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»

Отчёт

по лабораторной работе № 19

Название: Обработка списков на Prolog

Дисциплина: Функциональное и логическое программирование

Студент ИУ7-65Б _____ Д.В. Сусликов

(Группа) (Подпись, дата) (И.О. Фамилия)

Преподаватель _____ Н.Б. Толпинская

(Подпись, дата) (И.О. Фамилия)

Москва, 2021

Цель работы - изучить структуру, особенности и принципы оформления программы, и способ выполнения программы на Prolog

Задание

Используя хвостовую рекурсию, разработать эффективную программу, позволяющую:

1. Найти длину списка (по верхнему уровню);
2. Найти сумму элементов числового списка
3. Найти сумму элементов числового списка, стоящих на нечетных позициях исходного списка (нумерация от 0)

Убедиться в правильности результатов.

Для одного из вариантов вопроса и одного из заданий составить таблицу, отражающую конкретный порядок работы системы.

Листинг:

```
domains
list = integer*

predicates
    len(list, integer)
    len(list, integer, integer)

    sum(list, integer)
    sum(list, integer, integer)

    sum2(list, integer)
    sum2(list, integer, integer)

clauses
    len([], R, R):- !.
    len([_|T], R, Res):- R1 = R + 1, len(T, R1, Res).
    len([H|T], Res):- len([H|T], 0, Res).
```

```
    sum([], R, R):- !.
    sum([H|T], R, Res):- R1 = R + H, sum(T, R1, Res).
    sum([H|T], Res):- sum([H|T], 0, Res).

    sum2([], R, R):- !.
    sum2([_, H|T], R, Res):- R1 = R + H, sum2(T, R1, Res).
    sum2([H|T], Res):- sum2([H|T], 0, Res).


goal
    %len([1,2,3,4], Res).
    %sum([1,2,3,5], Res).
    sum2([1,2,3,11], Res).
```

Приведем таблицу для нахождения суммы.

sum([1,2,4], Res)

Текст процедуры:

```
1:      sum([], R, R):- !.
2:      sum([H|T], R, Res):- R1 = R + H, sum(T, R1, Res).
3:      sum([H|T], Res):- sum([H|T], 0, Res).
```

№ шага	Текущая резольвента - ТР	ТЦ, выбираемые правила: сравниваемые термы, подстановка	Дальнейшие действия с комментариями
1	sum([1,2], 0, Res).	ТЦ: sum([1,2], 0, Res)	Поиск знания с начала БЗ
	sum([1,2], 0, Res).	ПР1: [] = [1,2] R = 0 R = Res Неудача	Переход к следующему заголовку БЗ
	sum([1,2], 0, Res).	ПР2: H|T] = [1,2] R = 0 Res = Res Успех H = 1 T = [2] R = 0 Res = Res	Тело ПР2 заменяет цель в резольвенте
2	R1 = 0 + 1, sum([2], R1, Res)	R1 = 1	R1 = 0 + 1 заменяется на пустое тело

3	sum([2], 1, Res)	ТЦ: sum([2],1,Res)	Поиск знания с начала БЗ
	sum([2], 1, Res)	ПР1: [] = [2] R = 1 R = Res Неудача	Переход к следующему заголовку БЗ
	sum([2], 1, Res).	ПР2: H|T] = [2] R = 1 Res = Res Успех H = 2 T = [] R = 1 Res = Res	Тело ПР2 заменяет цель в резольвенте
4	R1 = 1 + 2, sum([], R1, Res)	R1 = 3	R1 = 1 + 2 заменяется на пустое тело в резольвенте
5	sum([], 3, Res)	ТЦ: sum([],3,Res)	Поиск знания с начала БЗ
	sum([], 3, Res)	ПР1: [] = [] R = 3 R = Res Успех [] = [] R = 3 R = Res = 3	Тело ПР1 заменяет цель в резольвенте

6	!	!. Истина	Отсечение. ! заменяется на пустое в резольвенте
7	Резольвента пуста		Выводится Res = 3 Откат
8	!	! Завершение процедуры	! заменяется на пустое в резольвенте
7	Резольвента пуста		Завершение работы программы

Вывод

Эффективность работы системы может быть достигнута за счет хвостовой рекурсии и использования отсечения в тех случаях, где заведомо известна единственность ответа на вопрос.

Ответы на вопросы

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

   Рекурсия – определение объекта через ссылку на самого себя. Один из способов организации повторных вычислений. Для организации хвостовой рекурсии необходимо, чтобы рекурсивный вызов был последним в теле рекурсивного правила, и не оставалось других точек выбора. Выход из рекурсии осуществляется либо достижением базиса рекурсии, либо условием в теле правила.

2) Какое первое состояние резольвенты?

   Исходная резольвента содержит вопрос.

3) В каких пределах программы уникальны переменные?

   Именованные переменные уникальны в рамках предложения, анонимные - уникальны везде.

4) В какой момент, и каким способом системе удается получить доступ к голове списка?

   Получить доступ к голове списка можно при его унификации со списком вида [H | T], где H - голова, T - хвост.

5) Каково назначение использования алгоритма унификации?

   Алгоритм унификации необходим для того, чтобы подобрать знание, положительно отвечающее на поставленный вопрос.

6) Каков результат работы алгоритма унификации?

   Результатом работы алгоритма является значение переменной «неудача». Если неудача = 1, то унификация невозможна; если неудача = 0, то унификация прошла успешно, а побочным действием работы алгоритма является содержимое результирующей ячейки – результирующая подстановка.

7) Как формируется новое состояние резольвенты?

   Резольвента меняется в 2 этапа:

   – Редукция (замена вопроса на тело правила, заголовок которого был успешно унифицирован);

– Применение подстановки.

8) Как применяется подстановка, полученная с помощью алгоритма унификации – как глубоко?

   В результате подстановки связываются переменные, которые еще не были связаны. После связывания всех утверждений, будет напечатано значение связанных переменных.

9) В каких случаях запускается механизм отката?

   В случае, когда унификация на текущем шаге завершается тупиковой ситуацией, или был получен ответ «да».

10) Когда останавливается работа системы? Как это определяется на формальном уровне?

    Когда резольвента пуста и все указатели находятся в конце БЗ.