

Introduction to ASP and Clingo

Víctor Gutiérrez-Basulto

List of Tasks

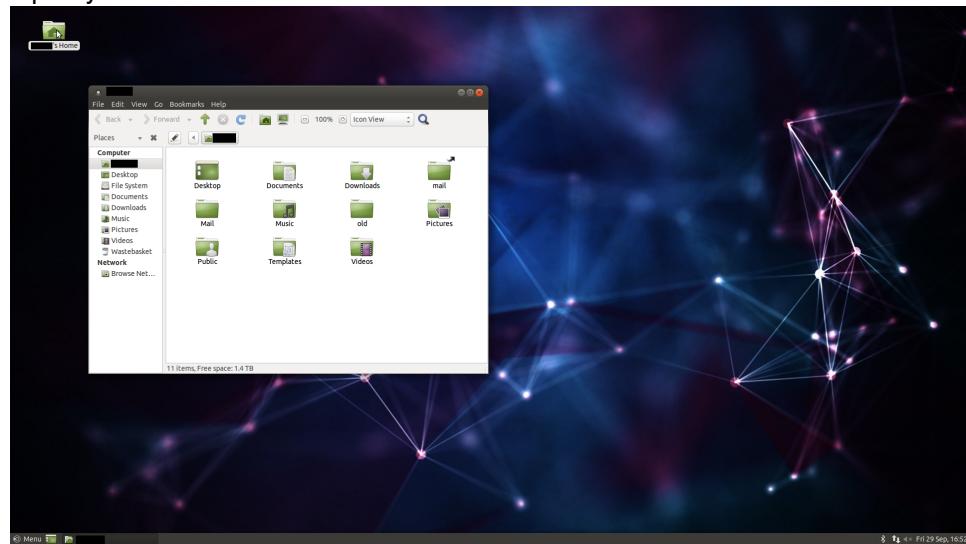
1	Setup your environment	1
2	Expand the program and the database	6

Installation (on personal computer)

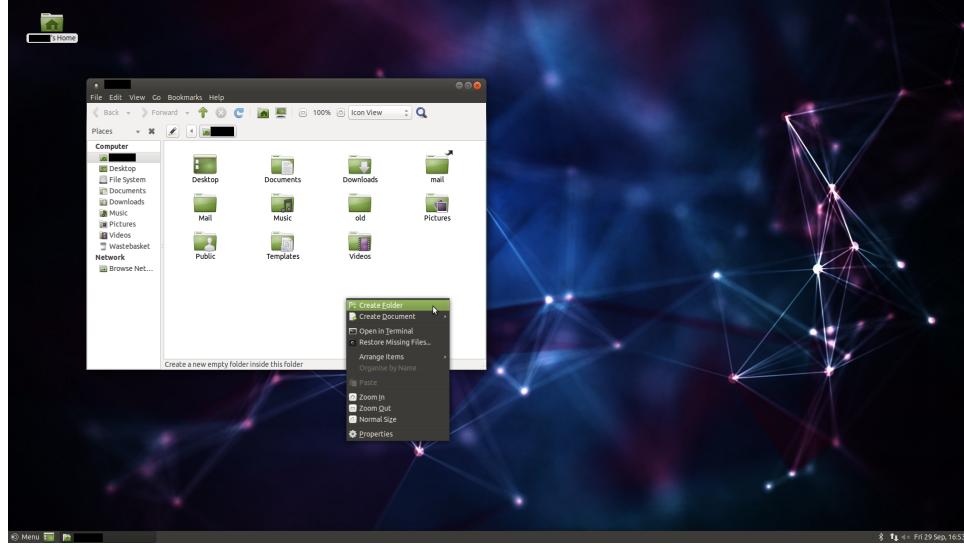
Download and install the correct (for your OS) binary version of clingo 5.2.1 from <https://github.com/potassco/clingo/releases>
Download and install the correct (for your OS) binary version of sublime text editor from <https://www.sublimetext.com/3>

