|  |  |
| --- | --- |
|  | **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»

**Лабораторная работа № 18**

|  |  |
| --- | --- |
| **Студент:** Керимов А. Ш.  **Группа:** ИУ7-64Б  **Преподаватель:** Толпинская Н. Б. |  |

Москва.

2020 г.

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

**Задание.**

**Ответить на вопросы:**

* **Что такое рекурсия? Как организуется хвостовая рекурсия в Prolog? Как организовать выход из рекурсии в Prolog?**

Рекурсия – один из способов организации повторных вычислений. В логическом программировании – способ заставить систему многократно использовать одну и ту же процедуру. При этом из нее должен быть выход.

Организация хвостовой рекурсии:

* + Рекурсивный вызов единственен и расположен в конце тела правила.
  + До вычисления рекурсивного вызова не должно быть возможности сделать откат (т. е. точки отката отсутствуют). Этого можно добиться, например, с помощью предиката отсечения.

Использовать отдельное правило, в конце которого будет находиться предикат отсечения.

* **Какое первое состояние резольвенты?**

Вопрос

* **В каком случае система запускает алгоритм унификации? Каково назначение использования алгоритма унификации? Каков результат работы алгоритма унификации?**

Пролог выполняет унификацию в двух случаях: когда цель сопоставляется с заголовком предложения или когда используется знак равенства, который является инфиксным предикатом (предикатом, который расположен между своими аргументами, а не перед ними).

Алгоритм унификации необходим для попытки "увидеть одинаковость" – сопоставимость двух термов, может завершаться успехом или тупиковой ситуацией. Результат унификации – ответ «да» или «нет».

* **В каких пределах программы переменные уникальны?**

Именованные переменные уникальны в пределах одного предложения, анонимные уникальны все.

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

Подстановка применяется к подцелям резольвенты, путем конкретизации переменных.

* **Как изменяется резольвента?**

Резольвента - текущая цель, существующая на любой стадии вычислений. Резольвенты порождаются целью и каким-либо правилом или фактом, которые просматриваются последовательно сверху вниз. Если резольвента существует при наиболее общей унификации, она вычисляется. Если пустая резольвента с помощью такой стратегии не найдена, то ответ на вопрос отрицателен.

* **В каких случаях запускается механизм отката?**

Механизм отката запускается в 2 случаях:

1. Если алгоритм попал в тупиковую ситуацию.

2. Если резольвента не пуста и решение найдено, но в базе знание остались не отмеченные предложения.

**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: Т1=Т2 и каков **результат** (и подстановка) | Дальнейшие действия: прямой ход или откат (почему и к чему приводит?) |
| 1… | … | … | Комментарий, вывод… |
| … | … | … | … |

# Практическая часть

*Листинг 1. Факториал и Фибоначчи*

**predicates**

factorial(**integer**, **integer**)

factorial\_rec(**integer**, **integer**, **integer**, **integer**)

fib(**integer**, **integer**)

fib\_rec(**integer**, **integer**, **integer**, **integer**)

**clauses**

factorial(N, F) :- factorial\_rec(N, F, 1, 1).

factorial\_rec(N, F, N, F) :- !.

factorial\_rec(N, F, N2, F2) :-

Tmp\_N = N2 + 1,

Tmp\_F = F2 \* Tmp\_N,

factorial\_rec(N, F, Tmp\_N, Tmp\_F).

fib(N, Res) :- fib\_rec(N, Res, 1, 1).

fib\_rec(1, Res, \_, Res) :- !.

fib\_rec(N, Res, N1, N2) :-

Tmp\_N = N - 1,

Tmp\_N2 = N1 + N2,

fib\_rec(Tmp\_N, Res, N2, Tmp\_N2).

**goal**

fib(3, Result).

**Примеры целей**

1. factorial(3, Result).

Result=6

1. fib(3, Result).

