лекцию).

#### Задание

Используя хвостовую рекурсию, разработать (комментируя назначение

аргументов) эффективную программу, позволяющую:

1. Найти длину списка (по верхнему уровню);
2. Найти сумму элементов числового списка;
3. Найти сумму элементов числового списка, стоящих на нечетных позициях исходного списка (нумерация от 0);
4. Сформировать список из элементов числового списка, больших заданного значения;
5. Удалить заданный элемент из списка (один или все вхождения).
6. Объединить два списка.

Убедиться в правильности результатов.

Для одного из вариантов ВОПРОСА уметь составить таблицу, отражающую конкретный порядок работы системы:

(Т.к. резольвента хранится в виде стека, то состояние резольвенты требуется отображать в столбик: вершина – сверху! Новый шаг надо начинать с нового состояния резольвенты!)

Форма таблицы:

Вопрос:.....

№ шага	Состояние резольвенты, и вывод: дальнейшие действия (почему?)	Для каких термов запускается алгоритм унификации: $T1=T2$ и каков результат (и подстановка)	Дальнейшие действия: прямой ход или откат (почему и к чему приводит?)
1....	...		Комментарий, вывод...
2	...		...

Содержание отчета

В отчете по лабораторной работе должны быть приведены:

- Полный текст задания!!!,
- Текст программы, Варианты вопросов,
- Таблица, демонстрирующая работу системы при одном из успешных вариантов вопроса.

#### Список рекомендуемой литературы

1. Шрайнер П.А. Основы программирования на языке Пролог. Курс лекций. Учебное пособие — М.: Интернет-Ун-т Информ. Технологий, 2005. Вирт Н. Алгоритмы и структуры данных: Пер. с англ. СПб.: Невский диалект, 2001. С.261 – 274, 324–336.
2. Ездаков А.Л. Функциональное и логическое программирование: учебное пособие — М.: БИНОМ. Лаборатория знаний, 2009.
3. А.Н. Адаменко, А.М. Кучуков. Логическое программирование и Visual Prolog — СПб.: БХВ-Петербург, 2003.
4. Братко И. Программирование на языке Пролог для искусственного интеллекта. - М.: Мир, 1990.