

version 6.4.1 is available for free download at: www.swi-prolog.org

Loading a source file:

- use the pull down menu: File -> Consult
- alternatively, you can use the command: `?- consult('c:/path/file').`
- extension `.pl` is assumed (rules are saved in `file.pl`)

Syntax notes:

- uppercase for variables, lowercase for values
- assignment done with `"is"`. e.g., `X is 3+5.`
- lists specified: `[X|Y]`, `X` is the car (first element), `Y` the cdr (remainder).
- commas indicate "and"
- trace by entering `"trace"`, then use `"c"` to step through the search.

Prolog answers with every match it can find.

Use semi-colon `(;)` to obtain each additional answer (if any). For example:

source file (rules):

```
person(X) :- mother_of(Y,X).
person(Y) :- mother_of(Y,X).
mother_of(mary, carmen).
mother_of(mary, andre).
```

queries:

```
?- person(mary).
true
?- person(andre).
true
?- person(X).
X = carmen <----- user presses ";" to get next answer
X = andre   <----- here too
X = mary    <----- here too
X = mary    <----- here too
false
?-
```

(in the above example, pressing `<cr>` instead of `";"` causes the system to say "Yes", because there are more answers, and then prompt `?-` for the next query.)

Useful operators:

`+` `*` `-` `/` `mod` `=` `>=` `=<` `\=` `<` `>`

A generate-and-test example:

```
number(0). number(1). number(2). number(3). number(4). number(5).
number(6). number(7). number(8). number(9). number(10). number(11).
number(12). number(13). number(14). number(15). number(16). number(17).
number(18). number(19). number(20).

triple(X,Y,Z) :- number(X), number(Y), number(Z),
                 W is X*X + Y*Y, V is Z*Z, W=V.
```

A list processing example:

```
last([X],X).
last([X|Y],Z) :- not(Y=[]), last(Y,Z).
```