|  |  |
| --- | --- |
|  | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ «Информатика и системы управления»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ДИСЦИПЛИНА «Функциональное и логическое программирование»

**Лабораторная работа № 18**

**“Рекурсия на Prolog”**

|  |  |
| --- | --- |
| **Студент**  Чалый А. А.  **Группа** ИУ7 – 62 Б  **Оценка (баллы) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **Преподаватель** Толпинская Н.Б. |  |

Москва.

2021 г.

**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.

**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!

Листинг 1. recs.pro

predicates

factorial(integer, integer)

f(integer, integer, integer, integer)

fib(integer, integer)

fibin(integer, integer, integer, integer)

clauses

factorial(X, Res):- f(X, Res, 1, 1).

f(X, Res, X, Res):- !.

f(X, Res, X1, Res1):- Tmp = X1+1, Res\_tmp = Res1 \* Tmp, f(X, Res, Tmp, Res\_tmp).

fib(X, R):- fibin(X, R, 1, 0).

fibin(1, R, R, \_):- !.

fibin(X, R, X1, X2):- R1 = X1 + X2, P = X - 1, fibin(P, R, R1, X1).

Результат работы программы:

Факториал:

factorial(4, Res) → 24

factorial(1, Res) → 1

Фибоначчи:

fib(4, Res) → 3

fib(1, Res) → 1

fib(2, Res) → 1

Таблица 1. factorial(3, Res)

|  |  |  |  |
| --- | --- | --- | --- |
| № шага | Состояние резольвенты, и вывод: дальнейшие действия (почему?) | Для каких термов запускается алгоритм унификации: Т1=Т2 и каков результат (и подстановка) | Дальнейшие действия: прямой ход или откат (почему и к чему приводит?) |
| 1 | factorial(3, Res)  В качестве первого состояния в резольвенту помещается вопрос | T1 = factorial(3, Res)  T2 = factorial(X, Res)  Попытка унификации. Унификация успешна.  Подстановка:  {Х=3, Res=Res} | Прямой ход |
| 2 | f(3, Res, 1, 1) | T1 = f(3, Res, 1, 1)  T2 = factorial(X, Res)  Попытка унификации. Унификация не успешна. Разные функторы | Переход к следующему предложению |
|  | f(3, Res, 1, 1) | T1 = f(3, Res, 1, 1)  T2 = f(X, Res, X, Res)  Попытка унификации. Унификация не успешна. 3!= 1 | Переход к следующему предложению |
|  | f(3, Res, 1, 1) | T1 = f(3, Res, 1, 1)  T2 =f(X, Res, X1, Res1)  Попытка унификации. Унификация успешна.  Подстановка:  {Х=3, Res=Res, X1 = 1, Res1 = 1} | Прямой ход |
| 3 | Tmp = 1+1  Res\_tmp = 1 \* Tmp  f(3, Res, Tmp, Res\_tmp) | Tmp = 2 | Прямой ход |
| 4 | Res\_tmp = 1 \* 2  f(3, Res, 2, Res\_tmp) | Res\_tmp = 2 | Прямой ход |
| 5 | f(3, Res,2, 2) | T1 = f(3, Res,2, 2)  T2 = factorial(X, Res)  Попытка унификации. Унификация не успешна. Разные функторы | Переход к следующему предложению |
|  | f(3, Res,2, 2) | T1 = f(3, Res,2, 2)  T2 = f(X, Res, X, Res)  Попытка унификации. Унификация не успешна. 3!= 2 | Переход к следующему предложению |
|  | f(3, Res, 2, 2) | T1 = f(3, Res, 2, 2)  T2 =f(X, Res, X1, Res1)  Попытка унификации. Унификация успешна.  Подстановка:  {Х=3, Res=Res, X1 = 2, Res1 = 2} | Прямой ход |
| 6 | Tmp = 2+1  Res\_tmp = 2 \* Tmp  f(3, Res, Tmp, Res\_tmp) | Tmp = 3 | Прямой ход |
| 7 | Res\_tmp = 2 \* 3  f(3, Res, 3, 6) | Res\_tmp = 6 | Прямой ход |
| 8 | f(3, Res, 3, 6) | T1 = f(3, Res,3, 6)  T2 = factorial(X, Res)  Попытка унификации. Унификация не успешна. Разные функторы | Переход к следующему предложению |
| 9 | f(3, Res,3, 6) | T1 = f(3, Res,3, 6)  T2 = f(X, Res, X, Res)  Попытка унификации. Унификация успешна. Подстановка:  {X=3, Res=Res, X=3, Res=6} | Прямой ход |
| 10 | ! | ! - указывает прологу отменить поиск альтернатив для целей до него | Альтернатив не искать. Завершение работы. Вывод результата. |