Result=5

**Порядок работы системы** для цели factorial(2, Result).

|  |  |  |  |
| --- | --- | --- | --- |
| № 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 | Цель заносится в резольвенту.  factorial(2, Result).  Запуск процесса редукции. | Попытка унификации:  factorial(2, Result) = factorial(N, F)  Результат: успех, подстановка: {N=2. Result=F} | Прямой ход  Преобразование резольвенты: замена текущей цели на тело найденного с помощью унификации правила, применение полученной подстановки |
| 2 | factorial\_rec(2, F, 1, 1) Выполнение процесса редукции. | Попытка унификации:  factorial\_rec(2, F, 1, 1) = factorial(N, F)  Результат: неудача, различные имена главных функторов | Прямой ход, переход к следующему предложению |
| 3 | factorial\_rec(2, F, 1, 1) Выполнение процесса редукции. | Попытка унификации:  factorial\_rec(2, F, 1, 1) = factorial\_rec(N, F, N, F)  Результат: неудача, различные константы | Прямой ход, переход к следующему предложению |
| 4 | factorial\_rec(2, F, 1, 1) Выполнение процесса редукции. | Попытка унификации:  factorial\_rec(2, F, 1, 1) = factorial\_rec(N, F, N2, F2)  Результат: успех, подстановка: {N=2, N2=1, F2=1} | Прямой ход  Преобразование резольвенты: замена текущей цели на тело найденного с помощью унификации правила, применение полученной подстановки |
| 5 | Tmp\_N = 1 + 1,  Tmp\_F = 1 \* Tmp\_N,  factorial\_rec(2, F, Tmp\_N, Tmp\_F)  Запуск алгоритма редукции для верхней подцели. | Попытка унификации:  Tmp\_N = 1 + 1  Результат: успех, подстановка: {Tmp\_N=2} | Прямой ход |
| 6 | Tmp\_F = 1 \* 2,  factorial\_rec(2, F, 2, Tmp\_F)  Запуск алгоритма редукции для верхней подцели. | Попытка унификации:  Tmp\_F = 1 \* 2  Результат: успех, подстановка: {Tmp\_F=2} | Прямой ход |
| 7 | factorial\_rec(2, F, 2, 2)  Запуск алгоритма редукции для верхней подцели. | Попытка унификации:  factorial\_rec(2, F, 2, 2) = factorial(N, F)  Результат: неудача, различные имена главных функторов | Прямой ход, переход к следующему предложению |
| 8 | factorial\_rec(2, F, 2, 2)  Выполнение процесса редукции. | Попытка унификации:  factorial\_rec(2, F, 2, 2) = factorial\_rec(N, F, N, F)  Результат: успех, подстановка: {N=2, F=2} | Прямой ход  Преобразование резольвенты: замена текущей цели на тело найденного с помощью унификации правила, применение полученной подстановки |
| 9 | !.  Выбор верхней подцели резольвенты, запуск редукции. | Выполнение отсечение | Завершение работы программы. Резольвента пуста. Result=2 |

**Порядок работы системы** для цели fib(2, Result).

|  |  |  |  |
| --- | --- | --- | --- |
| № 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 | Цель заносится в резольвенту.  fib(2, Result).  Запуск процесса редукции. | Попытка унификации:  fib(2, Result) = factorial(N, F)  Результат: неудача, различные имена главных функторов | Прямой ход, переход к следующему предложению |
| 2-3 | fib(2, Result) Выполнение процесса редукции. | Аналогично предыдущему пункту для следующих 2ух термов | Прямой ход, переход к следующему предложению |
| 4 | fib(2, Result) Выполнение процесса редукции. | Попытка унификации:  fib(2, Result) = fib(N, Res)  Результат: успех, подстановка: {N=2, Result=Res} | Прямой ход  Преобразование резольвенты: замена текущей цели на тело найденного с помощью унификации правила, применение полученной подстановки |
| 5 | fib\_rec(2, Res, 1, 1) Выполнение процесса редукции. | Попытка унификации:  fib\_rec(2, Res, 1, 1) = factorial(N, F)  Результат: неудача, различные имена главных функторов | Прямой ход, переход к следующему предложению |
| 6-8 | fib\_rec(2, Res, 1, 1) Выполнение процесса редукции. | Аналогично предыдущему пункту для следующих 3-х термов. | Прямой ход, переход к следующему предложению |
| 9 | fib\_rec(2, Res, 1, 1) Выполнение процесса редукции. | Попытка унификации:  fib\_rec(2, Res, 1, 1) = fib\_rec(1, Res, \_, Res)  Результат: неудача, различные константы | Прямой ход, переход к следующему предложению |
| 10 | fib\_rec(2, Res, 1, 1) Выполнение процесса редукции. | Попытка унификации:  fib\_rec(2, Res, 1, 1) = fib\_rec(N, Res, N1, N2)  Результат: успех, подстановка: {N=2, N1=1, N2=1} | Прямой ход  Преобразование резольвенты: замена текущей цели на тело найденного с помощью унификации правила, применение полученной подстановки |
| 11 | Tmp\_N = 2 - 1,  Tmp\_N2 = 1 + 1,  fib\_rec(Tmp\_N, Res, 1, Tmp\_N2)  Запуск алгоритма редукции для верхней подцели. | Попытка унификации:  Tmp\_N = 2 - 1  Результат: успех, подстановка: {Tmp\_N=1} | Прямой ход |
| 12 | Tmp\_N2 = 1 + 1,  fib\_rec(1, Res, 1, Tmp\_N2)  Запуск алгоритма редукции для верхней подцели. | Попытка унификации:  Tmp\_N2 = 1 + 1Результат: успех, подстановка: {Tmp\_N2=2} | Прямой ход |
| 13 | fib\_rec(1, Res, 1, 2)  Запуск алгоритма редукции для верхней подцели. | Попытка унификации:  fib\_rec(1, Res, 1, 2) = factorial(N, F)  Результат: неудача, различные имена главных функторов | Прямой ход, переход к следующему предложению |
| 14-16 | fib\_rec(1, Res, 1, 2)  Запуск алгоритма редукции для верхней подцели. | Аналогично предыдущему пункту для следующих 3-х термов. | Прямой ход, переход к следующему предложению |
| 17 | fib\_rec(1, Res, 1, 2)  Выполнение процесса редукции. | Попытка унификации:  fib\_rec(1, Res, 1, 2) = fib\_rec(1, Res, \_, Res)  Результат: успех, подстановка: {Res=2} | Прямой ход  Преобразование резольвенты: замена текущей цели на тело найденного с помощью унификации правила, применение полученной подстановки |
| 18 | !.  Выбор верхней подцели резольвенты, запуск редукции. | Выполнение отсечение | Завершение работы программы. Резольвента пуста. Result=2 |