



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«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»

Лабораторная работа № 17

Дисциплина	Функциональное и логическое программирование.
Тема	Формирование эффективных программ на Prolog.
Студент	Степанов А. О.
Группа	ИУ7-63Б
Оценка (баллы)	
Преподаватель	Толпинская Н.Б.

Москва, 2020 г.

ЗАДАНИЕ

1. Максимум из двух чисел

- (a) без использования отсечения,
- (b) с использованием отсечения;

2. Максимум из трех чисел

- (a) без использования отсечения,
- (b) с использованием отсечения

Листинг 1: Максимум 2 чисел

```
1 predicates
2     maxWithCut(integer, integer, integer).
3     maxWoutCut(integer, integer, integer).
4
5 clauses
6     % Max of two numbers with using cut
7     maxWithCut(A, B, Max) :- A > B, Max = A, !.
8     maxWithCut(_, B, Max) :- Max = B.
9
10    % Max of two numbers without using cut
11    maxWoutCut(A, B, Max) :- A > B, Max = A.
12    maxWoutCut(A, B, Max) :- A <= B, Max = B.
13
14 goal
15     write("maxWithCut_of_(3,2):\t"),
16     maxWithCut(3, 2, Max);
17
18     write("maxWithCut_of_(1,2):\t"),
19     maxWithCut(1, 2, Max);
20
21     write("maxWithCut_of_(1,1):\t"),
22     maxWithCut(1, 1, Max);
23
24     write("maxWoutCut_of_(3,2):\t"),
25     maxWoutCut(3, 2, Max);
26
27     write("maxWoutCut_of_(1,2):\t"),
28     maxWoutCut(1, 2, Max);
```

```

29
30     write("maxWoutCut_of_(1,_,1):\t"),
31     maxWoutCut(1, 1, Max).

```

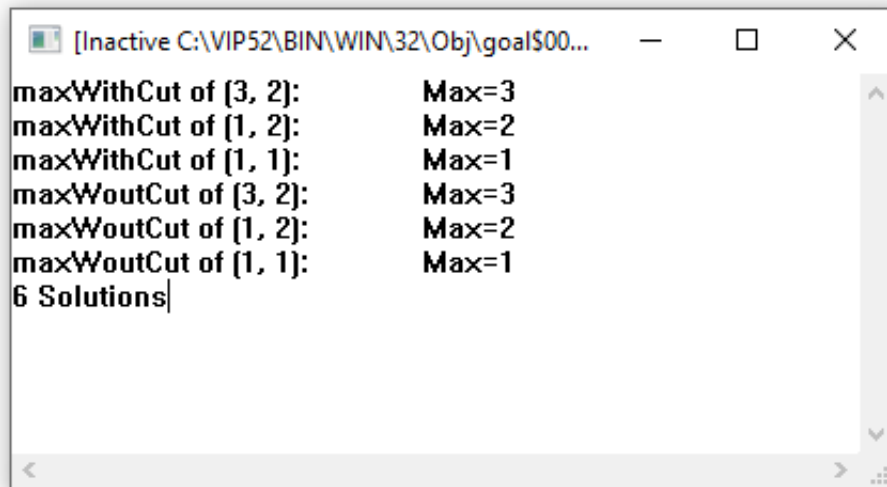
Листинг 2: Максимум 3 чисел

```

1  predicates
2      maxWithCut(integer, integer, integer, integer).
3      maxWoutCut(integer, integer, integer, integer).
4
5  clauses
6      % Max of three numbers with using cut
7      maxWithCut(A, B, C, Max) :- A > B, A > C, Max = A, !.
8      maxWithCut(_, B, C, Max) :- B > C, Max = B, !.
9      maxWithCut(_, _, C, Max) :- Max = C.
10
11     % Max of three numbers without using cut
12     maxWoutCut(A, B, C, Max) :- A >= B, A >= C, Max = A.
13     maxWoutCut(A, B, C, Max) :- B > A, B >= C, Max = B.
14     maxWoutCut(A, B, C, Max) :- C > A, C > B, Max = C.
15
16  goal
17      write("maxWithCut_of_(1,_,2,_,3):\t"),
18      maxWithCut(1, 2, 3, Max);
19      write("maxWithCut_of_(3,_,2,_,1):\t"),
20      maxWithCut(3, 2, 1, Max);
21      write("maxWithCut_of_(4,_,2,_,4):\t"),
22      maxWithCut(4, 2, 4, Max);
23      write("maxWithCut_of_(1,_,5,_,3):\t"),
24      maxWithCut(1, 5, 3, Max);
25
26      write("maxWoutCut_of_(1,_,2,_,3):\t"),
27      maxWoutCut(1, 2, 3, Max);
28      write("maxWoutCut_of_(3,_,2,_,1):\t"),
29      maxWoutCut(3, 2, 1, Max);
30      write("maxWoutCut_of_(4,_,2,_,4):\t"),
31      maxWoutCut(4, 2, 4, Max);
32      write("maxWoutCut_of_(1,_,5,_,3):\t"),
33      maxWoutCut(1, 5, 3, Max).

