|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **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»

**ОТЧЕТ**

*к лабораторной работе № 18*

*По курсу: «Функциональное и логическое программирование»*

Студент ИУ7-65Б

Юмаев А.Р.

Преподаватель

Строганов Ю.В.

Толпинская Н.Б

*Москва, 2020 г.*

**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

**Ответы на вопросы**

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

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

Одним из способов организации повторых вычислений является рекурсия. Если говорить о логическом программировании, то рекурсией называется способ заставить систему много раз использовать одну и ту же процедуру, так же, для корректности работы у рекурсии должен быть выход.

Организация хвостовой рекурсии:

* Рекурсивный вызов должен быть расположен к конце тела правила
* Рекурсивный вызов должен быть единственным
* До вычисления рекурсивного вызова не должно быть точек отката (этого можно достичь, например, поставив предикат отсечения перед рекурсивным вызовом).

Для выхода из рекурсии используется отдельное правило, в конце которого может находиться предикат отсечения.

* **Какое первое состояние резольвенты?**

На первом шаге в резольвенте находится заданный вопрос (цель).

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

Система запускает алгоритм унификации, в случае, если есть, что доказывать. Формально, алгоритм унификации запускается для выбранной (по принципу стека) подцели в резольвенте и сопоставляет текущую цель с предложениями в БЗ.

Для поиска ответа на вопрос системе необходимо найти подходящее знание в БЗ, для поиска такого знания используется алгоритм унификации. Формально, он помогает системе понять, что заголовок подошел: алгоритм попарно пытается сопоставить термы (текущую цель и термы из БЗ) и построить для них общий пример (для этого используется подстановка).

Алгоритм унификации может завершиться успехом и неудачей, если в результат – успех – в результирующей ячейке сформируется ниаболее общий унификатор (подстановка), в качестве побочного эффекта.

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

Именованная переменная уникальна в рамках предложения, в котором она используются. Любая анонимная переменная является уникальной.

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

Применение подстановки заключается в замене каждого вхождения переменной на соответствующий терм (). В результате применения подстановки переменные конкретизируются значениями, которые будут далее использованы при доказательстве истинности тела выбранного правила то есть значения переменных переходят на следующих шаг доказательства.

* **Как изменяется резольвента?**

В резольвенте хранятся цели, истинность которых необходимо доказать. Для хранения целей в резольвенте используется стек. Она (резольвента) меняется в ходе доказательства с помощью алгоритма редукции (замена текущей цели на тело правила, найденного в БЗ с помощью алгоритма унификации). Преобразование происходит следующим образом:

* берется верхняя цель и заменяется на тело правила (если речь идет о факте, так как факт - частный случай правила, не имеющий тела, цель просто убирается), найденного путем сопоставления алгоритмом унификации из БЗ.
* к полученной конъюнкции целей применяется подстановка, полученная алгоритмом унификации
* **В каких случаях запускается механизм отката?**

Во время работы системы, в случае, если решение не найдено, и из данного состояния невозможен переход в новое состояние (тупиковое состояние), применяется механизм отката. Также для поиска альтернативных решений (резольвента пуста, но не все правила были рассмотрены).

|  |
| --- |
| predicates  f(integer, integer, integer)  factorial(integer, integer)    fibonacci(integer, integer)  fib(integer, integer, integer, integer)  clauses  factorial(N,F) :- f(F,N,N).  f(F,1,F) :- !.  f(F,N1,F1) :- NewN = N1 - 1,  NewF = F1 \* NewN,  f(F,NewN,NewF).     fibonacci(N,F) :-fib(F,N,1,0).   fib(F,1,F,\_) :- !.  fib(F,I,Pr1,Pr2) :-  I1 = I - 1,  NewPr1 = Pr1 + Pr2,  NewPr2 = Pr1,  fib(F,I1,NewPr1,NewPr2). |

**Описание параметров программы**

1. Factorial(**N, F**)

* N – число, факторого которого ищется.
* F – переменная для связывания со значением результата

1. f(**F, N1, F1**)

* F – переменная для связывания со значением результата
* N1 **–** счетчик, на первом шаге равен N, нужен для организации условия выхода из рекурсии.
* F1 – переменная для связывания со значением текущего результата

1. fibonacci(**N, F**)

* N – номер вычисляемого числа Фибоначчи
* F – переменная для связывания со значением результата

1. fib(**F, I, Pr1, Pr2**)

* F – переменная для связывания со значением результата
* I – счетчик, на первом шаге равен N, нужен для организации условия выхода из рекурсии.
* Pr1, Pr2 – аргументы, соответсвующие значениям чисел фибоначчи на двух предыдуших шагах. (нужно для вычисления).

