



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 ИУ7-64Б
Лозовский А.А.

Преподаватель
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.

Ответы на вопросы

1. 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.

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

2. Какое первое состояние резольвенты?

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

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

Алгоритм унификации формализует процесс логического вывода. Используется для нахождения «нужного» для доказательства истинности (выводимости) правила в данный момент времени. Алгоритм унификации может завершиться успехом, тогда в результирующей ячейке формируется подстановка, содержащая значения переменных, при которых вопрос станет примером программы, в качестве побочного эффекта строится наибольший общий терм. Также алгоритм может завершиться неудачей (тупиковая ситуация).

4. В каких пределах программы переменные уникальны?

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

5. Как применяется подстановка, полученная с помощью алгоритма унификации?

Полученная с помощью алгоритма унификации подстановка применяется к целям в резольвенте. Подстановка применяется путем замены именованной переменной на соответствующий терм.

6. Как изменяется резольвента?

Для хранения резольвенты используется стек. Меняется она в ходе доказательства. Преобразования выполняются с помощью редукции – замены текущей цели на тело найденного в программе правила (с помощью унификации цели и заголовка правила программы). Преобразование резольвенты разделено на два этапа:

- 1) Берется верхняя из подцелей резольвенты (по стековому принципу) и заменяется на тело правила, найденного в программе.
- 2) Затем к полученной конъюнкции целей применяется подстановка (наибольший общий унификатор цели и сопоставленного с ней правила).

7. 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).

domains

int = integer

predicates

fact(int, int, int, int)
factorial(int, int)

fibonacci(int, int)
fib(int, int, int, int)

clauses

/ Обертка */*

factorial(N,Res) :- fact(N, Res,1,1).

*/*Условие выхода */*

fact(N,Res,N, Res) :- !.

*/*Основной шаг процедуры*/*

fact(N, Res,TmpN,TmpF) :- NewN = TmpN + 1,
NewF = TmpF * NewN,
fact(N,Res,NewN,NewF).

/ Проверка на корректный запрос и обертка */*

fibonacci(N, Res):- N < 1, !, fail.
fibonacci(N, Res):- fib(N, 1, 0, Res).

*/*Условие выхода */*

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

*/*Основной шаг процедуры*/*

fib(N, Prev1, Prev2, Res) :- NewN = N - 1,
NewPrev2 = Prev1,
NewPrev1 = Prev1 + Prev2,
fib(NewN, NewPrev1, NewPrev2, Res).

Описание параметров программы

- Factorial(N, Res)
 - N – аргумент, факториал которого нужно найти
 - Res – аргумент-результат.
- fact(N, Res,TmpN,TmpF)
 - N – аргумент, факториал которого нужно найти
 - Res – аргумент-результат.

- TmpN – счетчик, принимающий значения от 1 до N. Нужен для выхода из рекурсии и для вычисления факториала.
- TmpF – аргумент для хранения накапливаемого значения факториала.
- fibonacci(N, Res)
 - N – аргумент-номер числа фибоначчи, которое нужно вычислить
 - Res – аргумент-результат.
- fib(N, Prev1, Prev2, Res)
 - N – аргумент-номер числа фибоначчи, которое нужно вычислить
 - Res – аргумент-результат
 - Prev1, Prev2 – аргументы, соответствующие значениям чисел фибоначчи на двух предыдущих шагах. (нужно для вычисления).

Примеры целей и результатов работы программы

1. **Goal factorial(3, Res).**
Result Res=6
2. **Goal fibonacci(3, Res).**
Result Res=2
3. **Goal fibonacci(3, 2).**
Result yes
4. **Goal factorial(3, 6).**
Result yes

Описание порядка поиска объектов

Цель: factorial(2, Res).

