Отчет

по лабораторной работе №4

Студент группы 8О-206Б Гончар Д.А., №4 по списку

Работа выполнена 24.11.2016г.

Преподаватель: Левинская Мария Александровна

Итоговая оценка: \_\_\_\_

Подпись преподавателя: \_\_\_\_

В качестве системы программирования был выбран SWI Prolog 4.26.

Цель работы

Познакомиться на практике с методами анализа естественно-языковых текстов системах логического программирования, реализовать несложный фрагмент естественного-языкового интерфейса к модельной задаче и протестировать его на ряде примеров.

Условие задачи

Реализовать синтаксический анализ арифметических выражений для перевода его в префиксную форму. В выражении допустимы знаки +,-,\*,/.

Листинг программы

symbol(1,"+").

symbol(1,"-").

symbol(2,"\*").

symbol(2,"/").

term(1,"+",Exp1,Exp2,plus1(Exp1,Exp2)).

term(1,"-",Exp1,Exp2,minus(Exp1,Exp2)).

term(2,"\*",Exp1,Exp2,multiply(Exp1,Exp2)).

term(2,"/",Exp1,Exp2,divide(Exp1,Exp2)).

subexpression(N,[S|List], Other,Exp1,Exp) :- symbol(N,S), N1 is N+1,

expression(N1,List,List1,Exp2),!,

term(N,S,Exp1,Exp2,Term),

subexpression(N,List1, Other,Term,Exp).

subexpression(\_,List,List,Exp,Exp).

expression(N,List,Other,Exp) :- N<3, N1 is N+1,

expression(N1,List,List1,Exp1),!,

subexpression(N,List1, Other,Exp1,Exp).

expression(3,[Int| Other], Other,is\_int(I)) :- I=Int.

append1([H|L1],L2,[H|L]) :- append1(L1,L2,L).

append1([],L,L).

convert(plus1(Exp1,Exp2),["+"|List]) :- convert(Exp1,L1),

convert(Exp2,L2),

append1(L1,L2,List).

convert(minus(Exp1,Exp2),["-"|List]) :- convert(Exp1,L1),

convert(Exp2,L2),

append1(L1,L2,List).

convert(multiply(Exp1,Exp2),["\*"|List]) :- convert(Exp1,L1),

convert(Exp2,L2),

append1(L1,L2,List).

convert(divide(Exp1,Exp2),["/"|List]) :- convert(Exp1,L1),

convert(Exp2,L2),

append1(L1,L2,List).

convert(is\_int(I),[Int]) :- I=Int.

calculate(List,Res) :- expression(1,List,\_,Term),

convert(Term,Res).

Разбор программы

Пролог-программа осуществляет разбор заданного выражения и преобразует его в префиксную форму при помощи предиката calculate(List,Res). Предикаты expression и subexpression преобразуют заданное арифметические выражение в следующий вид:

«1\*2+3/4 -> plus multiply(is\_int(1),is\_int(2)) divide(is\_int(3),is\_int(4))». После преобразования полученный набор букв конвертируется в читабельную префиксную форму вида «+ \* 1 2 / 3 4».

Ответ

?- calculate([5,"+",2,"\*",3],Res).

Res = ["+", 5, "\*", 2, 3] ;

false.

?- calculate([1,"\*",а,"-",2,"/",b],Res).

Res = ["-", "\*", 1, а, "/", 2, b] ;

false.

Вывод

В арифметическое выражение может входить одна или более компонент, таких, как числа, арифметические операторы, переменная, конкретизированная арифметическим выражением, и т.д. В результате выполнения лабораторной работы было выявлено, что арифметические выражения могут быть проанализированы на языке программирования Пролог и переведены в префиксную форму. Т.е. был написан синтаксический анализатор, допускающий знаки (+,-,\*,/), переменные(a,b,c,…) и константы.