



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

Дисциплина Функциональное и логическое программирование

Тема Работа программы на Prolog

Студент Куприй А. А.

Группа ИУ7-63Б

Оценка (баллы) _____

Преподаватель Толпинская Н. Б., Строганов Ю. В.

Москва, 2020 г.

Цель работы – получить навыки построения модели предметной области, разработки и оформления программы на Prolog, изучить принципы, логику формирования программы и отдельные шаги выполнения программы на Prolog.

Задачи работы: приобрести навыки декларативного описания предметной области с использованием фактов и правил.

Изучить способы использования термов, переменных, фактов и правил в программе на Prolog, принципы и правила сопоставления и отождествления, порядок унификации.

Задание:

Используя хвостовую рекурсию, разработать программу, позволяющую найти

- $n!$,
- 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!

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.

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

Какое первое состояние резольвенты?

Заданный вопрос (goal).

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).

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 резольvente путем замены текущей переменной на соответствующий терм.

Как изменяется резольвента?

Преобразования резольventы 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 резольvente 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 запустится в случае неудачи алгоритма унификации.

Текст программы

predicates

```
factorial(integer, integer).  
factorial(integer, integer, integer).  
  
fibb(integer, integer).  
fibb(integer, integer, integer, integer, integer).  
sign(integer, integer).
```

clauses

```
factorial(N, -1) :- N < 0, !.  
factorial(0, 1) :- !.  
factorial(N, Result) :- factorial(N, 1, Result).  
  
factorial(1, Result, Result) :- !.  
factorial(N, Cur, Result) :-  
    NewN = N - 1,  
    NewMult = Cur * N,  
    factorial(NewN, NewMult, Result).  
  
fibb(0, 0) :- !.  
fibb(1, 1) :- !.  
fibb(N, Result) :-  
    N < 0,  
    NormalN = N * -1,  
    fibb(NormalN, PrevResult),  
    sign(NormalN, Sign),  
    Result = PrevResult * Sign, !.  
fibb(N, Result) :- fibb(N, 2, 0, 1, Result).  
  
fibb(N, N, Prev1, Prev2, Result) :- Result = Prev1 + Prev2, !.  
fibb(N, CurN, Prev1, Prev2, Result) :-  
    NewN = CurN + 1,  
    Next = Prev1 + Prev2,  
    fibb(N, NewN, Prev2, Next, Result).  
  
sign(N, 1) :- (N mod 2) = 1, !.  
sign(_, -1).
```

goal

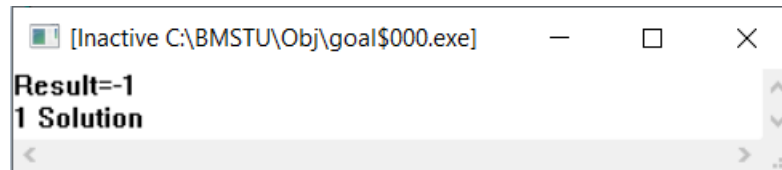
```
%factorial(-10, Result).  
%factorial(0, Result).  
factorial(5, Result).
```

```
%fibb(-9, Result).  
%fibb(-2, Result).  
%fibb(2, Result).  
%fibb(8, Result).
```

Результаты работы программы:

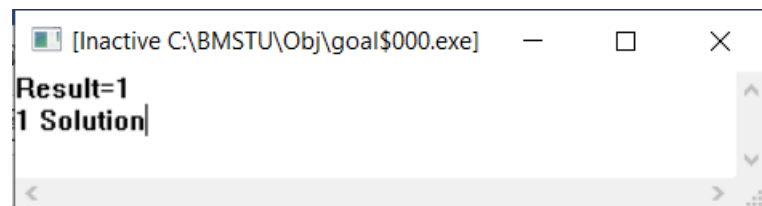
goal

factorial(-10, Result).



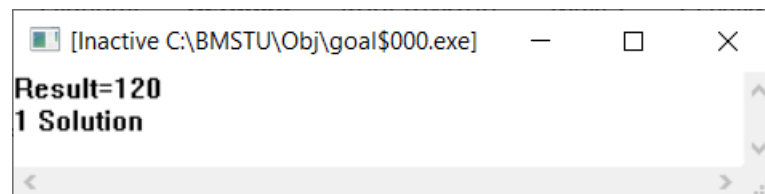
goal

factorial(0, Result).



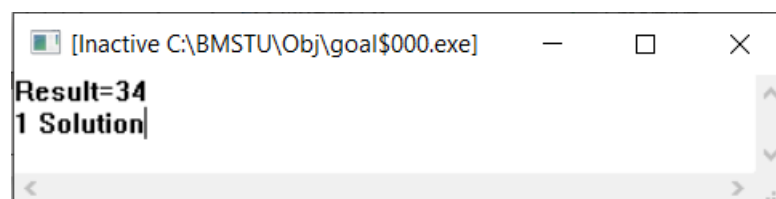
goal

factorial(5, Result).



