Assigment 3
Logic & Knowledge Representation
DMKM

Carlos López Roa
me@mr3m.me

January 11, 2016

1 Express the syntax of the different commands

```
<command> ::= <instruction>

<instruction> ::= cal <args1>
<instruction> ::= cat <args2>
<instruction> ::= cp <file> <target> | cp <opt3> <file> <target> |
cp <opt3> <file> <file2> <target> | cp <file> <file2> <target>
<instruction> ::= grep <expr> | grep -<opt4> <expr> | grep -<opt4e> <expr> |
grep -<opt4> -<opt4e> <expr> | grep <expr> <file2> |
grep -<opt4> <expr> <file2> | grep -<opt4e> <expr> <file2> |
grep -<opt4> -<opt4e> <expr> <file2>

<args1> ::= <month> | <year> | <month> <year> | " "
<month>::= 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12
<year>  ::= 1 | 2 | ... | 9998 | 9999

<args2>::= -<opt2> <file> | -<opt2> | <file> | " "
<opt2>  ::= n | b | s | u | v | e | t

<opt3>  ::= <opt31> | <opt32>* | <opt31> <opt32>*
<opt31> ::= -r | -R
<opt32> ::= -f | -i | -p
<file2> ::= <file>*

<opt4>  ::= b | c | i | h | l | n | b | s | y
<opt4e> ::= e
```

2 Write a PROLOG program readcommand(C) that reads a line on the current input stream and that returns the list of ascii codes it contains.

```
\% Reads a character recursively from the prompt and
stops when encounters the new line character.

read_command([C | L]) :-
get0(C),
C \== 10 ,
!,
read_command(L).
read_command([10]).
```

```
?- read_command(C),print(C).
|    Hello World
[72,101,108,108,111,32,87,111,114,108,100,10]
```

3 Write a PROLOG program that parses the command line obtained in the previous question and that returns the command under the form of a PROLOG terms defined as follows:

- cal

```
?- parse(C).
|    cal
C = calendar(1, 2016) .

?- parse(C).
|    cal 1990
C = calendar(1990) .

?- parse(C).
|    cal 1 1990
C = calendar(1, 1990) .
```

- cat

```
?- parse(C).
|    cat
C = concatenate(option_list, file_list) .

?- parse(C).
|    cat -nbsuvet
C = concatenate("-nbsuvet", file_list) .

?- parse(C).
|    cat -asdfasd
false.

?- parse(C).
|    cat file1 file2 file3
C = concatenate(option_list, ["file1", "file2", "file3"]) .

?- parse(C).
|    cat -nbs file1 file2 file3
C = concatenate("-nbs", ["file1", "file2", "file3"])
```

- cp

```
?- parse(C).
|    cp
false.

?- parse(C).
|    cp file
false.

?- parse(C).
```

```
|    cp file target
C = copy(option_list, "file", "target") .

?- parse(C).
|    cp file1 file2 target
C = copy(option_list, ["file1", "file2"], "target") .

?- parse(C).
|    cp -r -f -i -p file1 target
C = copy(["-r", "-f", "-i", "-p"], "file1", "target") .
```

- grep

```
?- parse(C).
|    grep
false.

?- parse(C).
|    grep expr
C = search_expr(option_list_1, option_2, "expr", list_files) .

?- parse(C).
|    grep expr target
C = search_expr(option_list_1, option_2, "expr", "target") .

?- parse(C).
|    grep -bcihlnvsy expr target
C = search_expr([-, b, c, i, h, l, n, v|...], option_2, "expr", "target") .

?- parse(C).
|    grep -bci -e expr target
C = search_expr([-, b, c, i], -e, "expr", "target") .
```