- Assignment 4 is due Tomorrow.
- "Bad reasoning as well as good reasoning is possible; and this fact is the foundation of the practical side of logic."

  Charles Sanders Peirce, 1877

Since the midterm...

Done:

- Syntax and semantics of propositional definite clauses
- Model a simple domain using propositional definite clauses
- Bottom-up proof procedure computes a consequence set using modus ponens.
- Top-down proof procedure answers a query using resolution.
- The box model provides a way to procedurally understand the top-down proof procedure with depth-first search.

Today:

- Logical variables and Datalog

Clicker Question

In the top-down proof procedure, answer clause

$$yes \text{ :- } happy, \; green, \; good.$$

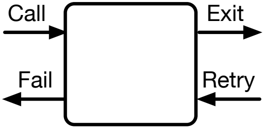
can be resolved with which clause(s) in a KB

(i) $green \text{ :- } good.$
(ii) $good.$
(iii) $sleepy \text{ :- } green.$
(iv) $good \text{ :- } nice, \; green.$
(v) $good \text{ :- } happy, \; green.$

Click on:

A (i), (ii), (iv) and (v) only
B all of the clauses
C (v) only
D none of the clauses, and so the proof fails
E A-D are all incorrect

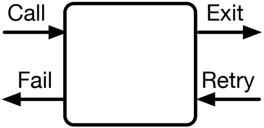
Box Model



Try in Prolog:

$?- \; trace.$

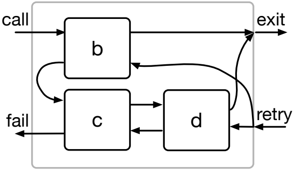
Example: backeg.pl




```
:- dynamic f/0.
a :- b, c, d.
a :- e.
b.
c :- s.
c :- t.
d :- f.
e :- t.
e :- s.
s.
t.
```

Clicker Question

Given Box diagram for a



which of the following is not true

A a :- b must be a clause in the knowledge base

B c is called when b fails

C a exits when b exits

D a fails when c fails

E one of the above is false.

Syntax of Datalog

Propositional definite clauses extended to have

- A variable starts with upper-case letter or with underscore ('_')
- A constant is a sequence of letters, digits or underscore ('_') and starts with lower-case letter
  or is a sequence of digits (numeral)
  or is any sequence of characters between single quotes.
- A predicate symbol starts with lower-case letter.
- A term is either a variable or a constant.
- An atomic symbol (atom) is of the form p or $p(t_1, \ldots, t_n)$ where p is a predicate symbol and t_i are terms.

Clicker Question

For the program

```
hghg(Xyz,hhd) :-
    bbfj(Xyz,gfgf,Haa),
    hhhh(Haa, ggg).
```

Which of the following is not true of this program.

A `Xyz` is a variable

B `hghg` is a constant

C `ggg` is a constant

D `Haa` is a variable

E `hhhh` is a predicate symbol

Variables

- Variables in a clause mean that the clause is true for all values the variables could take (Universal quantification).
- A query with variables is asking for instances of the variables that logically follow from the knowledge base.

Queries and Answers

A query is a way to ask if a body is a logical consequence of the knowledge base:

$$?b_1, \cdots, b_m.$$

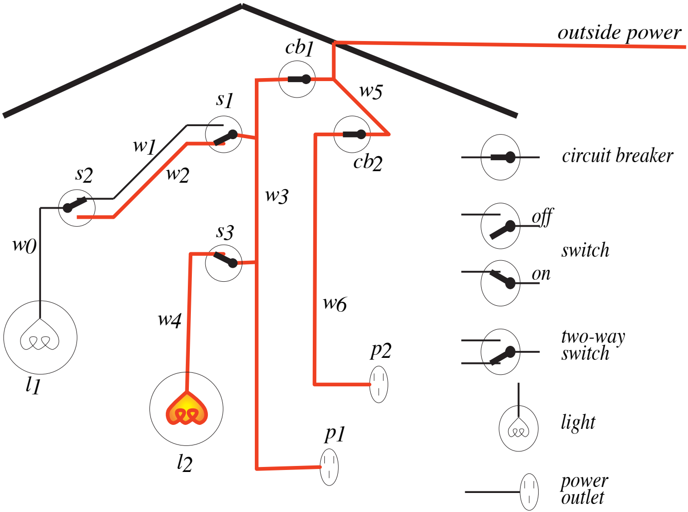
An answer is either

- an instance of the query that is a logical consequence of the knowledge base KB, or
- no (or false) if no instance is a logical consequence of KB.

