CSE341 – Programming Languages

(Fall 2020)

MIDTERM REPORT

MİRAY YILDIZ

161044023

In midterm project, firstly I read input from “input.txt” file. I put the data I read into the string first. Then I converted this string into a nested list with the help of "read-from-string" function. The name of the function I read the file is "file-get-contents" and the name of the function that I convert the string to a list is "convert". I call file-get-contents function into convert function.

The function that makes the control is the "find-solution" function. This function has one parameter. This parameter contains the name of the input file. In the find-solution function, I first call the "convert" function and create my list.

After creating my list, I created 3 more new lists. The names of these lists are fact, predicate and query.

I started checking my nested list. I went through the elements of the list in order. For example, our nested list given in the question was as follows:

```
(
    ( (“legs” (“X” 2)) ( (“mammal” (“X”)) (“arms” (“X” 2)) ) )
    ( (“legs” (“X” 4)) ( (“mammal” (“X”)) (“arms” (“X” 0)) ) )
    ( (“mammal” (“horse”)) () )
    ( (“arms” (“horse” 0)) () )
    ( () (“legs” (“horse” 4)) )
)
```

I checked the elements one by one. For example, the first item in the list above is:

```
( (“legs” (“X” 2)) ( (“mammal” (“X”)) (“arms” (“X” 2)) ) )
```

This list has 2 elements and none of them are empty. That's why I put this list on my predicate list.

While checking in turn, I put elements in the other two lists. For example :

```
( (“mammal” (“horse”)) () )
```

When I got to the list above, the second item of this list was empty. That's why I put this list on the fact list.

I added the lists with the first empty element to the query list. The following element can be given as an example:

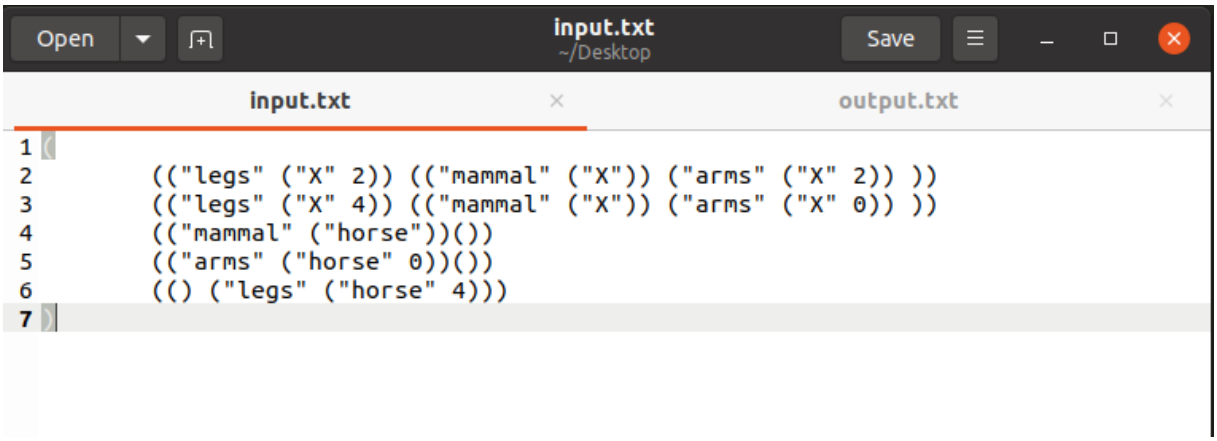
(() (“legs” (“horse” 4)))

At the end of these operations, I searched the query in the predicate list. The important point is to check the parameters as well. Because, for example, in the example above, there are 2 predicates. The first elements of both predicate contain "legs". But when we look at the next nested list, the information here is different.

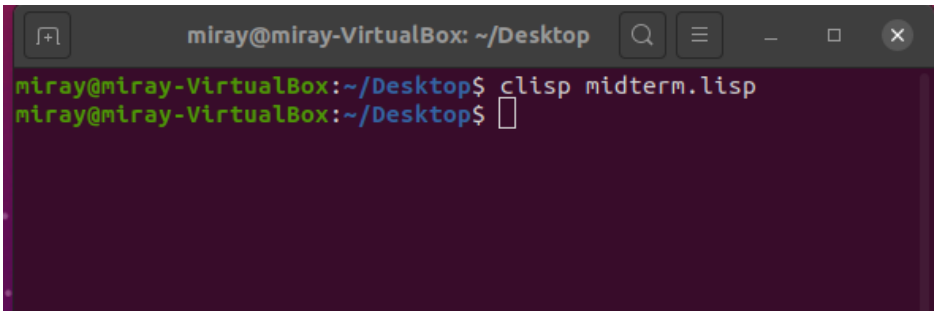
After checking them one by one, I put the information in the appropriate predicate in a separate list. Later, I checked this information in the fact list. If I have reached the correct information as a result of all my checks, the query is correct, if I could not, the query is incorrect. If the query is correct, I print "(T)" on the file, if it is wrong I print "()". I did this in the "write-file" function.