Таблица 2. fib(3, R)

|  |  |  |  |
| --- | --- | --- | --- |
| № шага | Состояние резольвенты, и вывод: дальнейшие действия (почему?) | Для каких термов запускается алгоритм унификации: Т1=Т2 и каков результат (и подстановка) | Дальнейшие действия: прямой ход или откат (почему и к чему приводит?) |
| 1 | fib(3, R)  В качестве первого состояния в резольвенту помещается вопрос | T1 = fib(3, R)  T2 = fib(X, R)  Попытка унификации. Унификация успешна.  Подстановка:  {Х=3, R=R} | Прямой ход |
| 2 | fib1(3, R, 1, 0). | T1 = fib1(3, R, 1, 0).  T2 = fib(X, R)  Попытка унификации. Унификация не успешна. Разные функторы | Переход к следующему предложению |
|  | fib1(3, R, 1, 0). | T1 = fib1(3, R, 1, 0).  T2 =fib1(1, R, R, \_)  Попытка унификации. Унификация не успешна. 3!=1 | Переход к следующему предложению |
|  | fib1(3, R, 1, 0). | T1 = fib1(3, R, 1, 0).  T2 =fib1(X, R, X1, X2)  Попытка унификации. Унификация успешна.  Подставновка  {X=3, R=R, X1=1, X2=0} | Прямой ход |
| 3 | R1 = 1 + 0  P = 3 — 1  fib1(P, R, R1, 1) | R1=1 | Прямой ход |
| 4 | P = 3 — 1  fib1(P, R, 1, 1) | P=2 | Прямой ход |
| 5 | fib1(2, R, 1, 1) | T1 = fib1(2, R, 1, 1).  T2 = fib(X, R)  Попытка унификации. Унификация не успешна. Разные функторы | Переход к следующему предложению |
|  |  | T1 = fib1(2, R, 1, 1).  T2 =fib1(1, R, R, \_)  Попытка унификации. Унификация не успешна. 2!=1 | Переход к следующему предложению |
|  |  | T1 = fib1(2, R, 1, 1).  T2 =fib1(X, R, X1, X2)  Попытка унификации. Унификация успешна.  Подставновка  {X=2, R=R, X1=1, X2=1} | Прямой ход |
| 6 | R1 = 1 + 1  P = 2 — 1  fib1(P, R, R1, 1) | R1= 2 | Прямой ход |
| 7 | P = 2 — 1  fib1(P, R, 2, 1) | P= 1 | Прямой ход |
| 8 | fib1(1, R, 2, 1) | T1 = fib1(1, R, 2, 1).  T2 = fib(X, R)  Попытка унификации. Унификация не успешна. Разные функторы | Переход к следующему предложению |
|  | fib1(1, R, 2, 1) | T1 = fib1(1, R, 2, 1).  T2 =fib1(1, R, R, \_)  Попытка унификации. Унификация успешна.  Подстановка: {  1=1, R=R, R=2, \_=1} | Прямой ход |
| 9 | ! | ! - указывает прологу отменить поиск альтернатив для целей до него | Альтернатив не искать. Завершение работы. Вывод результата. |

**Выводы**

Эффективность системы достигнута за счет использования хвостовой рекурсии и отсечений. Необходимо обеспечить, чтобы после выхода из рекурсии программа не пыталась искать альтернативные ответы, для этого используется отсечение. Это позволяет сократить количество шагов, необходимых для получения результата.

**Контрольные вопросы:**

**Что такое рекурсия?**

Рекурсия – это ссылка на описываемый объект при описании объекта.

**Как организуется хвостовая рекурсия в Prolog?**

* Рекурсивный вызов единственен и расположен в конце тела правила
* Не должно быть возможности сделать откат до вычисления рекурсивного вызова

**Как организовать выход из рекурсии в Prolog?**

С помощью отсечения.

**Какое первое состояние резольвенты?**

Заданный вопрос (goal).

**В каком случае система запускает алгоритм унификации?**

Система запускает алгоритм унификации автоматически при необходимости что-то доказать

**Каково назначение и результат использования алгоритма унификации?**

Унификация – механизм логического вывода. Результат – подстановка.

**В каких пределах программы переменные уникальны?**

Именованная переменная уникальна в рамках предложения, в котором она используется. Анонимные переменные всегда уникальны.

**Как применяется подстановка, полученная с помощью алгоритма унификации?**

Подстановка применяется к целям в резольвенте путем замены текущей переменной на соответствующий терм.

**Как изменяется резольвента?**

Преобразования резольвенты выполняются с помощью редукции. Редукцией цели G с помощью программы P называется замена цели G телом того правила из P, заголовок которого унифицируется с целью. Новая резольвента образуется в два этапа:

1. в текущей резольвенте выбирается одна из подцелей и для неё выполняется редукция;
2. к полученной конъюнкции целей применяется подстановка, полученная как наибольший общий унификатор цели и заголовка сопоставленного с ней правила.

**В каких случаях запускается механизм отката?**

Механизм отката запускается в случае неудачи алгоритма унификации.