

A QUICK TUTORIAL ON HOW TO WRITE DATALOG PROGRAMS IN XSB.

1- Connect to *timberlake* server, and open your editor and write a datalog program.

Here, I use *vi* editor and I write the example in the datalog lecture. Don't forget these

- a) The extension of the file should be *.P*
- b) Put *':-auto_table.'* at the beginning of your programs.



The image shows a PuTTY terminal window with a blue title bar. The title bar text is "timberlake.cse.buffalo.edu - PuTTY". The terminal area has a black background with white text. The prompt "timberlake (~) >" is followed by the command "vi example.P" and a green cursor. On the right side of the terminal window, there is a vertical scrollbar and a status bar at the bottom.

```
timberlake.cse.buffalo.edu - PuTTY
timberlake (~) > vi example.P
```

```
timberlake.cse.buffalo.edu - PuTTY
:- auto_table.

%facts
friend(joe,sue).
friend(ann,sue).
friend(sue,max).
friend(max,ann).

%%rules
fof(X,Y) :- friend(X,Y).
fof(X,Z) :- friend(X,Y), fof(Y,Z).

%%Query 1
query1(X) :- fof(X,ann).

%%Query2
query2(X) :- fof(X,Y), fof(Y,X).

~
~
~
~
"example.P" 18L, 240C                                18,1      All
```

2- To go into XSB , you have to type xsb and you'll see ?-

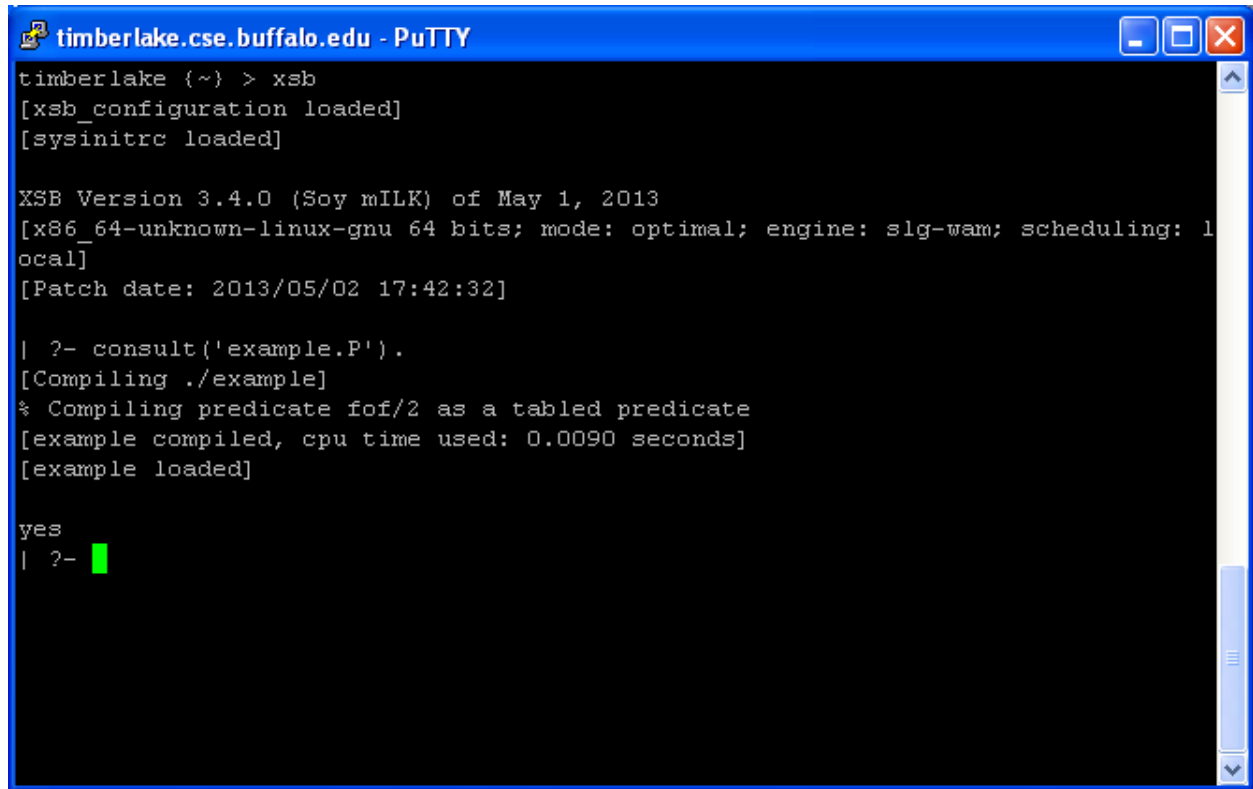
```
timberlake.cse.buffalo.edu - PuTTY
timberlake (~) > vi example.P
timberlake (~) > xsb
[xsb_configuration loaded]
[sysinitrc loaded]

XSB Version 3.4.0 (Soy mILK) of May 1, 2013
[x86_64-unknown-linux-gnu 64 bits; mode: optimal; engine: slg-wam; scheduling: local]
[Patch date: 2013/05/02 17:42:32]

| ?-
```

3- To load your file, we use *consult* command i.e. *consult('example.P')*, if your file is located in the same directory that you run xsb, there's no need to specify the path.