$$KB = \begin{cases} in(kim, r123). \\ in(X, Y) \texttt{ :- } part\_of(Z, Y),\ in(X, Z). \\ part\_of(r123, cs\_building). \end{cases}$$

Query	Answer
$?part\_of(r123, B).$	$B = cs\_building$
$?part\_of(r023, cs\_building).$	no
$?in(kim, r023).$	no
$?in(kim, B).$	$B = r123;$ $B = cs\_building$

Electrical Environment



Axiomatizing the Electrical Environment (elect_reln.pl)

% $light(L)$ is true if L is a light
$light(l_1)$. $light(l_2)$.
% $down(S)$ is true if switch S is down
$down(s_1)$. $up(s_2)$. $up(s_3)$.
% $ok(D)$ is true if D is not broken
$ok(l_1)$. $ok(l_2)$. $ok(cb_1)$. $ok(cb_2)$.

$?light(l_1)$. $\Longrightarrow$ *yes*
$?light(l_6)$. $\Longrightarrow$ *no*
$?up(X)$. $\Longrightarrow$ $X = s_2$; $X = s_3$

% $connected\_to(X, Y)$ is true if component X is connected to Y
% so electricity will flow from Y to X

$$connected\_to(w_0, w_1) \leftarrow up(s_2).$$
$$connected\_to(w_0, w_2) \leftarrow down(s_2).$$
$$connected\_to(w_1, w_3) \leftarrow up(s_1).$$
$$connected\_to(w_2, w_3) \leftarrow down(s_1).$$
$$connected\_to(w_4, w_3) \leftarrow up(s_3).$$
$$connected\_to(p_1, w_3).$$

$?connected\_to(w_0, W).$ $\Longrightarrow$ $W = w_1$
$?connected\_to(w_1, W).$ $\Longrightarrow$ no
$?connected\_to(Y, w_3).$ $\Longrightarrow$ $Y = w_2;\ Y = w_4;\ Y = p_1$
$?connected\_to(X, W).$ $\Longrightarrow$ $X = w_0, W = w_1;\ \ldots$

% $lit(L)$ is true if the light L is lit

$$lit(L) \leftarrow light(L),\ ok(L),\ live(L).$$

% $live(C)$ is true if there is power coming into C

$$live(Y) \leftarrow$$
$$connected\_to(Y, Z),$$
$$live(Z).$$
$$live(outside).$$

This is a recursive definition of *live*.

Recursion and Mathematical Induction

$$above(X, Y) \text{ :- } on(X, Y).$$
$$above(X, Y) \text{ :- } on(X, Z), \ above(Z, Y).$$

This can be seen as:

- Recursive definition of *above*: prove *above* in terms of a base case (*on*) or a simpler instance of itself; or
- Way to prove *above* by mathematical induction: the base case is when there are no blocks between X and Y, and if you can prove *above* when there are n blocks between them, you can prove it when there are $n + 1$ blocks.

Semantics: General Idea

A semantics specifies the meaning of sentences in the language.

An interpretation specifies:

- what objects (individuals) are in the world
- the correspondence between symbols in the computer and objects and relations in world
  - constants denote individuals
  - predicate symbols denote relations

Formal Semantics

An interpretation is a triple $I = \langle D, \phi, \pi \rangle$, where

- D, the domain, is a nonempty set. Elements of D are individuals.
- ϕ is a mapping that assigns to each constant an element of D. Constant c denotes individual $\phi(c)$.
- π is a mapping that assigns to each n-ary predicate symbol a relation: a function from D^n into $\{TRUE, FALSE\}$.

