Lecture 21

**Prolog:**

Some languages closer to math: scheme, Haskell –these are functional languages

Other types such as Prolog also known as programming logic

-can make logical definitions eg.

* Aristottel is a man
* All men are mortal
* Aristottel is mortal

**Constructive proof:**

* Constructs a function at end of proof
* Exhibits subject it is proving as opposed to a unconstructive proof.

**Predicate calculus in synthax**

**Semantics: trying to prove a theorem based on a set of axioms**

**(side note, in prolog “ :- “ should be thought of as a left pointing arrow)**

Example of constructive proof in prolog

(type)?- mortal (x)

(answer) X= aristottle

Prolog isn’t a great language for saying not in programs

(side note, when gathering answers from Prolog programs use “;” to gather all other solutions to a query after receiving first result)

If something can’t be proved in prolog then it is deemed false. It only knows what it is told.

There is a built in number function.

Prolog good for parsing. Can chop and change easily while looking at different parts of a list.

Some other synthax used for prolog:

* ‘Fail’ –something which is never true.
* ‘!’=”cut” at this point in the proof there are no other alternatives to be proved.