Московский авиационный институт

(государственный технический университет)

**Факультет прикладной математики и физики**

Кафедра вычислительной математики и программирования

**Лабораторная работа №4**

по курсу

«Логическое программирование»

*«Обработка естественного языка»*

|  |  |
| --- | --- |
| Выполнила: | Карпова В.А. |
| Группа: | 08-306 |
| № по списку: | 9 |
| Руководитель: | Левинская М.А. |
| Оценка: | |
| Дата: | |

Москва, 2012 г.

**Задание**

Реализовать грамматический разбор фраз на ограниченном естественном языке и преобразовать данные фразы в язык исчисления предикатов первого типа:

**Вариант 10**

Реализовать разбор фраз языка, представляющих собой положительные и отрицательные высказывания. В результате предикат должен выдавать все атомарные глубинные структуры.

Запрос:

?- decompose([“Саша” “любит” “игрушки” “,” “но” “не” “любит” “кубики” “и” “мячи”],X);

?- decompose([“Ира” “не” “любит” “стихи” “и” “прозы” “,” “а” “любит” “пьесы”],X)

Результат: X=likes(’Саша’, ’игрушки’).

X=not\_likes(’Саша’, ’кубики’).

X=not\_likes(’Саша’, ’мячи’).

X=not\_likes(’Ира’, ’стихи’).

X=not\_likes(’Ира’, ’прозы’).

X=likes(’Ира’, ’пьесы’).

**Грамматика**

Терминальные символы: слова английского языка, action (действия, выполняемые одним существом), action2 (действия, для двух объектов), noun(существительные), quantifiers (кванторы).

*Нетерминальные символы*: PHRASE (PH), ACTION, OBJECT

*Правила*:

PHRASE -> agent Ph (1)

PH -> PH i PH (2) | PH , no PH (3) | PH , a PH (4) |ACTION OBJECT (5) ACTION -> ne action (6)| action (7)

OBJECT -> OBJECT i OBJECT (8)| object (9)

*Начальный терминал*:

PH

**Метод решения**

% вспомогательные предикаты

remove(X, [X|T], T):- !.

remove(X, [A|T], [A|Y]):- remove(X, T, Y).

in\_string([], R, R):- !.

in\_string(L, R, Res):-

append([H1], T1, L),

atom\_concat(R, H1, H),

remove(H1, L, L1),

in\_string(L1, H, Res).

string(A, B):- A =.. B.

replace(X, Y, [X|T], [Y|T]):- !.

replace(X, Z, [A|T], [A|Y]):- replace(X, Z, T, Y).

% словари

agent\_list(['Sasha', 'Ira']).

object\_list([igrushki, kubiki, myachi, stixi, prosu, pieci, kukol]).

action\_list([lubit]).

printElem(\_, \_, []):-!.

printElem(Ac, Ag, Obj):-

append([O], T, Obj),

string(Res, [Ac, Ag, O]),

print(Res), print('\n'),

printElem(Ac, Ag, T).

Print([], \_, \_):-!.

Print(Ac, Ag, Obj):-

append([A1], NewAc, Ac),

append([O1], NewObj, Obj),

printElem(A1, Ag, O1),

Print(NewAc, Ag, NewObj).

% (1)

a\_phrase(P):-

append([A], Ph, P),

agent\_list(L),

sublist([A], L),

a\_ph(Ph, Res2, Res3),

Print(Res2, A, Res3), !.

% 2, 3, 4

a\_ph(Ph, [R1, R3], [R2, R4]):-

append(P1, T, Ph),

append(['i'], P2, T),

a\_ph(P1, R1, R2),

a\_ph(P2, R3, R4).

% 5

a\_ph(Ph, Res1, Res2):-

append(A, O, Ph),

a\_action(A, Res1),

a\_object(O, Res2).

% 6

a\_action(A, Res):-

append(['ne'], [A1], A),

action\_list(L),

sublist([A1], L),

in\_string(['ne\_', A1], '', Res).

% 7

a\_action([A], A):-

action\_list(L),

sublist([A], L).

% 8

a\_object(O, Res):-

append(O1, T, O),

append(['i'], O2, T),

a\_object(O1, Res1),

a\_object(O2, Res2),

append(Res1, Res2, Res).

% 9

a\_object(O, O):-

object\_list(L),

sublist(O, L).

% запятые можно просто убирать

del(L, R):- remove(',', L, R1), del(R1, R).

del(L, L).

f(A, B, X, Y):- replace(A, B, X, Y), !.

f(A, B, X, X).

decompose(Expr):-

del(Expr, Expr1),

f('a', 'i', Expr1, Expr2), % чтобы не писать лишних правил

f('no', 'i', Expr2, Expr3), % преобразуем "а","но" к "и"

a\_phrase(Expr3),!.

test1(['Sasha', lubit, igrushki, ',', no, ne, lubit, kubiki, i, myachi]).

test2(['Ira', ne, lubit, stixi, i, prosu, ',', a, lubit, pieci]).

test3(['Ira', ne, lubit, kukol, i, igrushki, i, lubit, kubiki, i, myachi]).

*Протокол*:

| ?- test1(X), decompose(X).

lubit(Sasha,igrushki)

ne\_lubit(Sasha,kubiki)

ne\_lubit(Sasha,myachi)

X = ['Sasha',lubit,igrushki,',',no,ne,lubit,kubiki,i,myachi]

 yes

| ?- test2(X), decompose(X).

ne\_lubit(Ira,stixi)

ne\_lubit(Ira,prosu)

lubit(Ira,pieci)

X = ['Ira',ne,lubit,stixi,i,prosu,',',a,lubit,pieci](47 ms) yes

| ?- test3(X), decompose(X).

ne\_lubit(Ira,kukol)

ne\_lubit(Ira,igrushki)

lubit(Ira,kubiki)

lubit(Ira,myachi)

X = ['Ira',ne,lubit,kukol,i,igrushki,i,lubit,kubiki,i,myachi]

 Yes

**Выводы**

В данной лабораторной работе мы рассмотрели простейший способ разбора языка. По фразе, данной нам, необходимо было построить грамматику. Однако, на практике используются более сложные методы анализа и алгоритмов, т.к. обработка естественного языка – занятие не простое. Полноценная обработка естественного языка применяется во многих практических задачах, так как обработка ошибок, подбор фраз по смыслу, восстановление данных по обрезкам фраз.

Существуют полноценные лексические анализаторы, такие как lex, jlex и т.д, которые устроены более сложным образом.

Основная цель в нашей лабораторной состояла в том, что нам необходимо было не только разобрать грамматику, но и составить предикаты по ней. Т.е. мы разобрали частный случай лексического анализатора.