**Примеры целей и результатов работы программы**

**Цель:** factorial(2, F).

**Результат:** **F**=2

**Цель:** fibonacci(2, F).

**Результат:** **F**=1

**Описание порядка поиска объектов**

**Цель:**  factorial(2, F).

|  |  |  |  |
| --- | --- | --- | --- |
| № 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 | **Сотояние резольвенты:**  ТЦ: factorial(2, F).  **Дальнейшие действия:**  Запуск алгоритма унификации для верхней подцели factorial(2, F). | **Попытка унификации:**  ТЦ: factorial(2, F).  factorial(N,F) (знание из БЗ)  **Результат:** успех  **Подстановка:** {N=2, F=F} | Прямой ход, преобразование резольвенты (замена текущей подцели на тело правила, найденного алгоритмом унификации + применение полученной постановки) |
| 2 | **Сотояние резольвенты:**  ТЦ: f(F,2,2).  **Дальнейшие действия:**  Запуск редукции для верхней подцели f(F, 2, 2). | **Попытка унификации:**  ТЦ: f(F,2,2).  factorial(N,F) (знание из БЗ)  **Результат:** неудача, разные имена главных функторов. | Прямой ход, переход к следующему предложению. |
|  | **Сотояние резольвенты:**  ТЦ: f(F,2,2).  **Дальнейшие действия:**  Выполнение алгоритма унификации для верхней подцели f(F, 2, 2). | **Попытка унификации:**  ТЦ: f(F,2,2).  f(F,1,F) (знание из БЗ)  **Результат:** неудача, разные константы. | Прямой ход, переход к следующему предложению. |
|  | **Сотояние резольвенты:**  ТЦ: f(F,2,2).  **Дальнейшие действия:**  Выполнение алгоритма унификации для верхней подцели f(F, 2, 2). | **Попытка унификации:**  ТЦ: f(F,2,2).  f(F,N1,F1) (знание из БЗ)  **Результат:** успех  **Подстановка:** {F=F, N1=2, F1=2}. | Прямой ход, преобразование резольвенты (замена текущей подцели на тело правила, найденного алгоритмом унификации + применение полученной постановки) |
| 3 | **Сотояние резольвенты:**  ТЦ: NewN = 2 - 1,  NewF = 2 \* NewN,  f(F,NewN,NewF).  **Дальнейшие действия:**  Запуск алгоритма унификации для верхней подцели | **Попытка унификации:**  ТЦ: NewN = 2 - 1  **Результат:** успех  **Подстановка:** {NewN=1} | Прямой ход, преобразование резольвенты (замена текущей подцели на тело правила, найденного алгоритмом унификации + применение полученной постановки) |
| 4 | **Сотояние резольвенты:**  ТЦ: NewF = 2 \* 1,  f(F,1,NewF).  **Дальнейшие действия:**  Запуск алгоритма унификации для верхней подцели | **Попытка унификации:**  ТЦ: NewF = 2 \* 1  **Результат:** успех  **Подстановка:** {NewF=2} | Прямой ход, преобразование резольвенты (замена текущей подцели на тело правила, найденного алгоритмом унификации + применение полученной постановки) |
| 5 | **Сотояние резольвенты:**  ТЦ: f(F,1,2).  **Дальнейшие действия:**  Запуск алгоритма унификации для верхней подцели | **Попытка унификации:**  ТЦ: f(F,1,2).  factorial(N,F) (знание из БЗ)  **Результат:** неудача, разные имена главных функторов | Прямой ход, переход к следующему предложению. |
|  | **Сотояние резольвенты:**  ТЦ: f(F,1,2).  **Дальнейшие действия:**  Выполнение алгоритма унификации для верхней подцели | **Попытка унификации:**  ТЦ: f(F,1,2).  f(F,1,F) (знание из БЗ)  **Результат:** успех  **Подстановка:** {F=2} | Прямой ход, преобразование резольвенты (замена текущей подцели на тело правила, найденного алгоритмом унификации + применение полученной постановки) |
| 6 | **Сотояние резольвенты:**  ТЦ: ! | Выполнение отсечения | Вывод, завершение работы программы (резольвента пуста) |

