

Логическое программирование

Занятие 6 / 31.05.2022

Список

- **Список** – это упорядоченная совокупность произвольного числа элементов
- Порядок расположения элементов в последовательности **является существенным**
- Элементами списка могут быть **любые термы** – константы, переменные, списки, структуры, которые включают другие списки, и т.п.

Список

- `[]` – пустой список
- `[B]` – список, состоящий из одного элемента B;
- `[A, B, C]` – список, состоящий из трех элементов A, B, C;
- `[x, C, d1, "hello", (d1, m2)]` – элементами списка могут быть любые термы – переменные, константы, структуры и пр.

Список

clauses

изучает("Иванов", ["математика", "физика"]).

СПИСОК

domains

```
ilist = integer*. % список int
```

```
slist = string*. % список string
```

domains

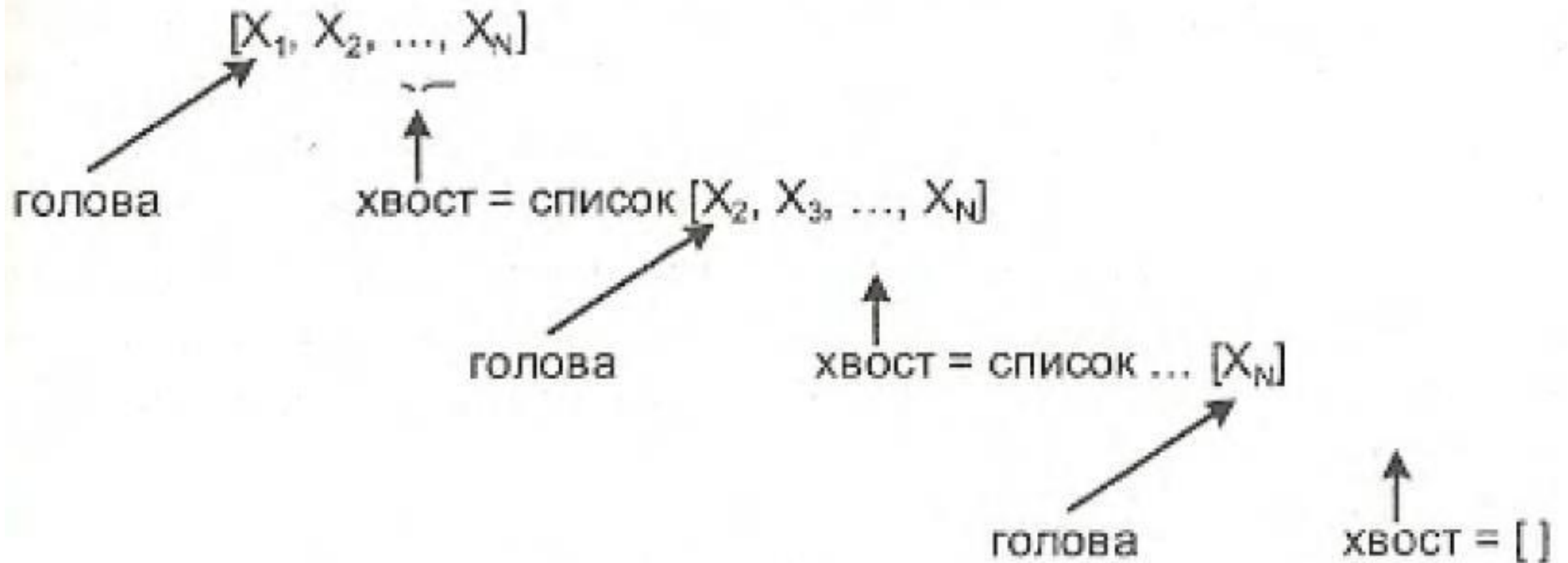
```
name = string.
```

```
group = integer.
```

```
student = student(name, group).
```

```
students = student*.
```

Голова и хвост списка



Голова и хвост списка

[a, b, c] a [b, c]

[a] a []

[] —

[a, [b, c]] a [[b, c]]

[[a, b, c], [], [c, d]] [[a, b, c]] [[], [c, d]]

[[[]], [[]]] [[]] [[[]]]

[a, [a, [a, [a, []]]]] a [[a, . . .]]

