|  |
| --- |
| Erasmus Mundus Master in Data Mining and Knowledge Management |
| Yet Another Datalog Interpreter (YADI) |
| Work Package 1 |
|  |
| **Gaurav Singh (UPMC), Sandra Mitrovic (UPMC), Arpaporn Skunkittiyut (EPUN) and Shalini Gangwar (EPUN)** |
| **18-Nov-12** |

|  |
| --- |
|  |

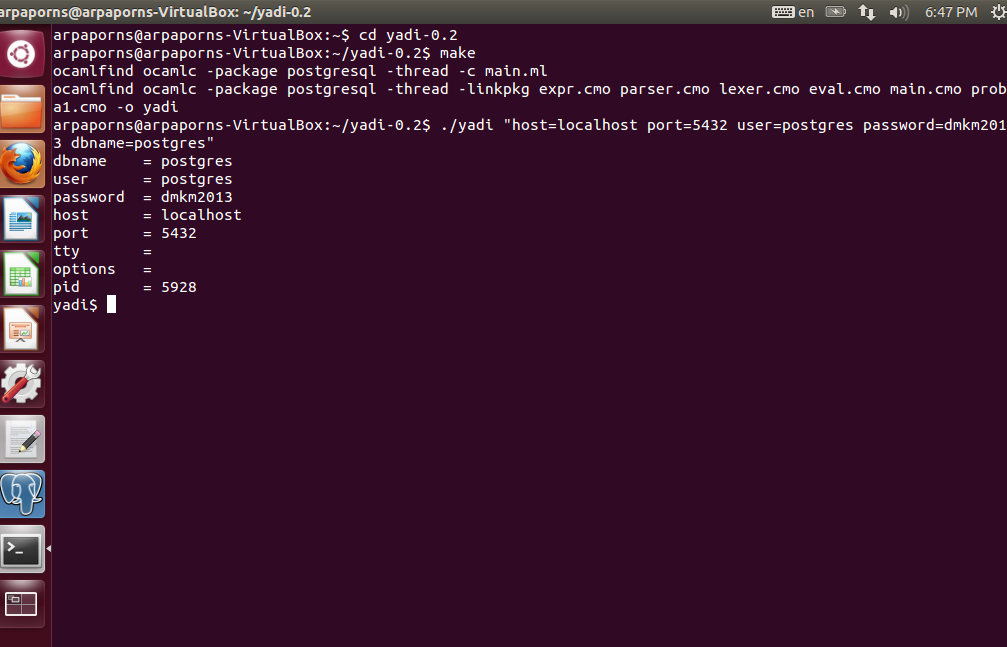
# User Manual

* 1. Start at terminal
  2. Go to the destination path of program by using command “cd yadi-0.2”
  3. Execute Make command

|  |
| --- |
| Make |

* 1. Connect to Database by using command

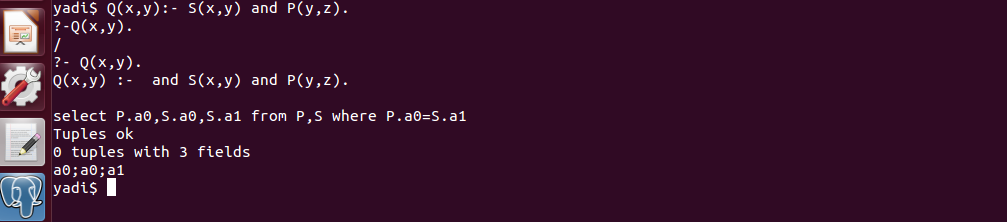
|  |
| --- |
| “./yadi "host=localhost port=5432 user=postgres password=dmkm2013 dbname=postgres" |



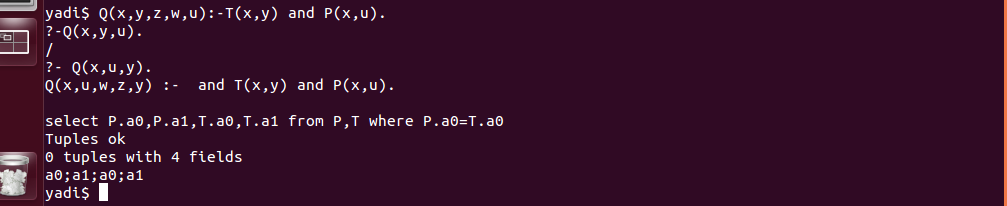
* 1. Query example

|  |
| --- |
| Q(x,y):- S(x,y) and P(y,z). ?-Q(x,y). / |

The result should be as following:



|  |
| --- |
| Q(x,y,z,w,u):- T(x,y) and P(x,u). ?-Q(x,y,u). / |



To illustrate, we insert some data into the table as following:

