Лабораторная работа № 10

Методические указания

Рекурсия на Prolog

Цель работы – изучить рекурсивные способы организации программ на Prolog, методы формирования эффективных рекурсивных программ и порядок реализации таких программ.

Задачи работы: приобрести навыки использования рекурсии на Prolog, эффективного способа ее организации и прядка работы соответствующей программы.

Изучить возможность и необходимость использования системных предикатов в рекурсивной программе на Prolog, принципы и особенности порядка работы такой программы. Способ формирования и изменения резольвенты в этом случае и порядок формирования ответа.

Краткие теоретические сведения

Prolog — это декларативный язык программирования, при использовании которого решение задачи получается путем логического вывода из ранее известных положений.

Одним из основных принципов логического программирования является недетерминизм порядка поиска ответа на вопрос, что приводит к возможности ошибок и требует использования механизма backtracking. При этом, система должна знать куда откатиться, для чего она использует специальный стек точек возврата. А чтобы восстановить предыдущее состояние резольвенты и выполнить ре- конкретизацию она еще должна хранить дополнительную информацию. Поэтому большое количество возможных переборов может сильно снизить эффективность работы программы. Значит, необходимо уметь эффективно описывать предметную область и отсекать бесперспективные пути поиска решения.

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 привил.

Задание

Используя хвостовую рекурсию, разработать программу, позволяющую найти

1. n!,
2. n-е число Фибоначчи.

Убедиться в правильности результатов.

Для одного из вариантов ВОПРОСА и каждого задания 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!

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

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	...		…

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

Для одного из вариантов ВОПРОСА и конкретной БЗ составить таблицу, отражающую конкретный порядок работы системы, с объяснениями: очередная проблема на каждом шаге и метод ее решения; каково новое текущее состояние резольвенты, как получено; какие дальнейшие действия? (Запускается ли алгоритм унификации? Каких термов? Почему этих?) ; вывод по результатам очередного шага и дальнейшие действия.

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.: нтернет-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.