



**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
Рекурсия на Prolog**

Студент Лучина Е.Д

Группа ИУ7-61Б

Преподаватель Толпинская Н.Б.

Москва.

май 2020 г.

Содержание отчета

[Полный текст задания!!!.](#)

[Ответы на вопросы.](#)

[Текст программы с комментариями обозначений и идентификаторов аргументов](#)

[Таблицы для каждого задания, демонстрирующие работу системы при одном из успешных вариантов вопроса.](#)

[Выводы: за счет чего достигнута эффективность работы системы.](#)

Задание

Ответить на вопросы:

1. Что такое рекурсия? Как организуется хвостовая рекурсия в Prolog? Как организовать выход из рекурсии в Prolog?
2. Какое первое состояние резольвенты?
3. В каком случае система запускает алгоритм унификации? Каково назначение использования алгоритма унификации? Каков результат работы алгоритма унификации?
4. В каких пределах программы переменные уникальны?
5. Как применяется подстановка, полученная с помощью алгоритма унификации?
6. Как изменяется резольвента?
7. В каких случаях запускается механизм отката?

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!

Шаг	Состояние резольвенты, и вывод: дальнейшие	Для каких термов запускается алгоритм унификации: $T1=T2$ и	Дальнейшие действия: прямой ход или откат (почему и к чему
-----	--	---	--

	действия (почему?)	каков результат (и подстановка)	приводит?)
1	Комментарий, вывод...
2

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. Рекурсия - ссылка на себя. Рекурсивная функция вызывает себя в своем теле.

В prolog рекурсия – это способ заставить систему использовать многократно одну и ту же процедуру. Будет реализовано рекурсивная процедура, которая в своем теле содержит вопрос того же предиката, что и заголовок. Тупик, остановка рекурсии достигается за счет предиката отсечения. Хвостовая рекурсия является самым эффективным способом организации рекурсии. Любая рекурсивная цель является последним термом в теле процедуры.

2. Какое первое состояние резольвенты?

Вопрос сформированный в goal.

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

Алгоритм унификации запускается, когда резольвента не пуста, программе еще есть на какие вопросы искать ответы. Назначение алгоритма унификации - подобрать знание, которое позволит на поставленный вопрос ответить да. Результат унификации может быть успешным (доказаны истинность утверждения) или неудачным (на основе данной базы знаний невозможно доказать утверждение). Побочным результатом унификации является конкретизация переменных.

4. В каких пределах программы переменные уникальны?

Именованные переменные уникальны в пределах предложения. Анонимные уникальны сами по себе в пределах всей программы.

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

Данная подстановка применяется к существующим в резольvente целям. Все вхождения переменных, о которых есть информация в подстановке, конкретизируются найденными значениями в дальнейших термах.

6. Как изменяется резольвента?

Преобразования резольвенты выполняются с помощью редукции - замены цели телом того правила из базы знаний, заголовок которого унифицируется с целью. Такие правила подбираются с помощью унификации и называются сопоставимыми с целью.

Новая резольвента образуется в два этапа

1. в текущей резольvente выбирается одна из подцелей (по стековому принципу - верхняя) и для неё выполняется редукция.
2. затем, к полученной конъюнкции целей применяется подстановка, полученная как наибольший общий унификатор цели и заголовка сопоставленного с ней правила.

Если подцель унифицируется с правилом, то число целей в резольvente уменьшается (факт - это правило с пустым телом). Если процесс унификации зашел в тупик осуществляется откат - восстановления предыдущих резольвент и значений переменных.

7. В каких случаях запускается механизм отката?

Если алгоритм унификации зашел в тупик или найдено решение, но оно не является гарантированно последним возможным.

Листинг программы

Факториал

```
predicates
    factorial(integer, integer)
clauses
    factorial(0, 1) :- !.
    factorial(X, Res) :- X < 0, !, fail.
    factorial(X, Res) :-
        X1 = X - 1, factorial(X1, R2), Res = X * R2.
```

Первый аргумент - n, второе - его факториал n!. Факториал нуля равен единице - первый терм в базе знаний. При успешной унификации вопроса с ним, программа свяжет переменную результата с единицей и завершится с единственным решением. Для отрицательного числа не определен факториал, проверим, что число отрицательное, завершим работу программы с неудачей. Fail - инициализация отката. Откат у пустой резольвенты при не имении решений дает "No Solution" в качестве результата. В случае

когда число положительно, рассчитаем факториал для значения на один меньше ($N1$), получим $R2$ и умножим число X на получившееся значение $R2$. Тупиком рекурсии является ситуация $X1 = 0$.

