



**«Московский государственный технический
университет имени Н.Э. Баумана»
(МГТУ им. Н.Э. Баумана)**

Факультет «Информатика и системы управления»
Кафедра «Программное обеспечение ЭВМ и информационные технологии»

Отчет по лабораторной работе №18
по курсу:
«Функциональное и Логическое программирование»

Студент группы ИУ7-63Б: Фурдик Н. О.
(Фамилия И.О.)

Преподаватель: Толпинская Н. Б., Строганов Ю. В.
(Фамилия И.О.)

Оглавление

Постановка задачи	2
Листинг программы	3
Описание порядка поиска ответов	3
Ответы на вопросы	6
Список литературы	9

Постановка задачи

Используя хвостовую рекурсию, разработать программу, позволяющую найти

- 1) $n!$;
- 2) n -е число Фибоначчи.

Убедиться в правильности результатов. Для каждого случая пункта 2 обосновать необходимость всех условий тела. Для одного из вариантов ВОПРОСА и каждого варианта задания 2 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!

Листинг программы

Ниже представлен листинг программы:

```
domains

list = integer*.

predicates

len(list, integer length).
sum(list, integer sum).
odd_sum(list, integer sum, integer index).

clauses

len([], 0) :-!.
len([_|Tail], Length) :-len(Tail, Tail_Length), Length = Tail_Length + 1.

sum([], 0) :-!.
sum([Head|Tail], Sum) :-sum(Tail, Sum_), Sum = Sum_ + Head.

odd_sum([], 0, _) :-!.
odd_sum([Head|Tail], Sum, Index) :-Index mod 2 = 1, Next_Index = Index + 1,
    odd_sum(Tail, Sum_, Next_Index), Sum = Sum_ + Head.
odd_sum([_|Tail], Sum, Index) :-Index mod 2 = 0, Next_Index = Index + 1,
    odd_sum(Tail, Sum, Next_Index).

goal

len([1,21,3,4,5], Length).
%[1,21,3,4,5]
%sum of numbers
%sum([1,21,3,4,5], Sum).
%Sum=34
%1 Solution

%sum of odd
%odd_sum([1,21,3,4,5], Sum, 0).
%Sum=25
%1 Solution
```

Листинг 1: Задания 1 и 2

Описание работы системы

Ниже представлен алгоритм поиска ответов на вопросы `fact(2, Result)` и `fibb(2, Result)`.

Таблица 1: Описание работы системы при решении факториала

№ 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	Резольвента: $\text{fact}(2, \text{Result})$. Начинается поиск совпадений по БЗ		прямой ход
2	Резольвента: $\text{fact}(2, \text{Result})$.	Нашли подходящее правило: $\text{fact}(\text{Curr}, \text{Next}) :- \text{Cur} = \text{Curr} - 1, \text{fact}(\text{Cur}, \text{Res}), \text{Next} = \text{Res} * \text{Curr}$. Подставляем $\text{Curr} = 2$, $\text{Next} = \text{Result}$	прямой ход
3	Резольвента: $\text{Cur} = \text{Curr} - 1$, $\text{fact}(2, \text{Result})$.	Пробуем связать: $\text{Cur} = 2 - 1 = 1$, удача, идем дальше.	прямой ход, подставляем $\text{Cur} = 1$ в $\text{fact}(\text{Cur}, \text{Res})$
4	Резольвента: $\text{fact}(1, \text{Res})$, $\text{fact}(2, \text{Result})$.	Поиск совпадений по БЗ. Нашли подходящее правило: $\text{fact}(\text{Curr}, \text{Next}) :- \text{Cur} = \text{Curr} - 1, \text{fact}(\text{Cur}, \text{Res}), \text{Next} = \text{Res} * \text{Curr}$. Подставляем $\text{Curr} = 1$, $\text{Next} = \text{Res}$	прямой ход
5	Резольвента: $\text{Cur} = \text{Curr} - 1$, $\text{fact}(2, \text{Result})$.	Пробуем связать: $\text{Cur} = 1 - 1 = 0$, удача, идем дальше.	прямой ход, подставляем $\text{Cur} = 0$ в $\text{fact}(\text{Cur}, \text{Res})$
6	Резольвента: $\text{fact}(0, \text{Res})$, $\text{fact}(1, \text{Res})$, $\text{fact}(2, \text{Result})$.	Поиск совпадений по БЗ. Нашли подходящее правило: $\text{fact}(0, 1) :- !$. Подставляем $\text{Res} = 1$, один из вопросов решен.	прямой ход
7	Резольвента: $\text{fact}(1, \text{Res})$, $\text{fact}(2, \text{Result})$.	Пробуем связать: $\text{Next} = \text{Res} * \text{Curr} = 1 * 1$, ($\text{Res} = 1$ из предыдущего шага). Подставляем $\text{Res} = 1$, один из вопросов решен.	прямой ход
8	Резольвента: $\text{fact}(2, \text{Result})$.	Пробуем связать: $\text{Next} = \text{Res} * \text{Curr} = 1 * 2$, ($\text{Res} = 1$ из предыдущего шага). Подставляем $\text{Result} = \text{Next} = 2$, вопрос решен.	прямой ход
Вывод	$\text{Result} = 2$		

