

Функциональное и логическое программирование

Лекция 6

2.12 Строки

Строка — последовательность символов, заключенная в двойные кавычки.

Предикаты для работы со строками:

1. `string_length(S, L)`

Определяет длину строки S.

Пример 1:

`string_length("abc", S).`

`S = 4`

`string_length("ab", 2).`

`true`

2. string_concat(S1,S2,S3)

Соединяет две строки S1 и S2 в третью S3. Можно использовать для разбиения строки.

Пример 2:

string_concat("ab", "123", S).

S = "ab123"

string_concat(S1, S2, "123").

S1 = "" S2 = "123";

S1 = "1" S2 = "23";

S1 = "12" S2 = "3";

S1 = "123", S2 = "";

false

3. sub_string(S,K,N,R,S1)

Выделяет в строке S подстроку S1, которая начинается с K-го элемента и содержит N символов. R – количество символов, стоящих в S после подстроки S1. Нумерация элементов строки начинается с 0.

Пример 3:

sub_string("abcdefgh", 2, 3, -, S).

S = "cde"

4. string_chars(S,L)

Преобразует строку S в список символов и наоборот.

Пример 4:

`string_chars("abc", L).`

`L = ['a', 'b', 'c']`

`string_chars(S, ['a', 'b', 'c']).`

`S = "abc"`

5. string_to_list(S,L)

Преобразует строку S в список кодов символов и наоборот.

Пример 5:

```
?- string_to_list("Aabc",L).
```

```
L = [65, 97, 98, 99].
```

```
?- string_to_list(X,[65, 97, 98, 99]).
```

```
X = "Aabc".
```

6. char_code(C,K)

Преобразует символ C в его код K и наоборот.

Пример 6:

```
?- char_code('A',C).
```

```
C = 65.
```

```
?- char_code(X,65).
```

```
X = 'A'.
```

7. split_string(S,R,D,L)

Преобразует строку S в список подстрок L, используя R как разделитель, удаляя из начала и конца подстрок символы строки D.

Пример 7:

?- split_string("asd? fgg, gh hj: j. hj hj", ' ', "?:.", L).

L = ["asd", "fgg", "gh", "hj", "j", "hj", "hj"].

8. `atomic_list_concat(L,R,S)`

Преобразует список L в строку S, используя R как разделитель и наоборот.

Пример 8:

```
?- atomic_list_concat([as,fg,jkl], ' ',S).  
S = 'as fg jkl'.
```


9. number_string(N,S)

Преобразует число N в строку S и наоборот.

Пример 9:

number_string(9,S).

S = "9"

number_string(N,"25")
N=25

10. atom_string(A,S)

Преобразует атом A в строку S и наоборот.

Пример 10:

?- atom_string(S,"Dfdg").

S = 'Dfdg'.

?- atom_string(S,"dfdg").

S = dfdg.

Пример 11:

Предикат, который преобразует строку S в строку S1, удаляя все пробелы.

```
del(S,S1):-string_chars(S,L),delete(L,' ',L1),string_chars(S1,L1).
```

```
?- del("asd fgg ghj  ghj",S).
```

```
S = "asdfggghjghj".
```

Пример 12:

Предикат, который считает количество вхождений символа С в строку S.

```
chet([],_,0).  
chet([C | T],C,N):-chet(T,C,N1),N is N1+1.  
chet(_ | T],C,N):-chet(T,C,N).
```

?- goal.

Vvedite stroku

|: "asdaaaa aaga".

Vvedite symvol

|: a.

8

true .

2.13 Предикаты для работы с файлами

1. `exists_file(<'имя файла'>)`

Завершается успешно, если файл с указанным именем существует (обратные слэши дублируются).

2. `open(<'имя файла'>, <режим>, F)`

Открытие файла для чтения, записи или добавления.

Режимы:

- `read` (для чтения);
- `write` (для записи);
- `append` (для добавления).

F – файловая переменная

Предикаты для работы с файлами являются внелогическими.
Чтение и обработку данных следует выполнять отдельно!