If the file loaded successfully you will see a 'yes.'



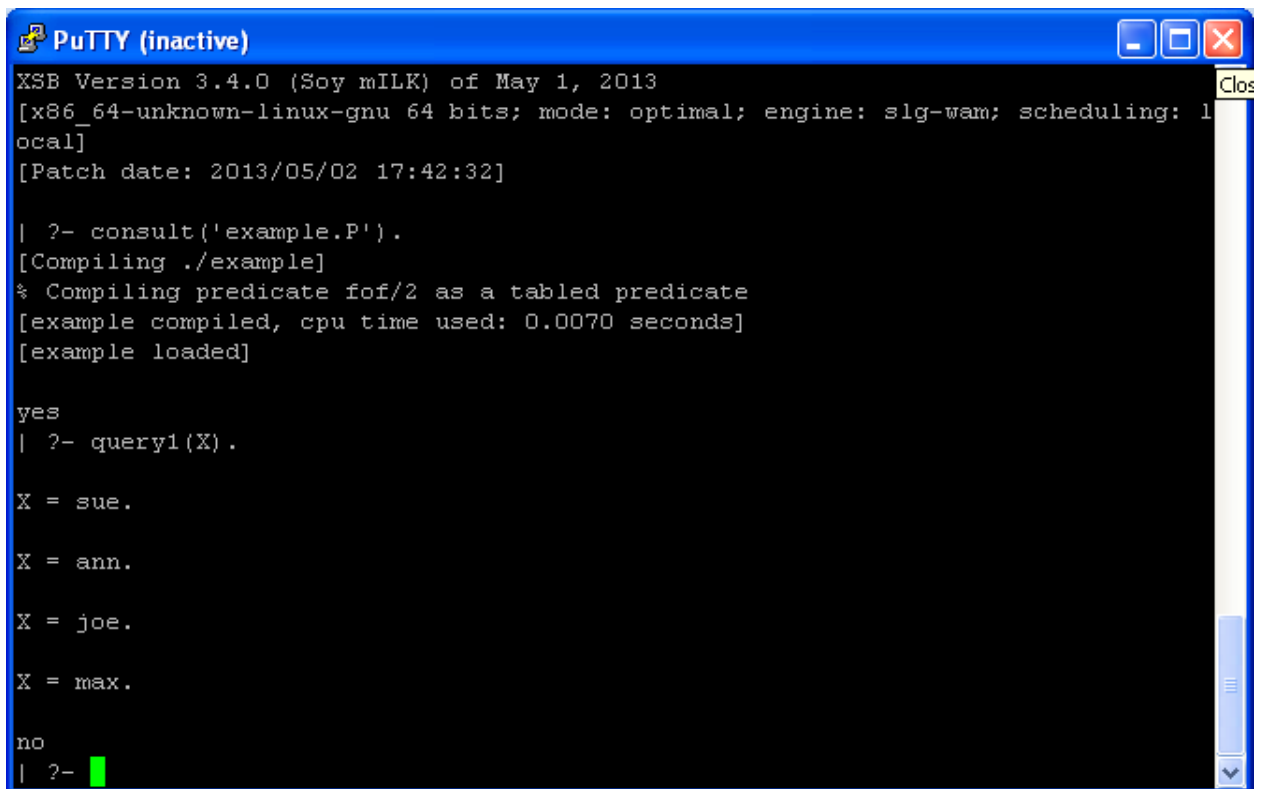
```
timberlake (~) > xsb
[xsb_configuration loaded]
[sysinitrc loaded]

XSB Version 3.4.0 (Soy mILK) of May 1, 2013
[x86_64-unknown-linux-gnu 64 bits; mode: optimal; engine: slg-wam; scheduling: local]
[Patch date: 2013/05/02 17:42:32]

| ?- consult('example.P').
[Compiling ./example]
% Compiling predicate fof/2 as a tabled predicate
[example compiled, cpu time used: 0.0090 seconds]
[example loaded]

yes
| ?- █
```

4- In the figures below you will see different queries I runned. If a query has more than one answer, XSB show one answer at the time, in order to see other answers, put *a dot and press enter* (You can also use setoff, and bagof)