1 Setup your environment

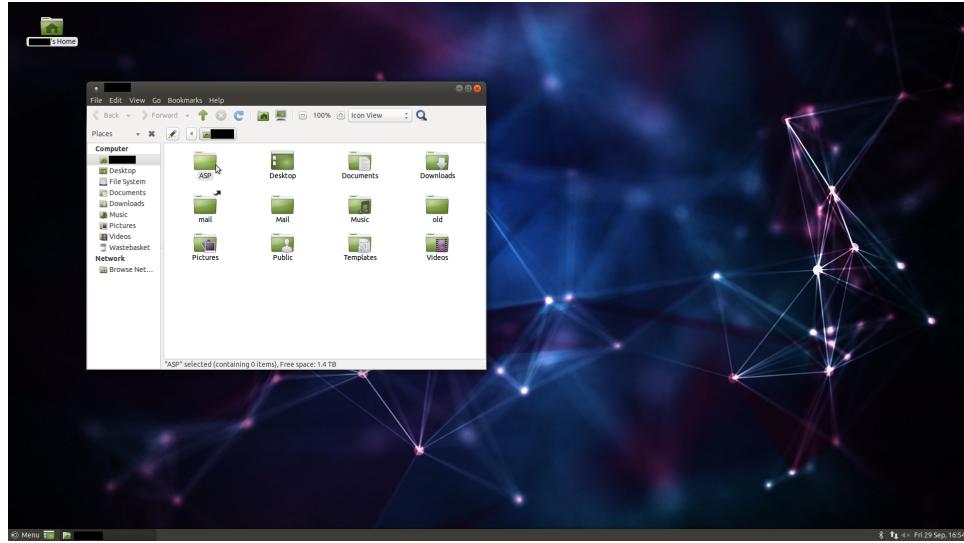
Login with your university credentials.
Open your home folder.



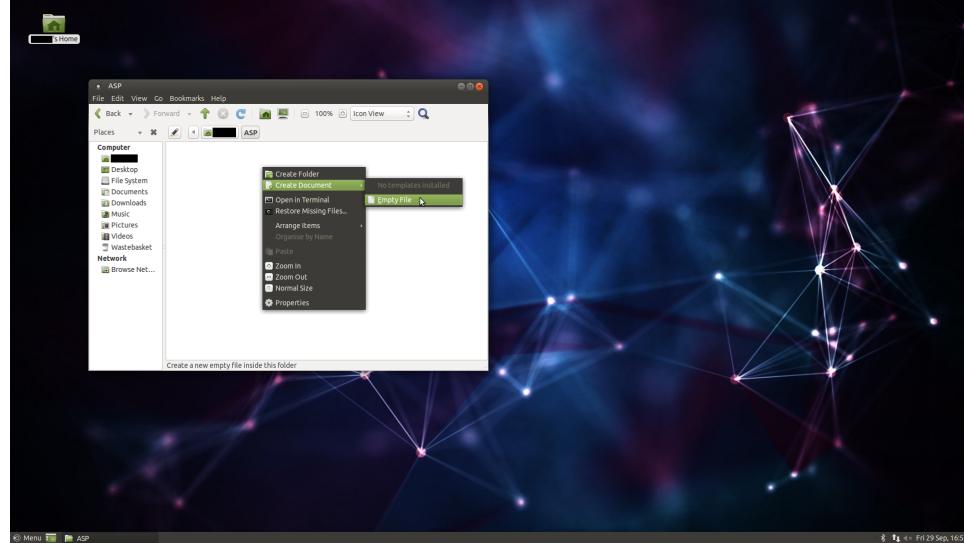
Create a new folder with a right-click and then selecting the option Create Folder.



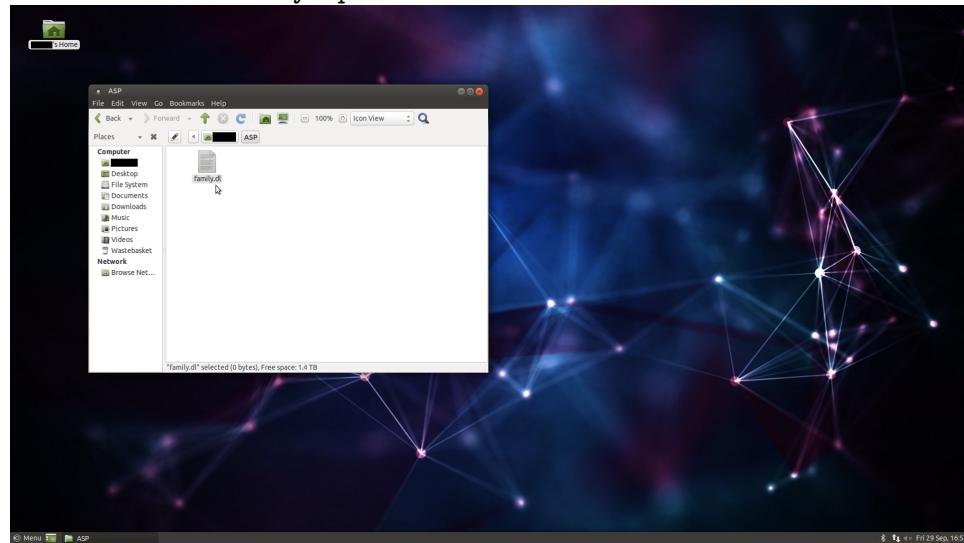
Name the folder as ASP.



