

Лабораторная работа № 18

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 порядка 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

Язык Prolog – позволяющий решать задачи искусственного интеллекта, является без-типовым языком программирования. 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 отсекаать бесперспективные пути поиска решения

Система работает формально, используя некоторые области памяти и информацию, хранящуюся в них. Система использует резольвенту (см. лекцию), алгоритм унификации и механизм отката. Резольвента хранит совокупность термов, истинность которых надо доказать – цели. Для доказательства очередной цели используется алгоритм унификации. Порядок просмотра всех утверждений программы, при этом, – сверху вниз. Если система выбрала некоторое знание для использования, она помечает это правило (возможная точка возврата).

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. Но этот процесс рано или поздно надо остановить. Поэтому в рекурсивных процедурах должна быть предусмотрена возможность выхода из рекурсии – специальное предложение процедуры. Напоминаю, что эффективный

способ организации рекурсии – это **хвостовая рекурсия**. В логическом программировании это особенно важно в силу недетерминизма 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.

Указания к выполнению работы

При разработке программы следует обратить внимание на эффективность программы.

Необходимо объяснить порядок и особенности выполнения программы и формирования результата.

Задание

Ответить на вопросы:

- Что такое рекурсия? Как организуется хвостовая рекурсия в Prolog? Как организовать выход из рекурсии в Prolog?
- Какое первое состояние резольвенты?
- В каком случае система запускает алгоритм унификации? Каково назначение использования алгоритма унификации? Каков результат работы алгоритма унификации?
- В каких пределах программы переменные уникальны?
- Как применяется подстановка, полученная с помощью алгоритма унификации?
- Как изменяется резольвента?
- В каких случаях запускается механизм отката?

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!

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...

...
-----	-----	-----	-----

Содержание отчета

В отчете по лабораторной работе должны быть приведены:

- **Полный текст задания!!!,**
- Ответы на вопросы,
- **Текст программы с комментариями** обозначений и идентификаторов аргументов,
- **Таблицы для каждого задания,** демонстрирующие работу системы при одном из успешных вариантов вопроса.
- **Выводы:** за счет чего достигнута **эффективность** работы системы.

В отчет требуется обязательно включить все разделы, указанные выше!!! Отчет представляется в электронном или печатном виде. **Учитываются качество и сроки** поступления отчета.

По результатам лабораторной работы проводится собеседование.

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.