A screenshot of a PuTTY terminal window titled "PuTTY (inactive)". The window has a blue title bar with standard window controls (minimize, maximize, close) on the right. The terminal content shows the XSB Prolog environment. It starts with the version string "XSB Version 3.4.0 (Soy mILK) of May 1, 2013" and system information "[x86_64-unknown-linux-gnu 64 bits; mode: optimal; engine: slg-wam; scheduling: local]". A patch date "[Patch date: 2013/05/02 17:42:32]" is also displayed. The user enters the Prolog query "consult('example.P').", which triggers compilation messages: "[Compiling ./example]", "% Compiling predicate fof/2 as a tabled predicate", "[example compiled, cpu time used: 0.0070 seconds]", and "[example loaded]". The user then enters "yes" and the query "query1(X).". The program returns four solutions: "X = sue.", "X = ann.", "X = joe.", and "X = max.". After the last solution, it returns "no". The prompt "| ?- " is followed by a green cursor. On the right side of the terminal, there is a vertical scrollbar and a "Close" button at the top right.

```
XSB Version 3.4.0 (Soy mILK) of May 1, 2013
[x86_64-unknown-linux-gnu 64 bits; mode: optimal; engine: slg-wam; scheduling: local]
[Patch date: 2013/05/02 17:42:32]

| ?- consult('example.P').
[Compiling ./example]
% Compiling predicate fof/2 as a tabled predicate
[example compiled, cpu time used: 0.0070 seconds]
[example loaded]

yes
| ?- query1(X).

X = sue.

X = ann.

X = joe.

X = max.

no
| ?- 
```

```
timberlake.cse.buffalo.edu - PuTTY
no
| ?- query2(X) .

X = max.
X = max.
X = max.
X = sue.
X = sue.
X = sue.
X = ann.
X = ann.
X = ann.

no
| ?- 
```

```
timberlake.cse.buffalo.edu - PuTTY
X = sue.
X = sue.
X = ann.
X = ann.
X = ann.

no
| ?- fof(ann,sue) .

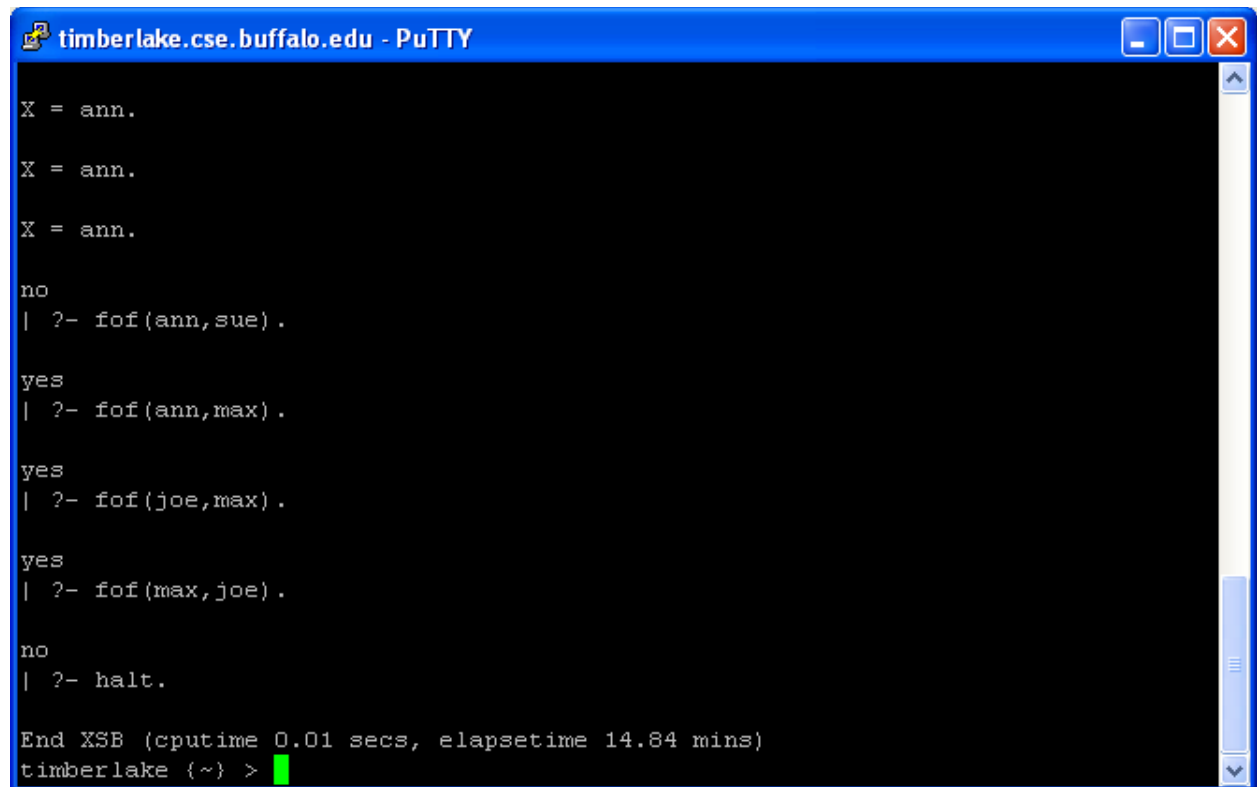
yes
| ?- fof(ann,max) .

yes
| ?- fof(joe,max) .

yes
| ?- fof(max,joe) .

no
| ?- 
```

In order to exit from XSB, use *halt* predicate, as shown below:



```
timberlake.cse.buffalo.edu - PuTTY

X = ann.
X = ann.
X = ann.
no
| ?- fof(ann,sue).
yes
| ?- fof(ann,max).
yes
| ?- fof(joe,max).
yes
| ?- fof(max,joe).
no
| ?- halt.

End XSB (cputime 0.01 secs, elapsetime 14.84 mins)
timberlake (~) >
```