Примеры работы программы. Тестирование.

factorial(-5, Result).	No Solution
factorial(0, Result).	Result=1 1 Solution
factorial(5, Result).	Result=120 1 Solution
factorial(13, Result). /*13! = 6227020800 больше чем максимальное значение integer*/	PROGRAM ERROR. Module:OBJ\GOAL\$000.PRO Pos:476 Message:1031 Arithmetic overflow
factorial(3, 6).	yes
factorial(4, _).	yes
Переменная в качестве первого аргумента	Free variable in expression
Другие типы в качестве аргумента	Type error

n-ое число Фибоначчи

predicates fibonacci(integer, integer) clauses fibonacci(0, 0) :- !. fibonacci(1, 1) :- !. fibonacci(N, R) :- N < 0, !, fail. fibonacci(N, R) :- N1 = N - 1, N2 = N - 2, fibonacci(N1, R1), fibonacci(N2, R2), R = R1 + R2.
--

Поиск n-ого числа последовательности Фибоначчи. Последовательность определяется следующим рекуррентным выражением:

$$F_0 = 0; F_1 = 1; F_n = F_{n-1} + F_{n-2}, n \geq 2$$

fibonacci(0, 0) реализация $F_0 = 0$. fibonacci(1, 1) :- !. реализация первого члена последовательности. При успешной унификации с одним из этих выражений заканчивается вычисление, завершается поиск решений для конкретных аргументов - тупик рекурсии. Если порядковый номер отрицательный программа завершается с результатом "No Solution" на строчке fibonacci(N, R) :- N < 0, !, fail. Если порядковый номер больше или равен двум выполняется последнее правило. Вычисляется предыдущее ($N1$ - его порядковый номер) число фибоначчи - $R1$ и предпредпоследнее ($N2$ - его номер, $R1$ - его значение). Далее $R1$ и $R2$ складываются. Результат связывается с полученной суммой. Вся база знаний отмечено, завершение программы.

Примеры работы программы. Тестирование.

fibonacci(-5, Res).	No Solution
fibonacci(0, Res).	Res=0 1 Solution

fibonacci(1, Res).	Res=1 1 Solution
fibonacci(2, Res).	Res=1 1 Solution
fibonacci(5, Res).	Res=5 1 Solution
fibonacci(8, 21).	yes

Таблица

Таблица1 - factorial(3, R).

Шаг	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(3, R)	Неуспешная унификация factorial(3, R) != factorial(0, 1)	
		Успешная унификация factorial(3, R) == factorial(X, Res) {X = 3, Res = R}	Прямой ход
2	3<0, !, fail.	3 < 0 ложно	Откат R снова несвязанная переменная
3	factorial(3, R)	Успешная унификация factorial(3, R) == factorial(X, Res) {X = 3, Res = R}	Прямой ход
4	X1 = 3 - 1, factorial(X1, R2), R = 3 * R2	X1 = 3 - 1 {X1 = 2}	Прямой ход
5	factorial(2, R2), R = 3 * R2	Неуспешная унификация factorial(2, R2) != factorial(0, 1)	
		Успешная унификация factorial(2, R2) == factorial(X, Res) {X = 2, Res = R2}	Прямой ход
6	2<0, !, Fail. R = 3 * R2	2 < 0 Ложно	откат
7	factorial(2, R2),	Успешная унификация	Прямой ход

	R = 3 * R2	factorial(2, R2) == factorial(X, Res) {X = 2, Res = R2}	
8	X1 = 2 - 1, factorial(X1, R2), R2 = 2 * R2. R = 3 * R2	X1 = 2 - 1 {X1 = 1}	Прямой ход
9	factorial(1, R2), R2 = 2 * R2. R = 3 * R2	Неуспешная унификация factorial(1, R2) и factorial(0, 1)	
		Успешная унификация factorial(1, R2) и Factorial(X, Res) {X = 1, Res = R2}	Прямой ход
10	1 < 0, !, Fail. Res = 2 * R2. R = 3 * R2	1 < 0 ложно	откат
11	factorial(1, R2), R2 = 2 * R2. R = 3 * R2	Успешная унификация factorial(1, R2) и factorial(X, Res) {X = 1, Res = R2}	Прямой ход
12	X1 = 1 - 1, factorial(X1, R2), R2 = 1 * R2. R2 = 2 * R2. R = 3 * R2	X1 = 1 - 1 {X1 = 0}	Прямой ход
13	factorial(0, R2), R2 = 1 * R2. R2 = 2 * R2. R = 3 * R2	Успешная унификация factorial(0, R2) и factorial(0, 1) R2 = 1	Найдено решение Завершение цели в резольvente
14	!	Завершение поиска решений	
15	R2 = 1 * 1. R2 = 2 * R2. R = 3 * R2	R2 = 1 * 1 R2 = 1	Прямой ход Завершение цели в резольvente
16	R2 = 2 * 1. R = 3 * R2	R2 = 2 * 1 R2 = 2	Прямой ход. Завершение цели в резольvente
17	R = 3 * 2	R = 3 * 2	Прямой ход. Завершение цели в резольvente