```

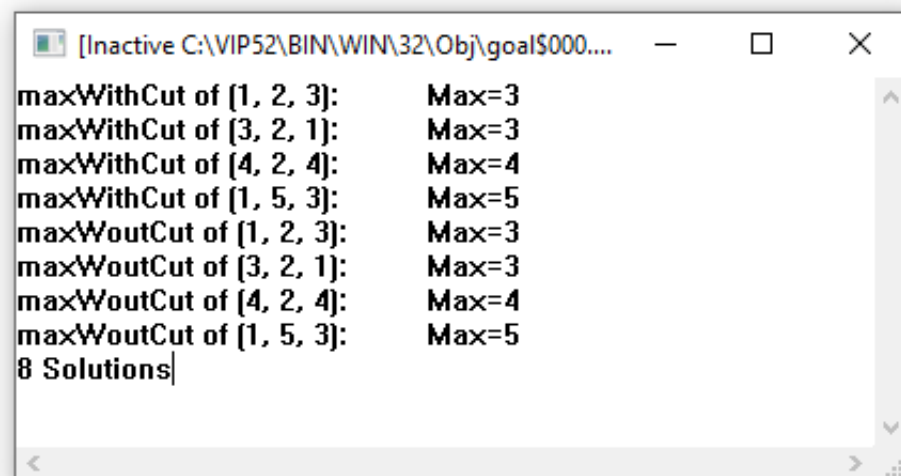
РЕЗУЛЬТАТЫ РАБОТЫ



A screenshot of a console window with the title bar "[Inactive C:\VIP52\BIN\WIN\32\Obj\goal\$000...". The window contains the following text:

```
maxWithCut of [3, 2]:      Max=3
maxWithCut of [1, 2]:      Max=2
maxWithCut of [1, 1]:      Max=1
maxWoutCut of [3, 2]:      Max=3
maxWoutCut of [1, 2]:      Max=2
maxWoutCut of [1, 1]:      Max=1
6 Solutions|
```

Рис. 1: Максимум 2 чисел



A screenshot of a console window with the title bar "[Inactive C:\VIP52\BIN\WIN\32\Obj\goal\$000...". The window contains the following text:

```
maxWithCut of [1, 2, 3]:    Max=3
maxWithCut of [3, 2, 1]:    Max=3
maxWithCut of [4, 2, 4]:    Max=4
maxWithCut of [1, 5, 3]:    Max=5
maxWoutCut of [1, 2, 3]:    Max=3
maxWoutCut of [3, 2, 1]:    Max=3
maxWoutCut of [4, 2, 4]:    Max=4
maxWoutCut of [1, 5, 3]:    Max=5
8 Solutions|
```

Рис. 2: Максимум 3 чисел

ФОРМИРОВАНИЕ ОТВЕТА

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!

Таблица 1: maxWithCut(1, 5, 3, Max)

№ 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	maxWithCut(1, 5, 3, Max)	Подстановка: A = 1, B = 5, C = 3, Max = Max maxWithCut(1, 5, 3, Max) maxWithCut(A, B, C, Max)	Прямой ход
2	A > B A > C Max = A !	Проверка: 1 > 5	Обратный ход
3	maxWithCut(1, 5, 3, Max)	Подстановка: B = 5, C = 3, Max = Max maxWithCut(1, 5, 3, Max) maxWithCut(_, B, C, Max)	Прямой ход
4	B > C Max = B !	Проверка: 5 > 3	Прямой ход
5	Max = B !	Подстановка: Max = 5	Прямой ход
6	!	Результат: Max = 5	Обратный ход