**Цель:**  fibonacci(2, F).

|  |  |  |  |
| --- | --- | --- | --- |
| № 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 | **Сотояние резольвенты:**  ТЦ: fibonacci(2, F).  **Дальнейшие действия:**  Запуск алгоритма унификации для ТЦ | **Попытка унификации:**  ТЦ: fibonacci(2, F).  и термов из текущей БЗ пока не конец БЗ, либо не найден унифицируемый терм.  **Результат:** неудача | Прямой ход, переход к следующему предложению |
|  | **Сотояние резольвенты:**  ТЦ: fibonacci(2, F). | **Попытка унификации:**  ТЦ: fibonacci(2, F).  fibonacci(N,F) (знание из БЗ)  **Результат:** успех  **Подстановка:** {N=2, F=F} | Прямой ход, преобразование резольвенты (замена текущей подцели на тело правила, найденного алгоритмом унификации + применение полученной постановки) |
| 2 | **Сотояние резольвенты:**  ТЦ: fib(F,2,1,0).  **Дальнейшие действия:**  Запуск алгоритма унификации для ТЦ | **Попытка унификации:**  ТЦ: fib(F,2,1,0).  и термов из текущей БЗ пока не конец БЗ, либо не найден унифицируемый терм.  **Результат:** неудача | Прямой ход, переход к следующему предложению |
|  | **Сотояние резольвенты:**  ТЦ: fib(F,2,1,0). | **Попытка унификации:**  ТЦ: fib(F,2,1,0).  fib(F,1,F,\_) (знание из БЗ)  **Результат:** неудача, несовпадающие константы | Прямой ход, переход к следующему предложению |
|  | **Сотояние резольвенты:**  ТЦ: fib(F,2,1,0). | **Попытка унификации:**  ТЦ: fib(F,2,1,0).  fib(F,I,Pr1,Pr2) (знание из БЗ)  **Результат:** успех  **Подстановка:** {F=F, I=2, Pr1=1, Pr2=0} | Прямой ход, преобразование резольвенты (замена текущей подцели на тело правила, найденного алгоритмом унификации + применение полученной постановки) |
| 3 | **Сотояние резольвенты:**  ТЦ: I1 = 2 - 1,  NewPr1 = 1 + 0,  NewPr2 = 1, fib(F,I1,NewPr1,NewPr2).  **Дальнейшие действия:**  Запуск алгоритма унификации для ТЦ | **Попытка унификации:**  ТЦ: I1 = 2 - 1  **Результат:** успех  **Подстановка:** {I1=1} | Прямой ход, преобразование резольвенты (замена текущей подцели на тело правила, найденного алгоритмом унификации + применение полученной постановки) |
| 4 | **Сотояние резольвенты:**  ТЦ: NewPr1 = 1 + 0,  NewPr2 = 1, fib(F,1,NewPr1,NewPr2).  **Дальнейшие действия:**  Запуск алгоритма унификации для ТЦ | **Попытка унификации:**  ТЦ: NewPr1 = 1 + 0  **Результат:** успех  **Подстановка:** { NewPr1 = 1 } | Прямой ход, преобразование резольвенты (замена текущей подцели на тело правила, найденного алгоритмом унификации + применение полученной постановки) |
| 5 | **Сотояние резольвенты:**  ТЦ: NewPr2 = 1,  fib(F,1,1,NewPr2).  **Дальнейшие действия:**  Запуск алгоритма унификации для ТЦ | **Попытка унификации:**  ТЦ: NewPr2 = 1  **Результат:** успех  **Подстановка:** { NewPr2 = 1} | Прямой ход, преобразование резольвенты (замена текущей подцели на тело правила, найденного алгоритмом унификации + применение полученной постановки) |
| 6 | **Сотояние резольвенты:**  ТЦ: fib(F,1,1,1).  **Дальнейшие действия:**  Запуск алгоритма унификации для ТЦ | **Попытка унификации:**  ТЦ: fib(F,1,1,1).  и термов из текущей БЗ пока не конец БЗ, либо не найден унифицируемый терм.  **Результат:** неудача | Прямой ход, переход к следующему предложению |
|  | **Сотояние резольвенты:**  ТЦ: fib(F,1,1,1). | **Попытка унификации:**  ТЦ: fib(F,1,1,1)  fib(F,1,F,\_) (знание из БЗ)  **Результат:** успех  **Подстановка:** {F=1} | Прямой ход, преобразование резольвенты (замена текущей подцели на тело правила, найденного алгоритмом унификации + применение полученной постановки) |
|  | **Сотояние резольвенты:**  ТЦ: **!** | **Выполнение отсечения** | Вывод, завершение работы программы (резольвента пуста) |

**Выводы**

Для повышения эффетивности программы на пролог, можно использовать следующее:

* Предикат отсечения, с целью избавиться от лишних вычислений, как, например, при поиске максимального/минимального числа (взаимоисключающие правила)
* При использовании отсечения, мне кажется, порядок следования предложений в БЗ также может помочь в повышении эффективности
* Хвостовая рекурсия, которая в отличии от обычной рекурсии, не тратит дополнительно память.