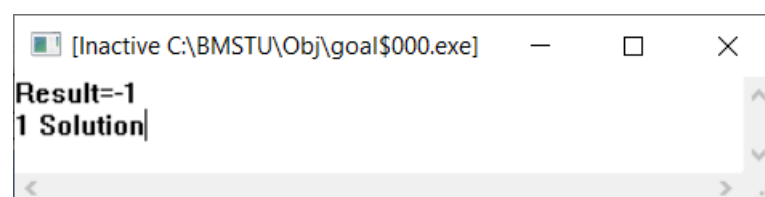
goal

fibb(-9, Result).



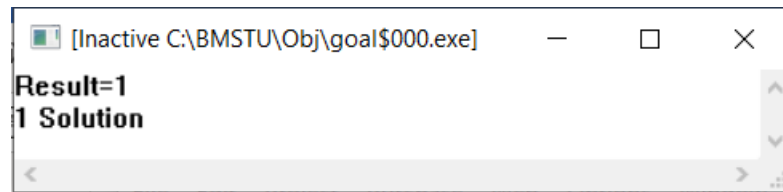
goal

fibb(-2, Result).



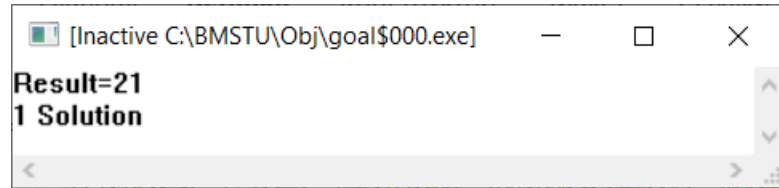
goal

fibb(2, Result).



goal

fibb(8, Result).



Таблицы

Цель:

goal

factorial(2, Result).

№ 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	factorial(2, Result).	factorial(2, Result) = factorial(N, -1) Успех N = 2 Result = -1	Прямой ход. Тело правила заносится в резольвенту.
2	N < 0 !	Сравнение: 2 < 0 Ложь	Откат к 1. Переход к следующему предложению
3	factorial(2, Result).	factorial(2, Result) = factorial(0, 1) Неудача	Переход к следующему предложению
4	factorial(2, Result).	factorial(2, Result) = factorial(N, Result) Успех N = 2 Result = Result	Прямой ход. Тело правила заносится в резольвенту.
5	factorial(N, 1, Result)	factorial(2, 1, Result) = factorial(1, Result, Result) Неудача	Переход к следующему предложению
6	factorial(N, 1, Result)	factorial(2, 1, Result) = factorial(N, Cur, Result) Успех N = 2 Cur = 1 Result = Result	Прямой ход. Тело правила заносится в резольвенту.
7	NewN = N – 1 NewMult = Cur * N factorial(NewN, NewMult, Result)	NewN = 2 – 1 = 1	Прямой ход.
8	NewMult = Cur * N factorial(NewN, NewMult, Result)	NewMult = 1 * 2 = 2	Прямой ход.

9	factorial(NewN, NewMult, Result)	factorial(1, 2, Result) = factorial(1, Result, Result) Успех Result = Result = 2	Прямой ход.
10			Резольвента пуста. Result = 2 Откат к 4. Конец процедуры factorial арности 2. Система завершает работу.

Цель:

goal

fib(2, Result).

№ 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	fibb(2, Result)	fibb(2, Result) = fibb(0, 0) Неудача	Переход к следующему предложению
2	fibb(2, Result)	fibb(2, Result) = fibb(1, 1) Неудача	Переход к следующему предложению
3	fibb(2, Result)	fibb(2, Result) = fibb(N, Result) Успех N = 2 Result = Result	Прямой ход. Тело правила заносится в резольвенту.
4	N < 0 NormalN = N * -1 fibb(NormalN, PreResult) sign(NormalN, Sign), Result = PreResult * Sign !	Сравнение: 2 < 0 Ложь	Откат к 3. Переход к следующему предложению
5	fibb(2, Result)	fibb(2, Result) = fibb(N, Result) Успех N = 2 Result = Result	Прямой ход. Тело правила заносится в резольвенту.
6	fibb(N, 2, 0, 1, Result)	fibb(2, 2, 0, 1, Result) = fibb(N, N, Prev1, Prev2, Result) Успех	Прямой ход. Тело правила заносится в резольвенту.

		Prev1 = 0 Prev2 = 1 Result = Result	
7	Result = Prev1 + Prev2 !	Result = 0 + 1 = 1	Прямой ход.
8	!		Найдено решение. Result = 1 Ввиду отсечения не будет попыток найти другие решения fibb(N, 2, 0, 1, Result). Откат к 5. Конец fibb арности 2. Система завершит свою работу.