Таблица 2: Описание работы системы при решении факториала

№ 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	Резольвента: $fibb(2, Result)$. Начинается поиск совпадений по БЗ		прямой ход
2	Резольвента: $fibb(2, Result)$.	Нашли подходящее правило: $fibb(Curr, Next) :- Cur = Curr - 1, Prev = Cur - 1, fibb(Cur, Cur_Res), fibb(Prev, Prev_Res), Next = Cur_Res + Prev_Res$. Подставляем $Curr = 2, Nex \doteq Result$	прямой ход
3	Резольвента: $Cur = Curr - 1,$ $fact(2, Result)$.	Пробуем связать: $Cur = 2 - 1 = 1$, удача, идем дальше.	прямой ход, подставляем $Cur = 1$ в $fact(Cur, Res)$
4	Резольвента: $Prev = Cur - 1,$ $fact(2, Result)$.	Пробуем связать: $Prev = 1 - 1 = 0$, удача, идем дальше.	прямой ход, подставляем $Cur = 1$ в $fibb(Cur, Cur_Res)$
5	Резольвента: $fibb(1, Cur_Res)$ $fibb(2, Result)$.	Поиск совпадений по БЗ. Нашли подходящее правило: $fibb(1, 1) :- !$. Подставляем $Cur_Res = 1$, один из вопросов решен.	прямой ход, подставляем $Prev = 0$ в $fact(Prev, Prev_Res)$
6	Резольвента: $fibb(0, Prev_Res)$ $fibb(2, Result)$.	Поиск совпадений по БЗ. Нашли подходящее правило: $fibb(0, 0) :- !$. Подставляем $Prev_Res = 0$, один из вопросов решен.	прямой ход
8	Резольвента: $fibb(2, Result)$.	Пробуем связать: $Next = Cur_Res + Prev_Res = 1 + 0$. Подставляем $Result = Next = 1$, вопрос решен.	прямой ход
Вывод	$Result = 1$		

Ответы на вопросы

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

Рекурсия – это один из способов организации повторных вычислений.

Для осуществления хвостовой рекурсии рекурсивный вызов определяемого предиката должен быть последней подцелью в теле рекурсивного правила и к моменту рекурсивного вызова не должно остаться точек возврата (непроверенных альтернатив).

Параметры должны изменяться на каждом шаге так, чтобы в итоге либо сработал базис рекурсии, либо условие выхода из рекурсии, размещенное в самом правиле.

- 2) **Какое первое состояние резольвенты?**

Изначально в резольвенте находится вопрос.

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

Система запускает унификацию в том случае, если ей был задан вопрос. Унификация вопроса и первого предложения базы знаний происходит на первом шаге работы программы.

Алгоритм унификации необходим для попытки "увидеть одинаковость" – сопоставимость двух термов, может завершаться успехом или тупиковой ситуацией.

Результат алгоритма унификации – ответ «да» или «нет».

- 4) **В каких пределах программы уникальны переменные?**

Именованные переменные уникальны в рамках одного предложения. Анонимная переменная уникальна всегда. Переменные предназначены для передачи значений «во времени и в пространстве».

- 5) **Как применяется подстановка, полученная с помощью алгоритма унификации?**

Пока стек не пуст – **цикл**:

- считать из стека в рабочую область очередное равенство $S=T$
- обработать считанное по правилам:
 - если S и T несовпадающие константы, то неудача=1, и выход из цикла
 - если одинаковые константы то следующий шаг цикла
 - если S переменная и T терм содержащий S , то неудача=1, и выход из цикла
 - если S переменная и T терм НЕ содержащий S , то отыскать в стеке и в результирующей ячейке все вхождения S и заменить на T . Добавить в результирующую ячейку равенство $S=T$. Следующий шаг цикла
 - если S и T составные термы с разными функторами или разными арностями, то неудача=1, выход из цикла
 - если S и T составные термы с одинаковыми функторами и арностью:
 $S = f(s_1, s_2, \dots, s_m); T = f(t_1, t_2, \dots, t_m)$, то занести в стек равенство
 $S_1 = T_1, S_2 = T_2, \dots, S_m = T_m$.
- очистить рабочее поле
- **конец цикла**

6) Как меняется резольвента?

На каждом шаге имеется некоторая совокупность целей - утверждений, истинность (выводимость) которых надо доказать. Эта совокупность называется резольвентой - её состояние меняется в процессе доказательства (Для хранения резольвенты система использует стек). Новая резольвента образуется в два этапа:

- в текущей резольвенте выбирается одна из подцелей (по стековому принципу - верхняя) и для неё выполняется редукция - замена подцели на тело найденного (подобранного, если удалось) правила (а как подбирается правило?),
- затем, к полученной конъюнкции целей применяется подстановка, полученная как наибольший общий унификатор цели (выбранной) и заголовок сопоставленного с ней правила.

7) В каком случае запускается механизм отката?

Механизм отката запускается в 2 случаях:

- Если алгоритм попал в тупиковую ситуацию.
- Если резольвента не пуста и решение найдено, но в базе знания остались не отмеченные предложения.

Литература

1. Краснова, Л.П.. Бухгалтерский учет [Текст]: учебник для вузов / Краснова, Л.П., Н.Т. Шалашова, Н.М. Ярцева — Москва: Юристъ, 2001. — 550 с.
2. Книга художника: от миллионных тиражей – к единичным экземплярам. Л. Шпринц. [Электронный ресурс]: / Л. Шпринц.. — Электрон. текстовые дан.. — Москва: [б.и.], 2000. — Режим доступа: <http://atbook.km.ru/news/000525.html>, свободный.
3. О правительственной комиссии по проведению административной реформы [Текст]: постановление Правительства РФ от 31 июля 2003 г. № 451 // Собрание законодательства. — 2003. — № 31. — Ст. 3150.