№ 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	Состояние резольвенты: factorial(2, Res) Дальнейшие действия: Запуск алгоритма редукции		
1.1	Состояние резольвенты: Пустая Дальнейшие действия: Выполнение редукции	Попытка унификации: factorial(2, Res) – текущая цель factorial(N,Res) - предложение БЗ. Результат: успех Подстановка: {N=2, Res=Res}	Прямой ход, переход в тело правила. Преобразование резольвенты путем замены рассматриваемой цели на тело найденного алгоритмом унификации правила.
2	Состояние резольвенты: fact(2, Res,1,1). Дальнейшие действия: Запуск алгоритма редукции		
2.1	Состояние резольвенты: Пустая Дальнейшие действия: Выполнение редукции	Попытка унификации: fact(2, Res,1,1) – текущая цель factorial(N,Res) - предложение БЗ. Результат: неудача, разные имена главных функторов.	Прямой ход, переход к следующему предложению в БЗ.

2.2	Состояние резольвенты: Пустая Дальнейшие действия: Выполнение редукции	Попытка унификации: fact(2, Res, 1, 1) fact(N, Res, N, Res) Результат: неудача, разные константы.	Прямой ход, переход к следующему предложению в БЗ.
2.3	Состояние резольвенты: Пустая Дальнейшие действия: Выполнение редукции	Попытка унификации: fact(2, Res, 1, 1) – текущая цель fact(N, Res, TmpN, TmpF) - предложение БЗ. Результат: успех Подстановка: { N=2, Res=Res, TmpN=1, TmpF=1 }	Прямой ход, переход в тело правила. Преобразование резольвенты путем замены рассматриваемой цели на тело найденного алгоритмом унификации правила.
3	Состояние резольвенты: NewN = 1 + 1, NewF = 1 * NewN, fact(2, Res, NewN, NewF). Дальнейшие действия: Запуск алгоритма редукции для верхней подцели.	Попытка унификации: NewN = 1 + 1 Результат: успех Подстановка: { NewN = 2 }	Прямой ход, применение подстановки к резольвенте
4	Состояние резольвенты: NewF = 1 * 2, fact(2, Res, 2, NewF). Дальнейшие действия: Запуск алгоритма редукции для верхней подцели.	Попытка унификации: NewF = 1 * 2 Результат: успех Подстановка { NewF=2 }	Прямой ход, применение подстановки к резольвенте
5	Состояние резольвенты: fact(2, Res, 2, 2). Дальнейшие действия: Запуск алгоритма редукции для верхней подцели.	Попытка унификации: fact(2, Res, 2, 2) factorial(N, Res) Результат: неудача, разные имена главных функторов.	Прямой ход, переход к следующему предложению в БЗ.
5.1	Состояние резольвенты: Пуста Дальнейшие действия: выполнение алгоритма редукции	Попытка унификации: fact(2, Res, 2, 2) fact(N, Res, N, Res) Результат: успех Подстановка: { N=2, Res=Res, N=2, Res=2 }	Прямой ход, переход в тело правила.
6	Состояние резольвенты: ! Дальнейшие действия: Применить редукцию к верхней подцели.	Выполнение отсечения.	Завершение работы программы. Цели выполнены, резольвента пуста. Res=2

Цель: fibonacci(2, Res).

№ 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	Состояние резольвенты: fibonacci(2, Res) Дальнейшие действия: Запуск алгоритма редукции		