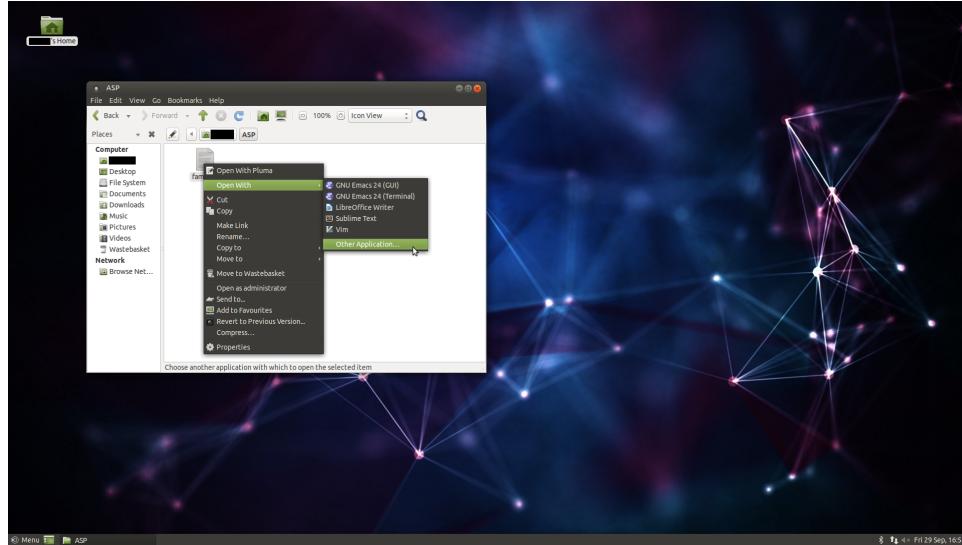
Enter the folder ASP and create a file with a right-click, Create Document, Empty File.



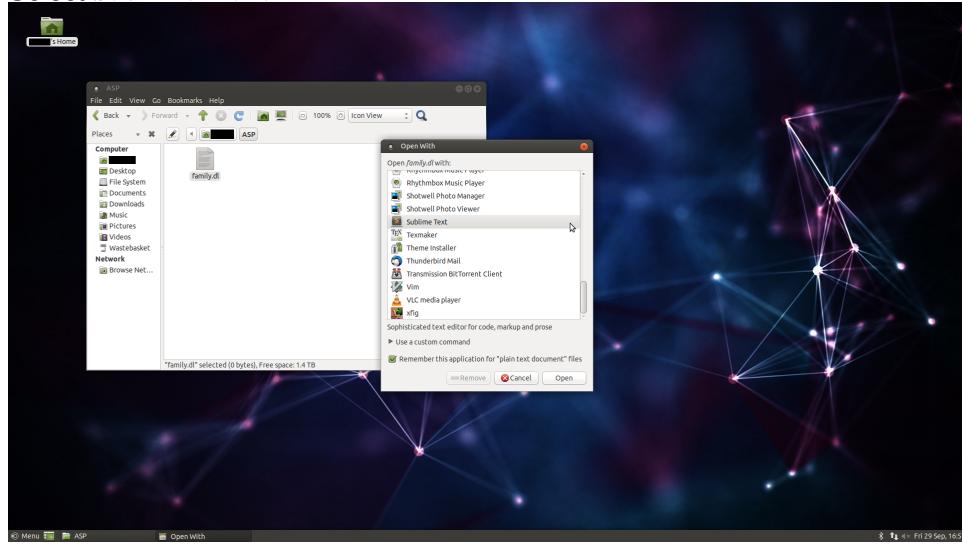
Name the file as family.lp.



