Inference rules:

$$\frac{P \wedge Q}{P}$$ 'and' elimination

$$\frac{P \vee Q \quad \neg P}{Q}$$ 'or' elimination

$$\frac{\neg\neg P}{P}$$ 'double negation' elimination

Very important rule: 'modus ponens'

$$\frac{P \Rightarrow Q \quad P}{Q}$$

$$\begin{array}{l} P \Rightarrow Q \\ Q \Rightarrow R \\ P \\ \hline R \end{array}$$

'forward chaining'

R ← goal proposition

Proofs in Logic

- start with a set of (true) assumptions
- apply inference rules to derive new truths
- finally the 'goal' proposition is derived

Complete proof

1. $P \wedge Q \Rightarrow R$
2. $R \wedge S \Rightarrow T$
3. P
4. Q
5. S

---

6. $P \wedge Q$ (3, 4 and introduction)
7. R (1, 6 modus ponens)
8. $R \wedge S$ (5, 7 and introduction)
9. T (2, 8 modus ponens)

We have proved that T is true based on the assumptions and the rules of inference.

$P \wedge Q \Rightarrow R$

$R \wedge S \Rightarrow T$

P

Q

S

$\Longrightarrow$

```
r :- p, q.
t :- r, s.
p.
q.
s.
```

```
?- t.
   |
   r, s.
   |
   p, q, s.
   |
   q, s.
   |
   s.
   |
  yes
```

In a declarative sense, Prolog is a theorem prover that works by backward chaining through implications (rules)

Prolog also allows for relations.

Logic also has relations (predicates)

We can represent a binary relationship as a "2-place predicate"

e.g. loves(john, jane).

loves(blackboard, wall).

1-place predicates:

e.g. black(cat)

black(tennis-ball)

0-place predicates are propositions.

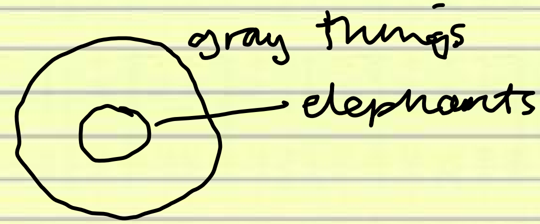
Note that each predicate can be written as a sentence:

John loves Jane.

The cat is black etc.

Sometimes we need to talk about variables.

<u>All</u> elephants are gray.



In logic this is represented by a quantification over a variable.

universal quantifier

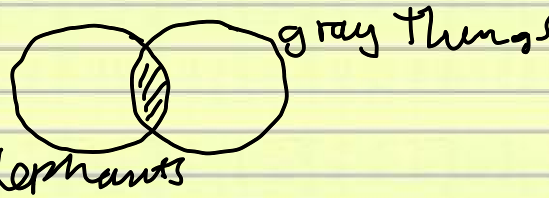
$$\forall x.\ \text{elephant}(x) \Rightarrow \text{gray}(x)$$

For all x, if x is an elephant, then x is gray.

x is an object in the universe of discourse

Note that $\forall x.\ \text{elephant}(x)$ says everything is an elephant

Some elephants are gray.



existential quantifier

$$\exists x.\, elephant(x) \wedge gray(x)$$

There exists an x, s.t. x is an elephant and x is gray.

If we take propositional calculus, add variables, relations (predicates) we get the predicate calculus. Sometimes it is called first-order logic.

$$\forall x.\, P(x) \wedge Q(x) \Rightarrow R(x)$$

this translates into prolog as:

```
r(X) :- p(X), q(X).
```

Every variable in Prolog is universally quantified.

Question: can any true logical statement be expressed in Prolog.

Answer: No.

Prolog can only represent certain kinds of statements.

1. Every statement must at most one term on the RHS of an implication.
2. Negations are not allowed on the RHS of a rule.

These restrict logic to the <u>Horn</u> clause subset

General form of clause:

$$T_0 :- T_1, T_2 \ldots T_n.$$

We cannot have, e.g.

$$\left.\begin{array}{l} p, q :- r, s. \\ \neg p :- q, r, s. \end{array}\right\} \text{not allowed}$$

So Prolog has a declarative semantics which comes from a subset of predicate calculus.

Family relationship examples.

```
mother(X,Y) :- parent(X,Y), female(X).
father(X,Y) :- parent(X,Y), male(X).
sibling(X,Y) :- mother(M,X), mother(M,Y).
sibling(X,Y) :- father(F,X), father(F,Y).
child(X,Y) :- parent(Y,X).
cousin(X,Y) :- parent(P1,X), parent(P2,Y),
               sibling(P1,P2).
ancestor(X,Y) :- parent(X,Y).
ancestor(X,Y) :- parent(Z,Y), ancestor(X,Z).

parent(john,mary).
parent(susan,jake).
parent(jim,john)
parent(jim,susan).
male(john).
male(jake).
male(jim).

female(mary).
female(susan).
```