



**«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 №17
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	8

Постановка задачи

В одной программе написать правила, позволяющие найти

1) Максимум из двух чисел

- без использования отсечения;
- с использованием отсечения.

2) Максимум из трех чисел

- без использования отсечения;
- с использованием отсечения.

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:

```
predicates

max_of_two(integer max, integer num1, integer num2)
max_of_three(integer max, integer num1, integer num2, integer num3)

clauses

%without
max_of_two(A, B, C) :-C >= B, A = C.
max_of_two(A, B, C) :-B > C, A = B.

%with
max_of_two(A, B, C) :-C >= B, A = C, !.
max_of_two(A, B, C) :-B >= C, A = B.

%without
max_of_three(A, B, C, D) :-C > D, C >= B, A = C.
max_of_three(A, B, C, D) :-B >= D, B > C, A = B.
max_of_three(A, B, C, D) :-D >= B, D >= C, A = D.

%without
max_of_three(A, B, C, D) :-C >= D, C >= B, A = C, !.
max_of_three(A, B, C, D) :-B >= D, B > C, A = B, !.
max_of_three(A, B, C, D) :-D >= B, D >= C, A = D.

goal

%max of two
max_of_two(Max, 69, 228).

%Max=228
%1 Solution

%max of three
max_of_three(Max, 1, 7, 7).

%Max=7
%1 Solution
```

Листинг 1: Задания 1 и 2

Описание работы системы

Ниже представлен алгоритм поиска ответов на вопросы `max_of_three(Max, 1, 3, 7)` и `max_of_three(Max, 3, 7, 1)` с использованием отсечения и без него.

Таблица 1: Описание работы системы без использования отсечения

№ ша- га	Состояние резольвенты, и вы- вод: дальнейшие действия (по- чему?)	Для каких термов запускается алгоритм унификации: $T1=T2$ и каков результат (и подста- новка)	Дальнейшие действия: прямой ход или откат (почему и к чему приводит?)
1	Резольвента: $\text{max_of_three}(\text{Max}, 1, 3, 7)$. Начинается поиск совпадений по БЗ		прямой ход
2	Резольвента: $\text{max_of_three}(\text{Max}, 1, 3, 7)$.	Нашли подходящее правило: $\text{max_of_three}(A, B, C, D) :- C > D, C \geq B, A = C$. Подстав- ляем $A = \text{Max}, B = 1, C = 3, D = 7$	прямой ход
3	Резольвента: $C > D$, $\text{max_of_three}(\text{Max}, 1, 3, 7)$.	Пробуем унифицировать: $3 > 7$, не подходит. Откат.	откат, продолжаем поиск
4	Резольвента: $\text{max_of_three}(\text{Max}, 1, 3, 7)$.	Нашли подходящее правило: $\text{max_of_three}(A, B, C, D) :- B \geq D, B > C, A = B$. Подстав- ляем $A = \text{Max}, B = 1, C = 3, D = 7$	прямой ход
5	Резольвента: $B \geq D$, $\text{max_of_three}(\text{Max}, 1, 3, 7)$.	Пробуем унифицировать: $1 \geq 7$, не подходит. Откат.	откат, продолжаем поиск
6	Резольвента: $\text{max_of_three}(\text{Max}, 1, 3, 7)$.	Нашли подходящее правило: $\text{max_of_three}(A, B, C, D) :- D \geq B, D \geq C, A = D$. Под- ставляем $A = \text{Max}, B = 1, C = 3, D = 7$	прямой ход
7	Резольвента: $D \geq B$, $\text{max_of_three}(\text{Max}, 1, 3, 7)$.	Пробуем унифицировать: $7 \geq 1$, подходит, идем дальше.	прямой ход
8	Резольвента: $D \geq C$, $\text{max_of_three}(\text{Max}, 1, 3, 7)$.	Пробуем унифицировать: $7 \geq 3$, подходит, идем дальше.	прямой ход
9	Резольвента: $A = D$, $\text{max_of_three}(\text{Max}, 1, 3, 7)$.	Связываем $A = D = 7$.	прямой ход
10	$\text{max_of_three}(\text{Max}, 1, 3, 7)$	Подставляем $\text{Max} = 7$. Ответ найден	прямой ход
Вывод:	$\text{Max} = 7$		

Таблица 2: Описание работы системы с использованием отсечения

№ ша- га	Состояние резольвенты, и вы- вод: дальнейшие действия (по- чему?)	Для каких термов запускается алгоритм унификации: $T1=T2$ и каков результат (и подста- новка)	Дальнейшие действия: прямой ход или откат (почему и к чему приводит?)
1	Резольвента: $\text{max_of_three}(\text{Max}, 3, 7, 1)$. Начинается поиск совпадений по БЗ		прямой ход
2	Резольвента: $\text{max_of_three}(\text{Max}, 3, 7, 1)$.	Нашли подходящее правило: $\text{max_of_three}(A, B, C, D) :- C \geq D, C \geq B, A = C, !$. Под- ставляем $A = \text{Max}, B = 1, C = 3, D = 7$	прямой ход
3	Резольвента: $C \geq D$, $\text{max_of_three}(\text{Max}, 3, 7, 1)$.	Пробуем унифицировать: $7 \geq 1$, подходит, идем дальше.	прямой ход
4	Резольвента: $C \geq B$, $\text{max_of_three}(\text{Max}, 3, 7, 1)$.	Пробуем унифицировать: $7 \geq 3$, подходит, идем дальше.	прямой ход
5	Резольвента: $A = D$, $\text{max_of_three}(\text{Max}, 3, 7, 1)$.	Связываем $A = D = 7$.	прямой ход
6	$\text{max_of_three}(\text{Max}, 3, 7, 1)$	Связываем $\text{Max} = A = 7$. От- вет найден. Поскольку отсече- ние есть, дальше не идем.	прямой ход
Вывод:	Max = 7		

Ответы на вопросы

1) Какое первое состояние резольвенты?

Изначально в резольвенте находится вопрос.

2) В каком случае система запускает алгоритм унификации? (Как эту необходимость на формальном уровне распознает система?)

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

3) Каково назначение использования алгоритма унификации?

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

4) Каков результат работы алгоритма унификации?

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

5) 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».

6) 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**

7) 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.

8) 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. Толпинская Н.Б. - Курс лекций по "Функциональному и Логическому программированию"[Текст], Москва 2020 год.
2. Анатолий Адаменко, Андрей Кучуков. Логическое программирование и Visual Prolog (с CD). — СПб.: БХВ-Петербург, 2003. — 990 с. — ISBN 5-94157-156-9.