Open the file with Sublime text editor: right-click on the file, Open With, Other Application.



Select Sublime Text.



Write in the Sublime text editor the following lines:

```
ancestor(X,Y) :- parent(X,Y).
```

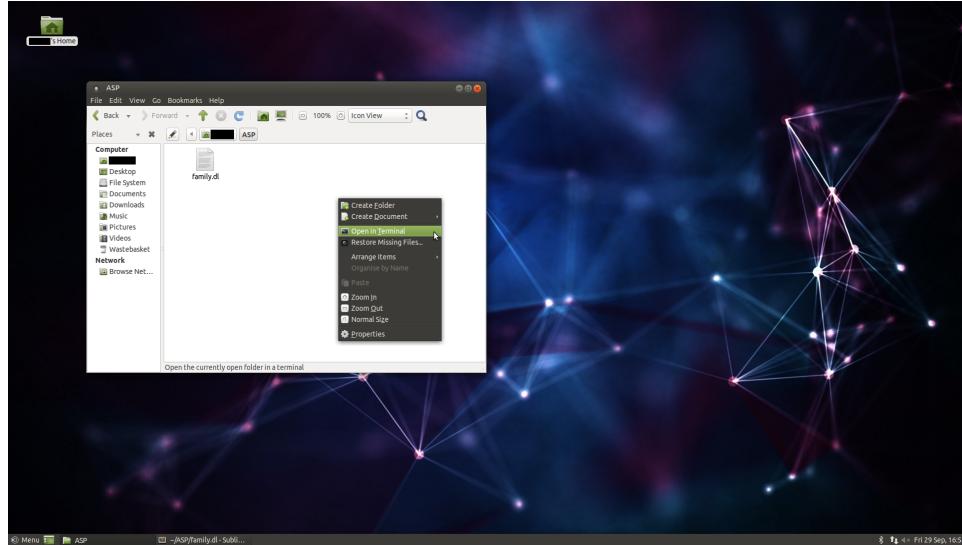
```
parent(john , bill ).
```

Please note that every information added to the file starts with a lower-case letter. Only variables are capitalised.

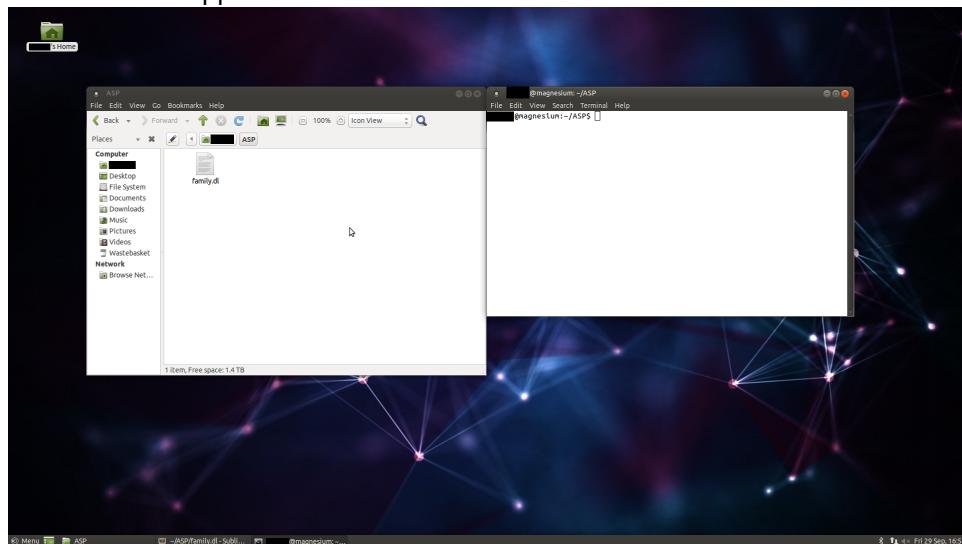
Save the file.

Open a terminal and navigate to the directory ASP. The easiest way is to right-

click when within the ASP folder, and select Open in Terminal.



A terminal will appear.