Таблица 2: maxWoutCut(3, 2, 1, Max)

№ 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	maxWoutCut(3, 2, 1, Max)	Подстановка: A = 3, B = 2, C = 1, Max = Max maxWoutCut(3, 2, 1, Max)	Прямой ход

		$\text{maxWoutCut}(A, B, C, \text{Max})$	
2	$A \geq B$ $A \geq C$ $\text{Max} = A$	Проверка: $3 \geq 2$	Прямой ход
3	$A \geq C$ $\text{Max} = A$	Проверка: $3 \geq 1$	Прямой ход
4	Пусто	Подстановка: $\text{Max} = 3$	Прямой ход
5	Пусто	Результат: $\text{Max} = 3$	Обратный ход
6	$\text{maxWoutCut}(3, 2, 1, \text{Max})$	Подстановка: $A = 3, B = 2, C = 1, \text{Max} = \text{Max}$ $\text{maxWoutCut}(3, 2, 1, \text{Max})$ $\text{maxWoutCut}(A, B, C, \text{Max})$	Прямой ход
7	$B > A$ $B \geq C$ $\text{Max} = B$	Проверка: $2 > 3$	Обратный ход
8	$\text{maxWoutCut}(3, 2, 1, \text{Max})$	Подстановка: $A = 3, B = 2, C = 1, \text{Max} = \text{Max}$ $\text{maxWoutCut}(3, 2, 1, \text{Max})$ $\text{maxWoutCut}(A, B, C, \text{Max})$	Прямой ход
9	$C > A$ $C > B$ $\text{Max} = C$	Проверка: $1 > 3$	Обратный ход

ВОПРОСЫ

1. Какое первое состояние резольвенты?

Первое состояние резольвенты – вопрос.

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

Пролог выполняет унификацию в двух случаях:

- когда цель сопоставляется с заголовком предложения;
- когда используется знак равенства, который является инфиксным предикатом (предикатом, который расположен между своими аргументами, а не перед ними).

3. Каково назначение использования алгоритма унификации?

Чтобы связать переменные из вопроса со значениями, находящимся в правилах и фактах.

4. Каков результат работы алгоритма унификации?

```
1 goal
2     P1 = birthday(person("Ivan", "Petrov"), date("August", 2, 1980)),
3     P1 = birthday(Name, date(_, _, 1980)), write(Name).
```

При согласовании первой подцели переменная P1 получит значение, указанное справа от знака “=”. При согласовании второй подцели P1 уже связана. Так как термы, находящиеся по обе стороны знака “=” сопоставимы, то переменная Name будет связана со значением person(“Ivan”, “Petrov”). При согласовании третьей подцели, стандартного предиката write, будет напечатано значение связанной переменной Name.

5. В каких пределах программы переменные уникальны?

Областью действия переменной в Prolog является одно предложение. В разных предложениях может использоваться одно имя переменной для обозначения разных объектов. Исключением является анонимная переменная. Каждая анонимная переменная – это отдельный объект.

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

Если унификация прошла успешно, то применяется подстановка. Переменные связываются со значениями.

7. Как изменяется резольвента?

Преобразование резольвенты выполняется с помощью редукции.

Редукция – замена цели телом того правила, заголовок которого унифицируется с целью. Новая резольвента получается в два этапа:

8. В каких случаях запускается механизм отката?

В том месте программы, где возможен выбор нескольких вариантов, Пролог сохраняет в специальный стек точку возврата для последующего возвращения в эту позицию. Точка возврата содержит информацию, необходимую для возобновления процедуры при откате. Выбирается один из возможных вариантов, после чего продолжается выполнение программы.

Во всех точках программы, где существуют альтернативы, в стек заносятся указатели. Если впоследствии окажется, что выбранный вариант не приводит к успеху, то осуществляется откат к последней из имеющихся в стеке точек программы, где был выбран один из альтернативных вариантов. Выбирается очередной вариант, программа продолжает свою работу. Если все варианты в точке уже были использованы, то регистрируется неудачное завершение и осуществляется переход на предыдущую точку возврата, если такая есть. При откате все связанные переменные, которые были означены после этой точки, опять освобождаются.