**Table P**

|  |  |
| --- | --- |
| A0 | A1 |
| ID001 | Arpaporn |
| ID002 | Shalini |
| ID003 | Sandra |
| ID004 | Gaurav |

**Table Q**

|  |  |
| --- | --- |
| A0 | A1 |
| ID001 | Thailand |
| ID002 | India |
| ID003 | Montrenegro |
| ID004 | India |

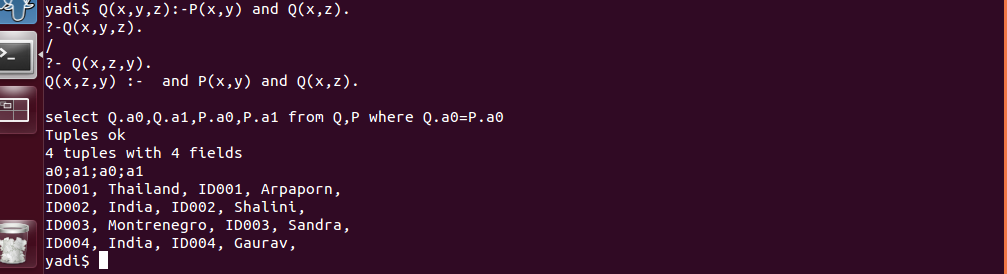
**Table R**

|  |  |
| --- | --- |
| A0 | A1 |
| Thailand | Bangkok |
| Spain | Madrid |
| Montrenegro | Ppodgorica |
| India | Delhi |

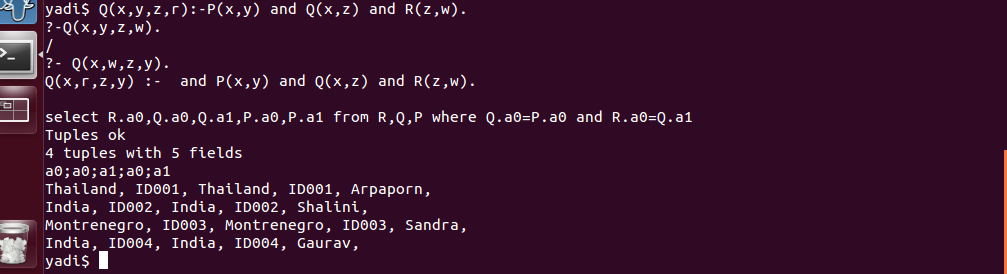
**Table S**

|  |  |
| --- | --- |
| A0 | A1 |
| ID001 | Nantes |
| ID002 | Nantes |
| ID003 | Paris |
| ID004 | Paris |

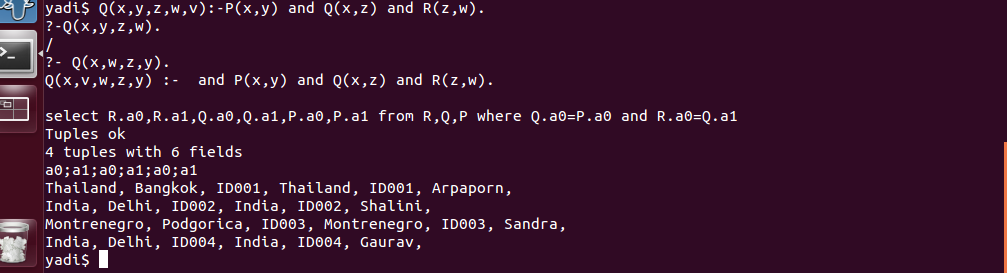
|  |
| --- |
| Q(x,y,z):- P(x,y) and Q(x,z). ?-Q(x,y,z). / |



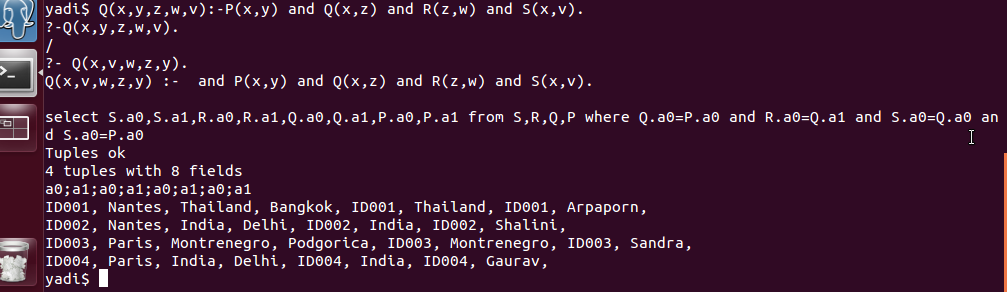
|  |
| --- |
| Q(x,y,z,r):- P(x,y) and Q(x,z) and R(z,w). ?-Q(x,y,z,w). / |



