Toy Logic Programming Language

Overview:

a. Token:

|UPPER_STRING {to identify variable}
|LOWER_STRING {to identify constant}
|FUNAME {to identify function name}
|COMMA |ENDLINE |ASSIGN |CP |EOF |OP

b. Parser (Context Free grammar):

Need to build two parser one for program reading and one for query reading

c. Internal Representation:

1.The program was store as array of
symbol(list of clause)*
Array data structure was used for search.
2.Now *symbol* is of type
*(string*int)* this represent the function name and arity
3.The *clause* is of type
| Rule (a,b) { where a is of type *af* and b is of type *af list}*
| Fact a { where a is of type *af}*
4. The *af* is of type *symbol*(term list)*
5. the *term* is of type
| Const c { where *c* is of type *string}*
| Var x { where *x* is of type *variable* and *variable* is *string}*
| Funct af

d. Search:

As search was to be done by step by step, a mutable stack was used to remember previous search environment and then continue in next step

e. Note:

1. The program should be in test.pl file
2. In case of recursive rule, the search terminates if stack length increases by some threshold
3. The internal variables are represented by strings _{original name}_{sub query #}_{# of iterations}
4. Type command ./compile.sh to run shell
5. Type command ./refresh.sh to delete all files except source code