Summary of procedural semantics of Prolog:

1. Try to match the current goal. If it matches with a fact, we are done and we can move to the next goal. If there are no more goals, answer 'yes'.
2. Try to match the current goal with the head of a rule. If so, replace the goal with the goal(s) on the RHS of the rule. The first goal (leftmost) goal becomes the current goal.
3. If matching fails, backtrack to a previous goal which has alternatives ('choice point'), undoing any variables bindings.
4. If all goals fail, answer 'no'
5. Matching is unification of terms
6. A term is anything of form P(a1, a2 ...)

Arguments can be of the form $p(q(r))$

functions

Prolog also has a declarative semantics.

Instead of saying "in order to satisfy a goal satisfy these other goals first", we say "this statement is true if these statements are also true"

In other words, instead of an algorithm, we think in terms of a logical proof.

e.g. a :- b, c, d.

a true if b is true <u>and</u> c is true <u>and</u> d is true

or if b and c and d are true, then a is true

So Prolog is capable of expressing complex conditional statements of truth. This is logic - statements about what is the case (i.e. what is true).

Logic starts with simple statements called propositional.

"5 is an integer"

"a dog is a mammal"

"it is raining"

"George Bush is president"

The propositional calculus allows for symbols to represent statements that are either true or false.

"5 is an integer" could be represented by P

We can join statements together using and, or, not.

There are compound statements.

$$\text{"}\underbrace{\text{It is cloudy}}_{P}\ \underset{\wedge}{\text{and}}\ \underbrace{\text{it is raining}}_{Q}\text{"}$$

$$\text{"}\underbrace{\text{It is hot}}_{P}\ \underset{\vee}{\text{or}}\ \underbrace{\text{it is cold}}_{Q}\text{"}$$

There are in fact 16 logical connectives for combinations of 2 statements, but only a few are used. There is a special one, called negation

it is <u>not</u> hot

$\neg P$

A very important connective is implication.

"If $\underbrace{\text{it is cloudy}}_{P}$ then $\underbrace{\text{it is raining}}_{Q}$"

$$P \underset{\text{implies}}{\Rightarrow} Q$$

The actual full meaning of implication is done with a truth table:

Let's draw: $P \wedge Q$

P	$\wedge$	Q
T	T	T
T	F	F
F	F	T
F	F	F

or is: $P \vee Q$

P	$\vee$	Q
T	T	T
T	T	F
F	T	T
F	F	F

not: $\neg P$

$\neg$	P
F	T
T	F

implies $P \Rightarrow Q$

P	$\Rightarrow$	Q
T	T	T
T	F	F
F	T	T
F	T	F

'trivially true'

$P <=> Q$ biconditional, sometimes $\equiv$

P	<=>	Q
T	T	T
T	F	F
F	F	T
F	T	F

$P <=> Q$ is same $P => Q \wedge Q => P$

P	=>	Q	$\wedge$	Q	=>	P
T	T	T	T	T	T	T
T	F	F	F	F	T	T
F	T	T	F	T	F	F
F	T	F	T	F	T	F

$P => Q$ can be written as $\neg P \vee Q$

$\neg$	P	$\vee$	Q
F	T	T	T
F	T	F	F
T	F	T	T
T	F	T	F

This is propositional calculus:

propositional symbols P, Q, R, ... etc.

$\wedge$, $\vee$, $\neg$, $\Rightarrow$, $\Leftrightarrow$ ...

We can derive the truth of certain statements from the truth of others:

If $P \wedge Q$ is true, then P is true

assumptions $\dfrac{P \wedge Q}{P}$ } logical proof

If $P \vee Q$ is true, and P is false, then Q is true

assumption $\dfrac{\begin{array}{c} P \vee Q \\ \neg P \end{array}}{Q}$

Double negation : $\neg\neg P$ is same as P

assumptions $$\frac{\neg\neg P}{P}$$

The derivations are called inference rules, and there are lots of them.