COMPSCI 367
Tutorial 9: Even More Prolog!
Jonathan Rubin.

1) **Cut**

! *prevents backtracking*

Consider, the following:

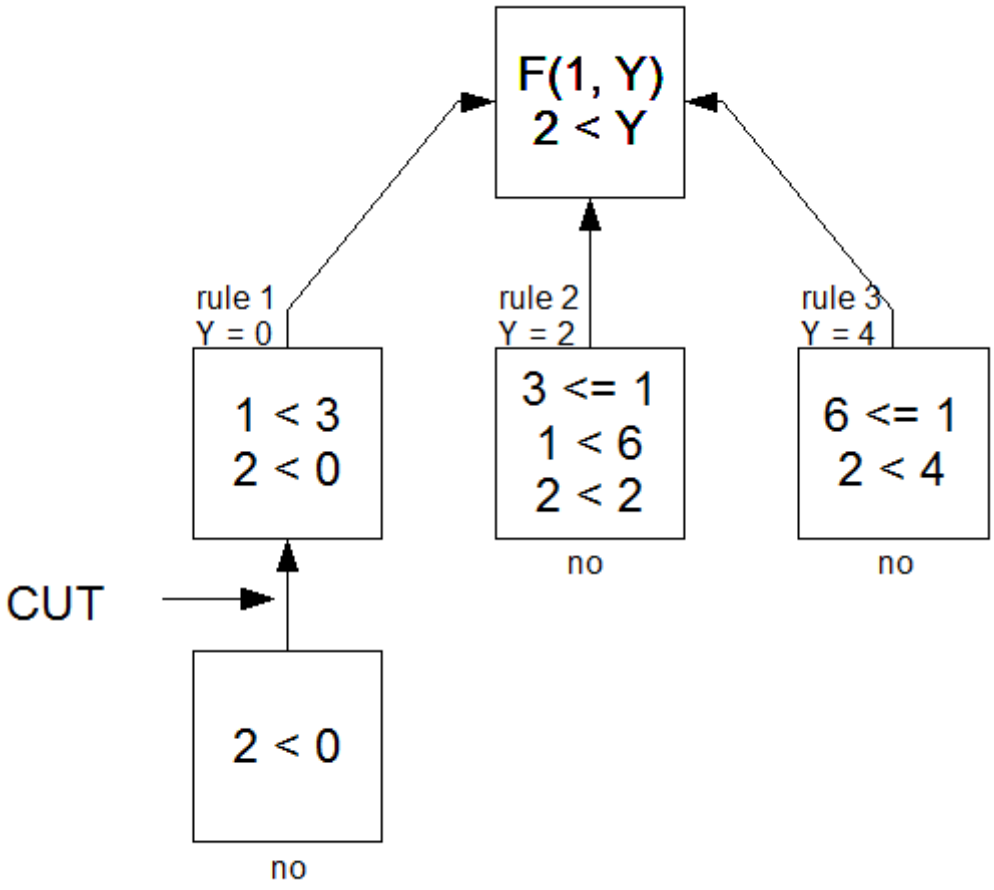
```
f(X, 0) :- X < 3.          % Rule 1

f(X, 2) :- 3 =< X, X < 6.  % Rule 2

f(X, 4) :- 6 =< X.         % Rule 3
```

?- f(1, Y), 2 < Y

Y becomes instantiated to 0 and 2 < 0 fails, but, through backtracking, prolog attempts two useless alternatives.



At the point marked cut, we already know that rules 2 and 3 are bound to fail.

The ! will prevent backtracking at the points at which it appears in the program:

f(X, 0) :- X < 3, !.

f(X, 2) :- 3 =< X, X < 6, !.

f(X, 4) :- 6 =< X.

Consider, a further example:

?- f(7, Y).

Y = 4.

Shows us that the term: 3 =< X is redundant, so we can remove it.

if X < 3 then Y = 0,
otherwise if X < 6 then Y = 2,
otherwise Y = 4.

In prolog:

f(X, 0) :- X < 3, !.

f(X, 2) :- X < 6, !.

f(X, 4).

Now, if we remove the cuts, some solutions may not be appropriate.

Exercise 1. Rewrite the max procedure, using cuts.

2) **Prolog I/O**

Read/Write

read(X)	*reads in some value from the current input stream*
read(stop)	*read in the constant, stop, from the current input stream*
write(C)	*write whatever value C is instantiated to, to the current output stream*
write(hello)	*write the constant, hello, to the current output stream*
writeln(bye)	*write the constant, bye, to the current output stream with a new line at the end*
nl	*output new line*
tab(N)	*output N blanks*

Exercise 2. *Write a prolog program that reads in values and outputs the cube of each value. The program should stop when it reads the value stop, e.g:*

?- cube.
Next item please: 5.

Cube of 5 is 125
Next item please: 12.
Cube of 12 is 1728
Next item please: stop.
yes

See/Tell

see(input.txt)	*Change the current input stream to the file, input.txt*
tell(output.txt)	*Change the current output stream to the file, output.txt*
user	*The user terminal is treated as a file called **user**.*
seen.	*Closes the current input file.*
told.	*Closes the current output file.*

Exercise 3. What changes would you need to make to the above cube procedure and what commands would you need to issue to have cube, read all the input from a file, write the results to a file and close both files when complete?

Exericise 4. Write a program that, given a list of numbers, it outputs a bar graph like so:

?- bars([3, 4, 6, 5]).
***
****
******
*****

3) Mathematic expressions in prolog

?- X is 6/2 + 5*1.
X = 8.

Will be automatically converted into the usual prolog form:

?- X is +(/(6, 2), *(5, 1)).
X = 8.

4). bagof, setof, findall

bagof(X, P, L)	W*ill produce the list L of all the objects X such that a goal P is satisfied.*
setof(X, P, L)	*Orders L and removes duplicates.*
findall(X, P, L)	*All of the objects X are collected regardless of (possibly) different solutions for variables in P that are not shared with X.*