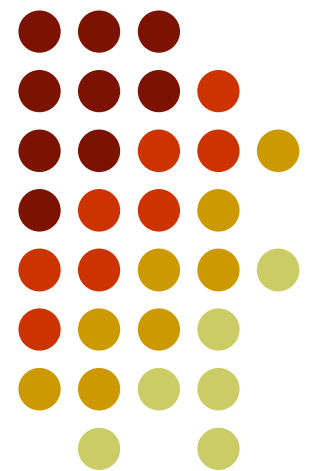


# Introduction to Prolog

---

CS181: Programming Languages





# Topics:

- Facts, rules, questions
- Operators
- Variables, constants
- Interpreter environment



# Facts

- Facts can be as simple as:

`'It is raining today'.`

or

`jill.`

- Useful facts usually contain **predicates**:

`boy(jack).`

`girl(jill).`

`friends(jack, jill).`

`go(jack, jill, 'up the hill').`

`give(jack, jill, crown).`



# Facts

- Names of constants and predicates begin with a lower case letter.
- The predicate (attribute or relationship, if you will) is written first, and the following objects are enclosed by a pair of parentheses and separated by commas.
- Every fact ends with the period character “.”.



# Order

- Order is generally speaking arbitrary, but once you decide on the order, you should be consistent. For example:

`eating(vladimir, burger) .`

intuitively means that “Vladimir is eating a burger”. We could have chosen to put the object of eating (i.e. food) first:

`eating(burger, vladimir) .`

which we can interpret as “A burger is being eaten by Vladimir”. The order is arbitrary in that sense.



# Order

- However,  
`eating(vladimir, burger) .`  
by no means implies that  
`eating(burger, vladimir) .`  
for clearly  
`eating(vladimir, burger) .`  
`eating(burger, vladimir) .`  
mean different things.
- Rule of thumb is to use ‘intuitive’ order, sticking to the English language when possible.



# Rules

- Rules are used to express dependency between a fact and another fact:

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

```
odd(X) :- not even(X) .
```

or a group of facts:

```
son(X, Y) :- parent(Y, X) , male(X) .
```

```
child(X, Y) :- son(X, Y) ; daughter(X, Y) .
```

- Remember that Prolog stands for 'Programming in Logic'?



# Logical operators

Prolog	Read as	Logical operation
<code>:-</code>	IF	Implication
<code>,</code>	AND	Conjunction
<code>;</code>	OR	Disjunction
<code>not</code>	NOT	Negation





# Questions

- A question starts with the “?-” symbol (and ends with a “.”). For example:

```
?- eating(vladimir, burger) .  
yes
```

```
?- eating(vladimir, X) .  
X = burger
```

- Facts, rules and questions are commonly referred to as **clauses**.



# Variables

- Remember the X of the previous slide? X is a variable.
- Variables start with an upper case letter.
- Another example of using variables:

```
?- eating(X, Y) .  
X = vladimir  
Y= burger
```



# Anonymous variables

- If we need to use a variable because the rule requires it, but that variable will never be subsequently used, we can resort to using the anonymous variable, denoted by an underscore, “\_”.
- For example:

```
?- eating(vladimir, _).  
yes
```



# Constants

- In Prolog, constants are either:
  - numbers
  - words starting with a lower case letter
  - enclosed in single quotes
- So, `vladimir` and `burger` and `jack` and `jill` and `'It is raining today'` were constants.



# Arithmetic operators

Symbol	Operation
+	addition
-	subtraction
*	multiplication
/	real division
//	integer division
mod	modulus
**	power



# Arithmetic operators

- For example, in questions:

```
?- X is 3*4.
```

```
X = 12
```

- Or in rules:

```
plus(X, Y, Z) :- Z is X + Y.
```



# Relational operators

Operator	Meaning
$X = Y$	equal to
$X \neq Y$	not equal to
$X < Y$	less than
$X > Y$	more then
$X \leq Y$	less then or equal to
$X \geq Y$	more then or equal to



# Relational operators

- For example, in questions:

`?- age(X, Y), Y < 30.`

- Or in rules:

`minimum(M, N, M) :- M =< N.`

`minimum(M, N, N) :- N =< M.`





# Interpreter environment

- `?- help(what).` Give help on predicate *what*.  
Actually, help has more options.  
Try `?- help(help).` to see them all.
- `?- apropos(what).` Display predicates, functions and sections that have ``what'` (or ``What'`, etc.) in their summary description.
- `?- halt.` Terminates the interpreter.
- `?- consult(file).` Load a program from a local file.
- `?- [filename].` Synonymous with `consult`.



# Interpreter environment

- `?- listing(what).` Lists all lines that start with the predicate *what*.
- `?- listing.` List all lines of the loaded program.
- `;` If there is more than one answer to a question, Prolog will pause after the first one. Typing “;” and hitting enter will lead Prolog to look for the following answers. Just hitting enter will make Prolog stop looking for answers.



# References:

- Clocksin, W.F., and Mellish C.S. *Programming in Prolog*. 4th edition. New York: Springer-Verlag. 1994.
- Aaby, A. *Prolog Tutorial*. Walla Walla College. 1997. On line.  
[http://cs.wwc.edu/~cs\\_dept/KU/PR/Prolog.html](http://cs.wwc.edu/~cs_dept/KU/PR/Prolog.html)
- Fisher, J.R. *Prolog :- tutorial*. CSU Pomona. On line.  
[http://www.intranet.csupomona.edu/~jrfisher/www/prolog\\_tutorial/contents.html](http://www.intranet.csupomona.edu/~jrfisher/www/prolog_tutorial/contents.html)