Execute the command: `clingo family.lp`, where `clingo` is the ASP solver we will use in this course, and `family.lp` is the file just created.

```

username@magnesium:~ASP/$ clingo family.dl
clingo version 5.2.1
Reading from family.dl
Solving...
Answer: 1
parent(john,bill) ancestor(john,bill)
SATISFIABLE

Models      : 1
Calls       : 1
Time        : 0.040s (Solving: 0.00s 1st Model: 0.00s Unsat: 0.00s)
CPU Time    : 0.000s
username@magnesium:~ASP/$

```

Explain such a behaviour

2 Expand the program and the database

The line:

```
ancestor(X,Y) :- parent(X,Y).
```

expresses the rule: if X is a parent of Y, then X is an ancestor of Y.

Moreover, the line

```
parent(john , bill ).
```

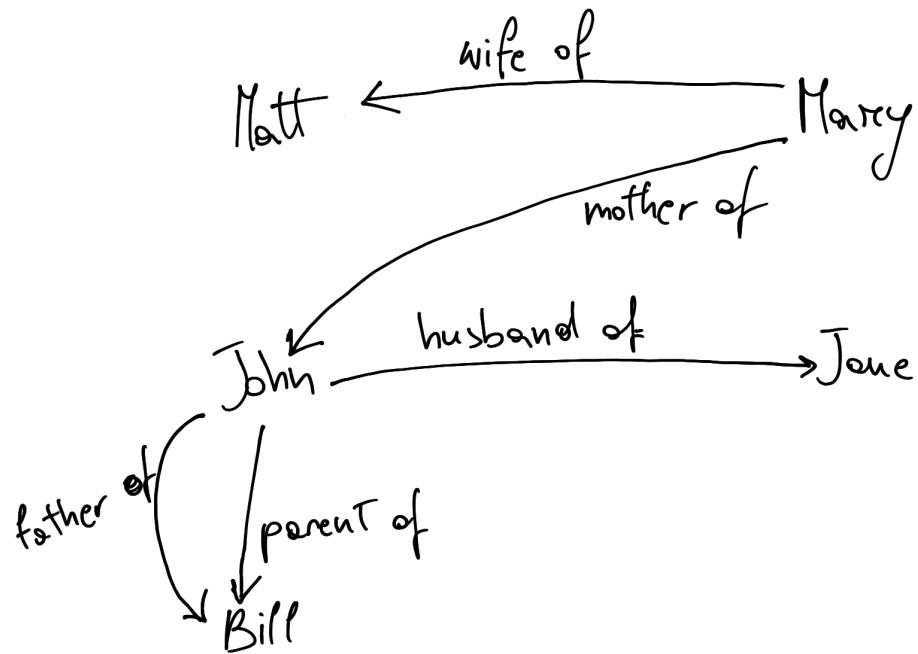
expresses the information: John is a parent of Bill.

Expand the program and the database to express:

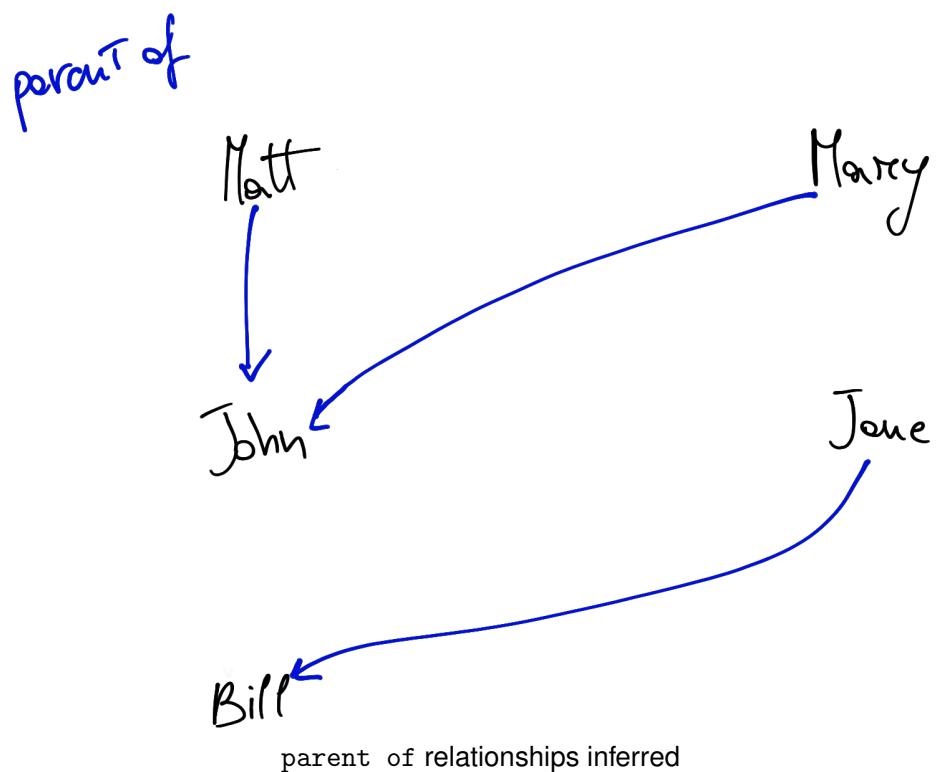
- if X is married with Y, then Y is married with X
- if X is husband of Y, then X is married with Y
- John is husband of Jane
- if X is wife of Y, then X is married with Y
- Mary is wife of Matt
- if X is father of Y, then X is a parent of Y
- John is father of Bill
- if X is mother of Y, then X is a parent of Y