3. `set_input(F)`
 `set_output(F)`

Перенаправление ввода из файла или вывода в файл.

4. `close(F)`
Заккрытие файла.

5. `see(<'имя файла'>)`

`tell(<'имя файла'>)`

Открытие и перенаправления ввода из файла или вывода в файл вместо 2 и 3.

При перенаправлении ввода на клавиатуру, а вывода на экран в качестве имени файла используют имя `user`.

6. `seen`

`told`

Заккрытие файлов, открытых с помощью `see` и `tell`.

7. seeing(F)

telling(F)

Связывает F с именем файла, являющегося текущим входным или выходным потоком.

8. at_end_of_stream

Успешно завершается, если найден конец файла.

9. read_line_to_codes(F,L)

Читает строку из входного потока F и преобразует ее в список кодов символов этой строки (без кода перевода строки).

10. read_line_to_string(F,S)

Читает строку из входного потока F (вводить строку без кавычек и точки).

11. read_stream_to_codes(F,L)

Читает содержимое из входного потока F (до конца файла) и преобразует его в список кодов символов (включая коды перевода строки 10).

Пример 1: Вывести содержимое текстового файла на экран.

```
goal1:-see('6.txt'),seeing(F),  
      read_stream_to_codes(F,L),string_to_list(S,L),write(S),seen.
```

~~—Предикат не проработал—~~

На самом деле все работает, в аудитории криво поставлен SWI Prolog.

Содержимое файла 6.txt:

Life, believe, is not a dream
So dark as sages say;
Oft a little morning rain
Foretells a pleasant day.
Sometimes there are clouds of gloom,
But these are transient all;
If the shower will make the roses bloom,
Oh, why lament its fall?
Rapidly, merrily,
Life's sunny hours flit by,
Gratefully, cheerily,
Enjoy them as they fly!

Результат работы:

Life, believe, is not a dream
So dark as sages say;
Oft a little morning rain
Foretells a pleasant day.
Sometimes there are clouds of gloom,
But these are transient all;
If the shower will make the roses bloom,
Oh, why lament its fall?
Rapidly, merrily,
Life's sunny hours flit by,
Gratefully, cheerily,
Enjoy them as they fly!

Пример 2:

Написать предикат, который выводит на экран строки из файла, начиная с некоторого номера.

Имя файла и номер строки вводятся с клавиатуры.

```
goal2:-writeln('file name'),read(Fn),
    check(Fn),
    writeln('number'),read(N),
    open(Fn,read,F),set_input(F),
    read_file(F,N),
    format('Sodergimoe s ~W stroki',N),
    write_screen(F),close(F).
check(Fn):-exists_file(Fn),!.
check(_):-write('file not'),fail.
read_file(_,N):-at_end_of_stream,!,
    format('v file < ~W strok',N),fail.
read_file(_,1):-!.
read_file(F,N):-read_line_to_codes(F,_),
    N1 is N-1,
    read_file(F,N1).
write_screen(_):-at_end_of_stream,!.
write_screen(F):-read_line_to_codes(F,L),
    string_to_list(S,L),writeln(S),
    write_screen(F).
```

Программа не проработала

Необходимо исправить выделенные буквы на строчные.

```
?- goal2.
file name
|: '6.txt'.
number
|: 5.
Sodergimoe s 5 stroki
Sometimes there are clouds of gloom,
But these are transient all;
If the shower will make the roses bloom,
Oh, why lament its fall?
Rapidly, merrily,
Life's sunny hours flit by,
Gratefully, cheerily,
Enjoy them as they fly!
```

Пример 3:

Написать предикат, который записывает вводимые с клавиатуры строки в файл out.txt. Окончание ввода – строка “#”.

```
write_file:-read(X),tell('out.txt'),write_s(X),
            told.
write_s("#"):-!.
write_s(X):-writeln(X),read(Y),write_s(Y).
```

```
?- write_file.
|: "asddf".
|: "dfgdhjhd".
|: "#".
```

Содержимое файла out.txt:

```
asddf
dfgdhjhd
```