TESTS :

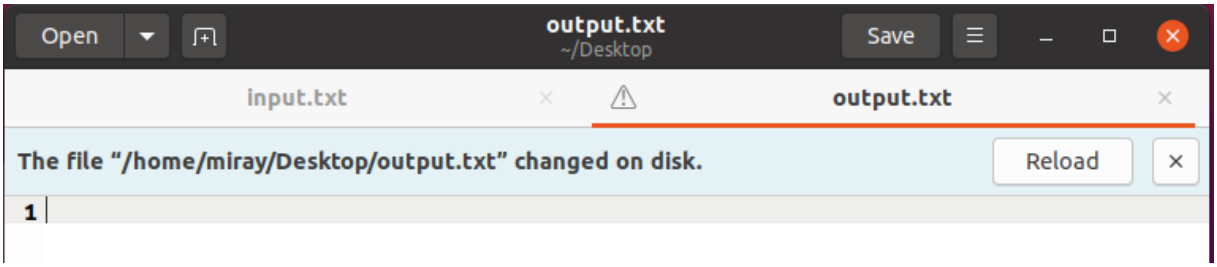
input.txt :



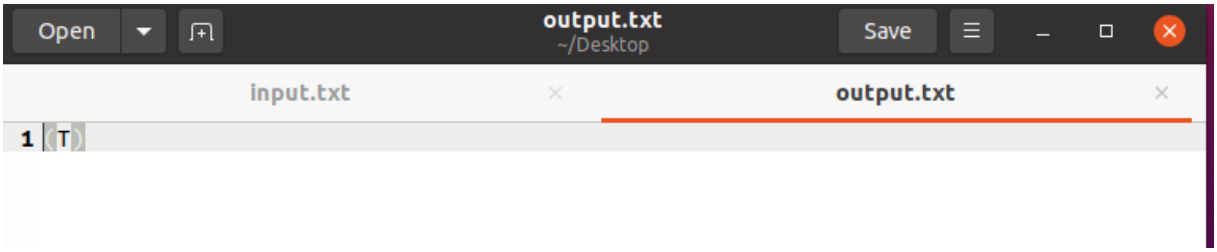
Compile :



output.txt :



Click reload:



It is true because When we follow the list, we see this predicate :

(("legs" ("X" 4)) (("mammal" ("X")) ("arms" ("X" 0))))

Then, in the fact list, we see

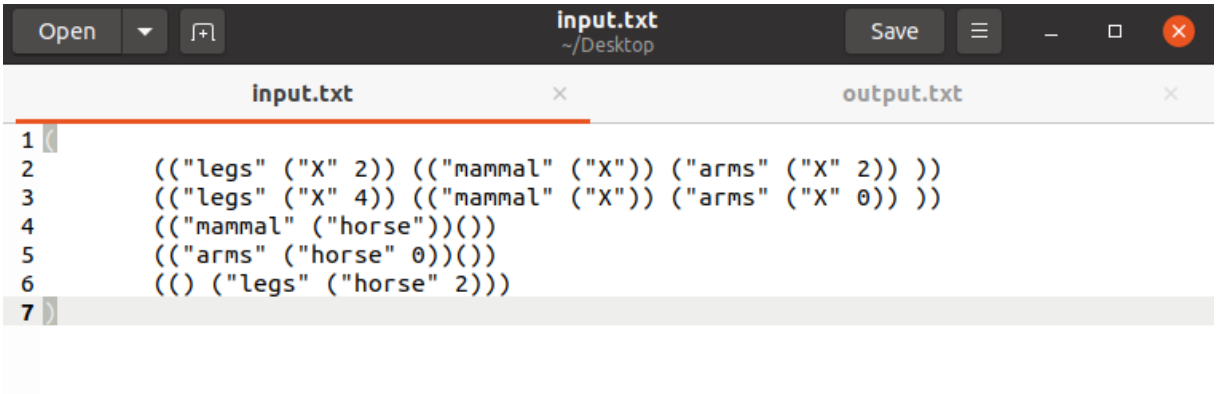
(("mammal" ("horse")) ())

(("arms" ("horse" 0)) ())

So, this query is true.

But, when we do this :

input.txt :



In this input, difference is in query.

(() ("legs" ("horse" 2)))

Compile :

```
miray@miray-VirtualBox: ~/Desktop
miray@miray-VirtualBox:~/Desktop$ clisp midterm.lisp
miray@miray-VirtualBox:~/Desktop$
```

output.txt :

```
Open   output.txt ~/Desktop   Save
input.txt   output.txt
The file "/home/miray/Desktop/output.txt" changed on disk.   Reload
1
```

Reload:

```
Open   output.txt ~/Desktop   Save
input.txt   output.txt
1 ()
```

As you can see above, the result is an empty list because horses are mammals but they don't have 2 legs. This information is also included in the list.