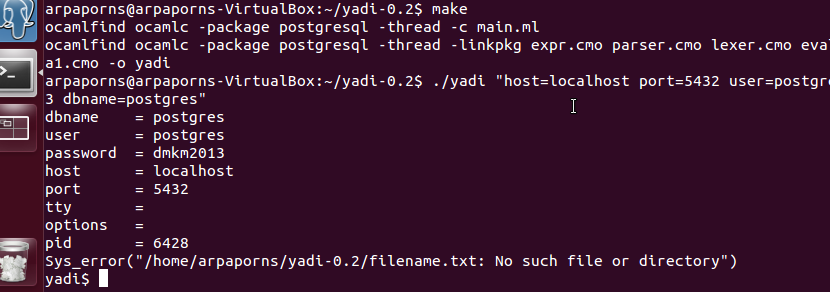
|  |
| --- |
| Q(x,y,z,w,v):- P(x,y) and Q(x,z) and R(z,w). ?-Q(x,y,z,w). / |



|  |
| --- |
| Q(x,y,z,w,v):- P(x,y) and Q(x,z) and R(z,w) and s(x,v). ?-Q(x,y,z,w,v). / |

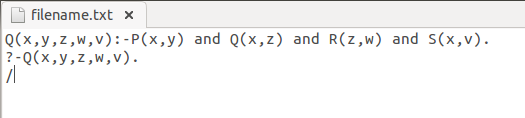


In addition, we have created the function to read the query from file and if there is no file found it will raise the error that “No such file or directory”

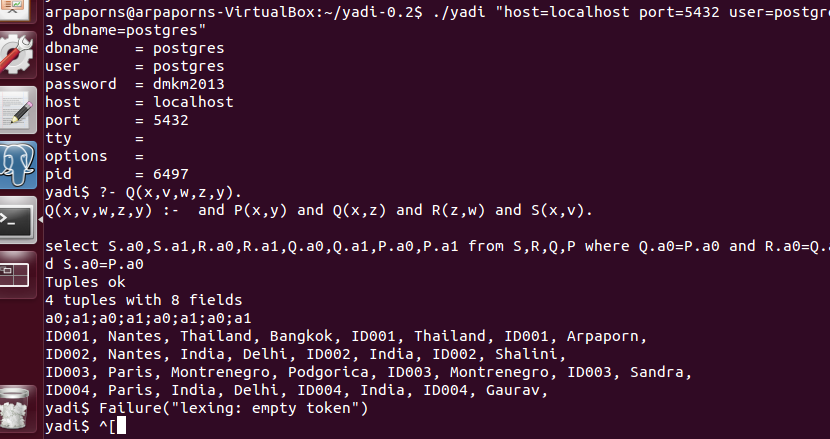


In the file, we can put the sample query in order to run automatically after get the connection to database.

For example:



If the file is placed in the correct directory, the result after user connects to the database will be shown as following:



# The current limitations

With current solution:

- predicate with variables ending by dot will be recognized as fact as well

- no views are created so idb predicates are not stored

- it is supposed that database schema already exists; columns in tables should be named as a0, a1, a2 etc.

- no negation or equalities are supported

# Known issues

- Since we modify the main function from "let lexbuf = Lexing.from\_channel stdin in" to "let lexbuf = Lexing.from\_string (print\_all\_lines filecontents) in" in order to get input from file instead of user key. And when we put the query in the file, program can execute the whole query stated in file, however at the end of the file it will display as parser error.

- We get the error from "PostgreSQL Ocaml" library on modification or changing connection string to the database through user interface

- In creating SELECT statement from datalog rule, in cases when variable in head appears in more than one predicate in the body we have extra columns in select part, e.g.

for:

yadi$ Q(x,y):-R(x,z) and S(z,x,y).

?-Q(x,y).

/

the current output is:

select S.a1,S.a2,R.a0 from S,R where S.a0=R.a1 and S.a1=R.a0

# TODO list for the second Work Package

* Change facts so that they can be recognized only as predicates with constants ending by dot
* Developing help file for Yadi
* Enhanced error handling (e.g. in case of wrong syntax, the compiler should response to user where the error occured and also stating the reason of particular error)
* Implementing boolean predicate
* Simplify the command (e.g. for database connection, no need of writing whole connection string but just typing one command (e.g. Dbconnect))
* Overcome all issues and limitation that are identified so far