1.1	Состояние резольвенты: Пуста Дальнейшие действия: Выполнение редукции для fibonacci(2, Res)	Попытка унификации: fibonacci(2, Res) factorial(N,F) Результат: неудача, разные имена главных функторов	Прямой ход, переход к следующему предложению
1.2	Состояние резольвенты: Пуста Выполнение редукции для fibonacci(2, Res)	Попытка унификации: fibonacci(2, Res) fact(N,F,N,F) Результат: неудача, разные имена главных функторов	Прямой ход, переход к следующему предложению
1.3	Состояние резольвенты: Пуста Выполнение редукции для fibonacci(2, Res)	Попытка унификации: fibonacci(2, Res) fact(N,F,TmpN,TmpF) Результат: неудача, разные имена главных функторов	Прямой ход, переход к следующему предложению
1.4	Состояние резольвенты: Пуста Выполнение редукции для fibonacci(2, Res)	Попытка унификации: fibonacci(2, Res) fibonacci(N, _) Результат: Успех Подстановка: {N=2}	Прямой ход, преобразование резольвенты, замена текущей цели на тело правила, найденного алгоритмом унификации с применением подстановки к полученной конъюнкции подцелей.
2	Состояние резольвенты: 2 < 1, !, fail. Запуск алгоритма редукции для верхней подцели.	Попытка унификации: 2 < 1 Результат: неудача.	Прямой ход, откат
3	Состояние резольвенты: fibonacci(2, Res) Выполнение редукции для fibonacci(2, Res)	Попытка унификации: fibonacci(2, Res) fibonacci(N, Res) Результат: успех Подстановка: {N=2, Res=Res}	Прямой ход, преобразование резольвенты, замена текущей цели на тело правила, найденного алгоритмом унификации с применением подстановки к полученной конъюнкции подцелей.
4	Состояние резольвенты: fib(2, 1, 0, Res) Запуск алгоритма редукции для fib(2, 1, 0, Res).	Попытка унификации: fib(2, 1, 0, Res) (по аналогии с шагами 1.1-1.4) со всеми термами пока не будет пройдена вся БЗ либо пока не будет найден унифицируемый терм в текущей БЗ.	Прямой ход, переход к следующему предложению.
	Состояние резольвенты: Пуста Выполнение алгоритма редукции для fib(2, 1, 0, Res)	Попытка унификации: fib(2, 1, 0, Res) fib(1, Res, _, Res) результат: неудача, несовпадающие константы.	Прямой ход, переход к следующему предложению
	Состояние резольвенты: Пуста Выполнение алгоритма редукции для fib(2, 1, 0, Res)	Попытка унификации: fib(2, 1, 0, Res) fib(N, Prev1, Prev2, Res) результат: успех Подстановка: {N=2, Prev1=1, Prev2=0, Res=Res}	Прямой ход, преобразование резольвенты, замена текущей цели на тело правила, найденного алгоритмом унификации с применением подстановки к полученной конъюнкции подцелей.

5	Состояние резольвенты: NewN = 2 - 1, NewPrev2 = 1, NewPrev1 = 1 + 0, fib(NewN, NewPrev1, NewPrev2, Res). Выполнение редукции для NewN = 2 - 1	Попытка унификации: NewN = 2 - 1 Результат: успех Подстановка: {NewN=1}	Прямой ход, преобразование резольвенты применением подстановки
6	Состояние резольвенты: NewPrev2 = 1, NewPrev1 = 1 + 0, fib(1, NewPrev1, NewPrev2, Res). Выполнение редукции для NewPrev2 = 1	Попытка унификации: NewPrev2 = 1 Результат: успех Подстановка: {NewPrev2 = 1 }	Прямой ход, преобразование резольвенты применением подстановки
7	Состояние резольвенты: NewPrev1 = 1 + 0, fib(1, NewPrev1, 1, Res) Выполнение редукции для NewPrev1 = 1 + 0,	Попытка унификации: NewPrev1 = 1 + 0 Результат: успех Подстановка: {NewPrev1 = 1 }	Прямой ход, преобразование резольвенты применением подстановки
8	Состояние резольвенты: fib(1,1, 1, Res) Выполнение редукции для fib(1, 1, 1, Res)	Попытка унификации: fib(1, 1, 1, Res) (по аналогии с шагами 1.1-1.4) со всеми термами пока не будет пройдена вся БЗ либо пока не будет найден унифицируемый терм в текущей БЗ.	Прямой ход, переход к следующему предложению.
	Состояние резольвенты: Пустая Выполнение редукции для fib(1, 1, 1, Res)	Попытка унификации: fib(1, 1, 1, Res) fib(1, Res, _, Res) результат: успех подстановка: {Res=1, Res=Res} (я не знаю как отобразить корректно что Res будет связан со значением 1)	Прямой ход, преобразование резольвенты, замена текущей цели на тело правила, найденного алгоритмом унификации с применением подстановки к полученной конъюнкции подцелей.
9	Состояние резольвенты: ! Выполнение редукции для !	Выполнение отсечения	Завершение работы, резольвента пуста.

Выводы

Для повышения эффективности программы на пролог, можно использовать отсечения, чтобы ограничить количество вычислений, в случае, если они избыточны (например, как в случае с взаимоисключающими правилами), также можно использовать хвостовую рекурсию, ее главное отличие от стандартной реализации рекурсии в том, что при вычислениях, не возникает истощения памяти.