Example Interpretation

Constants: *phone*, *pencil*, *telephone*.
Predicate Symbol: *noisy* (unary), *left_of* (binary).

- $D = \{$✂, ☎, ✎$\}$.
- $\phi(phone) =$ ☎, $\phi(pencil) =$ ✎, $\phi(telephone) =$ ☎.
- $\pi(noisy)$:

⟨✂⟩	*FALSE*	⟨☎⟩	*TRUE*	⟨✎⟩	*FALSE*

  $\pi(left\_of)$:

⟨✂, ✂⟩	*FALSE*	⟨✂, ☎⟩	*TRUE*	⟨✂, ✎⟩	*TRUE*
⟨☎, ✂⟩	*FALSE*	⟨☎, ☎⟩	*FALSE*	⟨☎, ✎⟩	*TRUE*
⟨✎, ✂⟩	*FALSE*	⟨✎, ☎⟩	*FALSE*	⟨✎, ✎⟩	*FALSE*

Important points to note

- The domain D can contain real things (entities, objects). (e.g., a person, a room, a course). D can't necessarily be stored in a computer.
- $\pi(p)$ specifies whether the relation denoted by the n-ary predicate symbol p is true or false for each n-tuple of individuals.
- If predicate symbol p has no arguments, then $\pi(p)$ is either *TRUE* or *FALSE*.

Truth in an interpretation

- A constant c denotes in I the individual $\phi(c)$.
- Ground (variable-free) atom $p(t_1, \ldots, t_n)$ is
  - *true* in interpretation I if $\pi(p)(\langle \phi(t_1), \ldots, \phi(t_n) \rangle) = TRUE$ in interpretation I and
  - *false* otherwise.
- Ground clause h :- b_1, ..., b_m is *false* in interpretation I if h is *false* in I and each b_i is *true* in I, and is *true* in interpretation I otherwise.

Example Truths

$D = \{$✂, ☎, ✎$\}$
$\phi(phone)=$☎, $\phi(pencil)=$✎,
$\phi(telephone)=$☎.
only ☎ is noisy
✂ is left of ☎ is left of ✎

A true
B false
C Huh?

$noisy(phone)$	*true*
$noisy(telephone)$	*true*
$noisy(pencil)$	*false*
$left\_of(phone, pencil)$	*true*
$left\_of(phone, telephone)$	*false*
$noisy(phone)$:- $left\_of(phone, telephone)$	*true*
$noisy(pencil)$:- $left\_of(phone, telephone)$	*true*
$noisy(pencil)$:- $left\_of(phone, pencil)$	*false*
$noisy(phone)$:- $noisy(telephone)$, $noisy(pencil)$	*true*

Models and logical consequences (recall)

- A knowledge base, KB, is *true* in interpretation I if and only if every clause in KB is *true* in I.
- A model of a set of clauses is an interpretation in which all the clauses are *true*.
- If KB is a set of clauses and g is a conjunction of atoms, g is a logical consequence of KB, written $KB \models g$, if g is *true* in every model of KB.
- That is, $KB \models g$ if there is no interpretation in which KB is *true* and g is *false*.

User's view of Semantics

1. Choose a task domain: intended interpretation.
2. Associate constants with individuals you want to name.
3. For each relation you want to represent, associate a predicate symbol in the language.
4. Tell the system clauses that are true in the intended interpretation: axiomatizing the domain.
5. Ask questions about the intended interpretation.
6. If $KB \models g$, then g must be true in the intended interpretation.

The computer doesn't know the intended interpretation and meaning of symbols, but it is important to convey the intended interpretation in comments for other people and for you in the future.

When we say "give the intended interpretation" means specify in comments what objects exist and the mappings of steps 2 and 3.

Computer's view of semantics

- The computer doesn't have access to the intended interpretation.
- All it knows is the knowledge base.
- The computer can determine if a formula is a logical consequence of KB.
- If $KB \models g$ then g must be true in the intended interpretation.
- If $KB \not\models g$ then there is a model of KB in which g is false. This could be the intended interpretation.