		{R = 6}	
18	Резольвента пуста		Завершение работы программы R = 6

Таблица2 - fibonacci(3, R)

Шаг	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	fibonacci(3, R)	Неуспешные унификации - откаты $\text{fibonacci}(3, R) \neq \text{fibonacci}(0, 0)$ $\text{fibonacci}(3, R) \neq \text{fibonacci}(1, 1)$ Успешная унификация $\text{fibonacci}(3, R) = \text{fibonacci}(N, R)$ $\{N = 3\}$ - подстановка Прямой ход	
2	$3 < 0$, !, fail.	$3 < 0$ ложно	откат
3	fibonacci(3, R)	Успешная унификация $\text{fibonacci}(3, R)$ и $\text{fibonacci}(N, R)$ $\{N = 3, R = R\}$	Прямой ход
4	$N1 = 3 - 1$, $N2 = 3 - 2$, $\text{fibonacci}(N1, R1)$, $\text{fibonacci}(N2, R2)$, $R = R1 + R2$.	$N1 = 3 - 1$ $\{N1 = 2\}$	Прямой ход
5	$N2 = 3 - 2$, $\text{fibonacci}(2, R1)$, $\text{fibonacci}(N2, R2)$, $R = R1 + R2$.	$N2 = 3 - 2$ $\{N2 = 1\}$	Прямой ход
6	fibonacci(2, R1), fibonacci(1, R2), $R = R1 + R2$.	Неуспешные унификации - откаты $\text{fibonacci}(2, R1) \neq \text{fibonacci}(0, 0)$ $\text{fibonacci}(2, R1) \neq \text{fibonacci}(1, 1)$ Успешная унификация $\text{fibonacci}(2, R1) = \text{fibonacci}(N, R)$ $\{N = 2, R = R1\}$ - подстановка Прямой ход	
7	$2 < 0$, !, fail. fibonacci(1, R2),	$2 < 0$ ложно	откат

	$R = R1 + R2.$		
8	fibonacci(2, R1), fibonacci(1, R2), $R = R1 + R2.$	Успешная унификация $\text{fibonacci}(2, R1) =$ $\text{fibonacci}(N, R)$ $\{N = 2, R = R1\}$	Прямой ход
9	$N1 = 2 - 1,$ $N2 = 2 - 2,$ fibonacci(N1, R1), fibonacci(N2, R2), $R = R1 + R2$ fibonacci(1, R2), $R = R1 + R2.$	$N1 = 2 - 1$ $\{N1 = 1\}$	Прямой ход
10	$N2 = 2 - 2,$ fibonacci(1, R1), fibonacci(N2, R2), $R = R1 + R2$ fibonacci(1, R2), $R = R1 + R2.$	$N2 = 2 - 2$ $\{N2 = 0\}$	Прямой ход
11	fibonacci(1, R1), fibonacci(0, R2), $R = R1 + R2$ fibonacci(1, R2), $R = R1 + R2.$	Не успешная унификация $\text{fibonacci}(1, R1) =$ $\text{fibonacci}(0, 0)$	откат
		Успешная унификация $\text{fibonacci}(1, R1) =$ $\text{fibonacci}(1, 1)$ $\{R1 = 1\}$	Прямой ход
12	!. fibonacci(0, R2), $R = 1 + R2.$ fibonacci(1, R2), $R = R1 + R2.$	Завершение поиска решений для fibonacci(1, R1)	Завершение цели резольвенты
13	fibonacci(0, R2), $R = 1 + R2$ fibonacci(1, R2), $R = R1 + R2.$	Успешная унификация $\text{fibonacci}(0, R2) =$ $\text{fibonacci}(0, 0)$ $\{R2 = 0\}$	Прямой ход

14	!. $R = 1 + 0$ fibonacci(1, R2), $R = R1 + R2$.	Завершение поиска для fibonacci(0, R2)	Завершение цели резольвенты
15	$R = 1 + 0$ fibonacci(1, R2), $R = R1 + R2$.	$R = 1 + 0$ {R = 1}	Прямой ход {R1 = 1}
16	fibonacci(1, R2), $R = 1 + R2$.	Не успешная унификация fibonacci(1, R2) = fibonacci(0, 0)	откат
		Успешная унификация fibonacci(1, R2) = fibonacci(1, 1) {R2 = 1}	Прямой ход
17	!. $R = 1 + 1$.	Завершение поиска других решений для fibonacci(1, R2)	Прямой ход
18	$R = 1 + 1$.	$R = 2$	Прямой ход
19	Резольвента пуста	Все терты в БЗ отмечены	Завершение программы $R = 2$ 1 Solution

Выводы: за счет чего достигнута эффективность работы системы.

Эффективность достигнута за счет использования предикатов отсечения и fail. Они отсекают поиск решений в бесперспективных ветвях поиска. А в данном случае являются необходимыми тупиками рекурсии.