Another example,

İnput.txt :

```
Open   input.txt ~/Desktop   Save
input.txt   output.txt
1 (
2         (("legs" ("X" 2)) (("bird" ("X")) ("arms" ("X" 0)) ))
3         (("legs" ("X" 4)) (("mammal" ("X")) ("arms" ("X" 0)) ))
4         (("mammal" ("horse"))())
5         (("bird" ("chicken")))
6         (("arms" ("horse" 0))())
7         (("arms" ("chicken" 0)()))
8         (() ("legs" ("chicken" 2)))
9 )
```

Compile :

```
miray@miray-VirtualBox: ~/Desktop
miray@miray-VirtualBox:~/Desktop$ clisp midterm.lisp
miray@miray-VirtualBox:~/Desktop$
```

output.txt :

```
Open   output.txt ~/Desktop   Save
input.txt   output.txt
The file "/home/miray/Desktop/output.txt" changed on disk.   Reload
1
```

Reload:

```
Open   output.txt ~/Desktop   Save
input.txt   output.txt
1 (T)
```

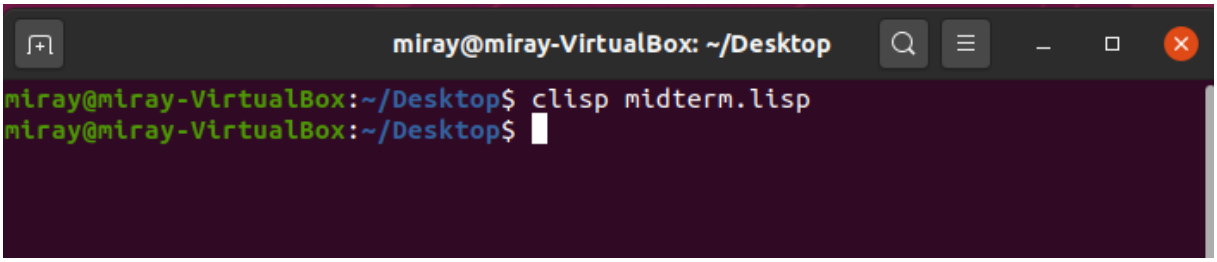
It is true because chicken is a bird and it has 2 leg.

Last example :

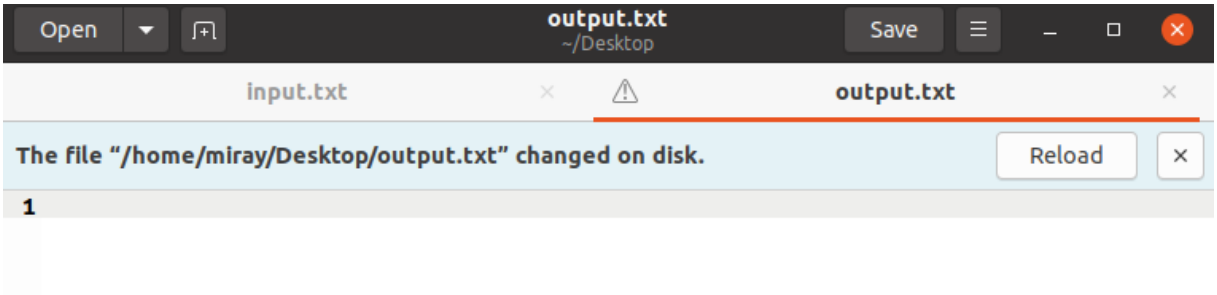
input.txt :

```
Open   input.txt ~/Desktop   Save
input.txt   output.txt
1 (
2         (("legs" ("X" 2)) (("bird" ("X")) ("arms" ("X" 0)) ))
3         (("legs" ("X" 4)) (("mammal" ("X")) ("arms" ("X" 0)) ))
4         (("mammal" ("horse"))())
5         (("bird" ("chicken")))
6         (("arms" ("horse" 0))())
7         (("arms" ("chicken" 0)()))
8         (() ("legs" ("chicken" 4)))
9 )
```

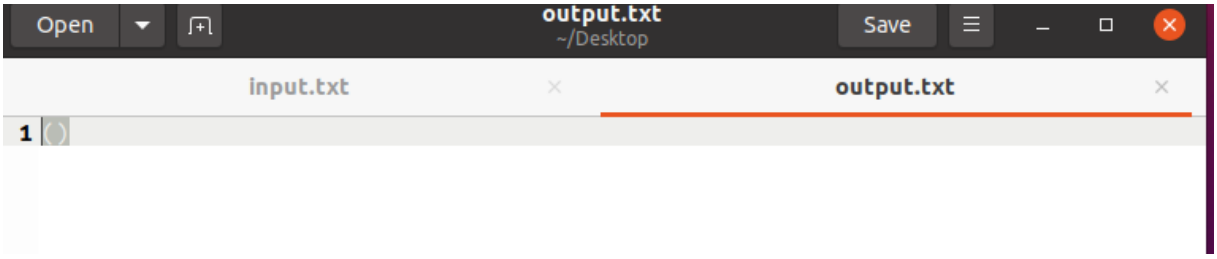
Compile :



output.txt :



Reload:



İt is false because chicken has not 4 legs.

NOTE : Only problem is in my code, It only works when one query is given.