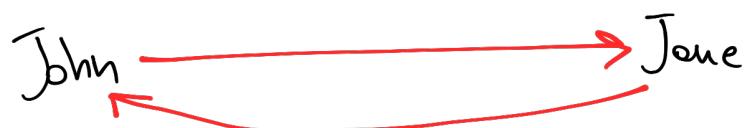
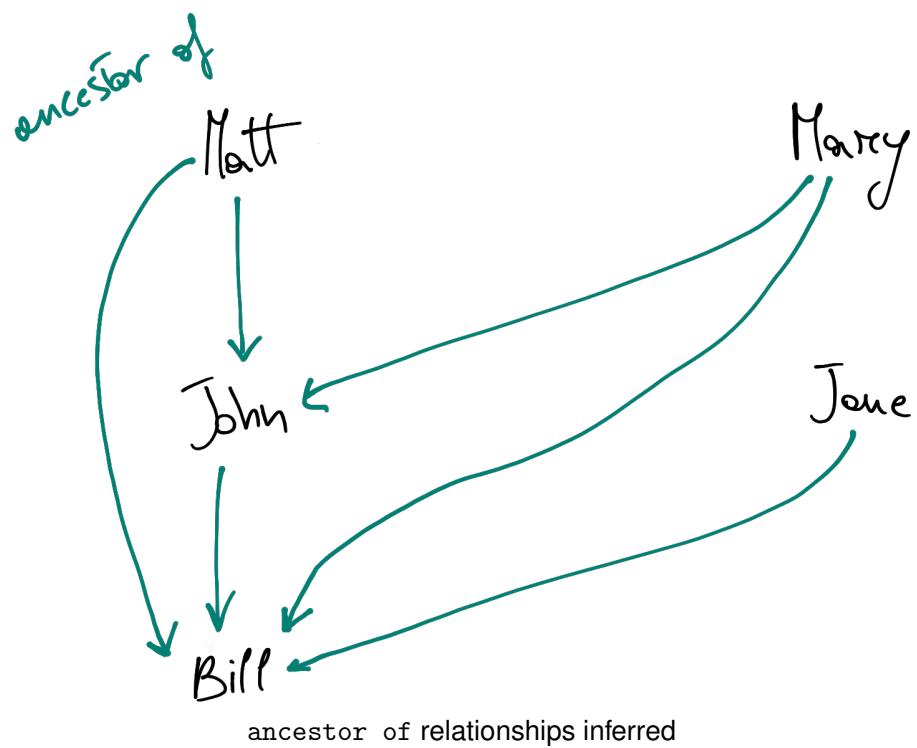
- Mary is mother of John
- if X is married to Y, and Y is a parent of Z, then X is a parent of Z
- if X is an ancestor of Y, and Y is an ancestor of Z, then X is an ancestor of Z

Execute clingo and analyse the output of the reasoning process, critically evaluating the reasoning steps necessary for a given conclusion.



Provided Database





Bill
married to relationships inferred