Шаблон списка

- **Шаблон списка** – это форма описания множества (семейства) списков, обладающих вполне определенными свойствами
- Шаблон $[X | Y]$ описывает любой произвольный список, состоящий не менее чем из одного элемента
- Шаблон $[X_1, X_2 | Y]$ описывает любой произвольный список, состоящий не менее чем из двух элементов
- Шаблон Z – описывает любой список, в том числе и пустой

Шаблон списка

- При унификации происходит сопоставление шаблонов. Если шаблоны целевого утверждения и утверждения базы знаний представляют списки с несовместимыми различными свойствами (разные классы списков), то унификация заканчивается неудачей
- Так, нельзя сопоставить списки:

$$\begin{array}{l} [X1, X2^2 | T] \text{ и } [a] \\ \hline [b, X | Y] \text{ и } [a, b] \\ \hline b \neq a \end{array}$$

Принадлежность элемента списку

- Предикат `contains`

Как его реализовать?

`cont(X, L)`

`cont(X, [X|_]) :- !.`

`cont(X, [_|List]) :- cont(X, List).`

Принадлежность элемента списку

domains

ilist = integer*.

class predicates

contains : (integer, ilist) determ (i,i).

clauses

contains(X, [X|_]) :- !.

contains(X, [_|T]) :- contains(X, T).

[H | T]

Простые задачи на списки

- Добавление элемента в начало списка (перед текущей головой)
- Добавление элемента в конец списка
- Определение индекса первого совпадения в списке. Если элемента в списке нет, вернуть -1. Для реализации предиката потребуется вспомогательный предикат
- Количество элементов в списке

Простые задачи на списки

- Реализовать предикаты, вычисляющие сумму, количество и среднее значение элементов целочисленных списков
- Вывод элементов списка

Предикат *присоединить*

- Предикат append

Как его реализовать?

$$\begin{array}{l} \text{app}(X, Y, Z) \\ \uparrow \\ \text{out} \end{array} \quad \begin{array}{l} (i, i, 0) \quad (0, 0, i) \\ (0, i, 0) \end{array} \quad Z = X + Y$$

$$\text{app}([], Y, Y)$$

$$\text{app}(X, Y, [1, 2, 3]) \text{ vardef}$$

$$\text{app}([H|T], Y, Z) :-$$

$$\text{app}(T, Y, Z1),$$

$$Z = [H | Z1]$$

Некорр.

Предикат *присоединить*

```
class predicates
```

```
    append : (A*, A*, A*) nondeterm (i, i, o) (o, i,  
i) (i, o, i) (o, o, i) (i, i, i).
```

```
clauses
```

```
    append([], Y, Y).
```

```
    append([X1|T1], Y, [X1|T2]) :- append(T1, Y, T2).
```

Операции со списками

- Список – рекурсивная структура => все операции со списками имеют рекурсивную структуру
- Операции можно разделить по сложности на три группы:
 - Самая простая группа – список задан и надо найти его характеристики (сумму, среднее)
 - Вторая по сложности группа – преобразование списка по заданному правилу
 - Третья – записать в список элементы, получаемые от генератора или по определенному закону

Задачи

- Даны баллы студента за все время обучения в виде списка, [4, 5, 3, 5, 4, 3, 4, ...]. Необходимо определить средний балл, количество пятерок, процент четверок среди всех баллов
- Дан список целых чисел. Необходимо исключить из него все элементы другого списка
- Дано целое число N . Необходимо сгенерировать список, содержащий все числа от 1 до N включительно
- Даны два вектора, представленные в виде списков. Необходимо найти произведение векторов