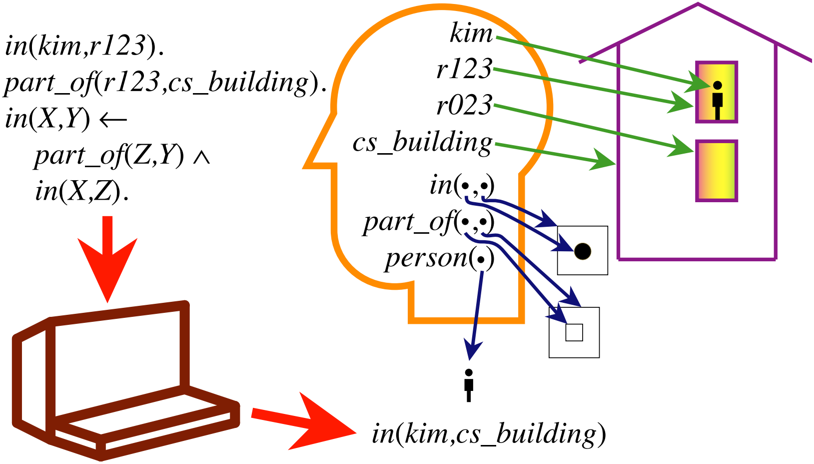
Individuals and Relations

- It is useful to view the world as consisting of individuals (objects, things) and relations among individuals.
- Often features are made from relations among individuals and functions of individuals.
- Reasoning in terms of individuals and relationships can be simpler than reasoning in terms of features, if we can express general knowledge that covers all individuals.
- Sometimes we may know some individual exists, but not which one.
- Sometimes there are infinitely many individuals we want to refer to (e.g., set of all integers, or the set of all stacks of blocks).

Role of Semantics in Automated Reasoning



Features of Automated Reasoning

- Users can have meanings for symbols in their head.
- The computer doesn't need to know these meanings to derive logical consequence.
- Users can interpret any answers according to their meaning.

Decision-theoretic Planning

- flat or modular or hierarchical
- explicit states or features or individuals and relations
- static or finite stage or indefinite stage or infinite stage
- fully observable or partially observable
- deterministic or stochastic dynamics
- goals or complex preferences
- single agent or multiple agents
- knowledge is given or knowledge is learned
- perfect rationality or bounded rationality

Representational Assumptions of Datalog

- An agent's knowledge can be usefully described in terms of *individuals* and *relations* among individuals.
- An agent's knowledge base consists of *definite* and *positive* statements.
- The environment is *static*.
- There are only a finite number of individuals of interest in the domain. Each individual can be given a unique name.

$\Longrightarrow$ Datalog

Syntax of Datalog

- A variable starts with upper-case letter.
- A constant starts with lower-case letter or is a sequence of digits (numeral).
- 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.

Syntax of Datalog (cont)

- A definite clause is either an atomic symbol (a fact) or of the form:

$$\underbrace{a}_{\text{head}} \leftarrow \underbrace{b_1 \wedge \cdots \wedge b_m}_{\text{body}}$$

  where a and b_i are atomic symbols.
- query is of the form $?b_1 \wedge \cdots \wedge b_m$.
- knowledge base is a set of definite clauses.

Example Knowledge Base

$$
\begin{aligned}
&in(kim, R) \leftarrow \\
&\qquad teaches(kim, cs322) \wedge \\
&\qquad in(cs322, R). \\
&grandfather(william, X) \leftarrow \\
&\qquad father(william, Y) \wedge \\
&\qquad parent(Y, X). \\
&slithy(toves) \leftarrow \\
&\qquad mimsy \wedge borogroves \wedge \\
&\qquad outgrabe(mome, Raths).
\end{aligned}
$$

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 & 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 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 $\pi(p)(\langle\phi(t_1), \ldots, \phi(t_n)\rangle) = \textit{TRUE}$ in interpretation I and
- false otherwise.

Ground clause $h \leftarrow b_1 \wedge \ldots \wedge b_m$ 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

In the interpretation given before, which of following are true?

$$noisy(phone)$$
$$noisy(telephone)$$
$$noisy(pencil)$$
$$left\_of(phone, pencil)$$
$$left\_of(phone, telephone)$$
$$noisy(phone) \leftarrow left\_of(phone, telephone)$$
$$noisy(pencil) \leftarrow left\_of(phone, telephone)$$
$$noisy(pencil) \leftarrow left\_of(phone, pencil)$$
$$noisy(phone) \leftarrow noisy(telephone) \wedge noisy(pencil)$$

Example Truths

In the interpretation given before, which of following are true?

$noisy(phone)$	true
$noisy(telephone)$	true
$noisy(pencil)$	false
$left\_of(phone, pencil)$	true
$left\_of(phone, telephone)$	false
$noisy(phone) \leftarrow left\_of(phone, telephone)$	true
$noisy(pencil) \leftarrow left\_of(phone, telephone)$	true
$noisy(pencil) \leftarrow left\_of(phone, pencil)$	false
$noisy(phone) \leftarrow noisy(telephone) \wedge noisy(pencil)$	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

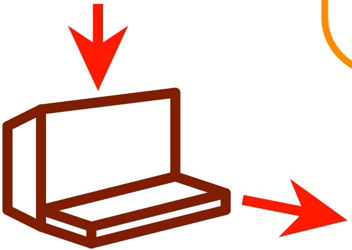
1. Choose a task domain: intended interpretation.
2. Associate constants with individuals you want to name.
3. For each relation you want to represent, associate a predicate symbol in the language.
4. Tell the system clauses that are true in the intended interpretation: axiomatizing the domain.
5. Ask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

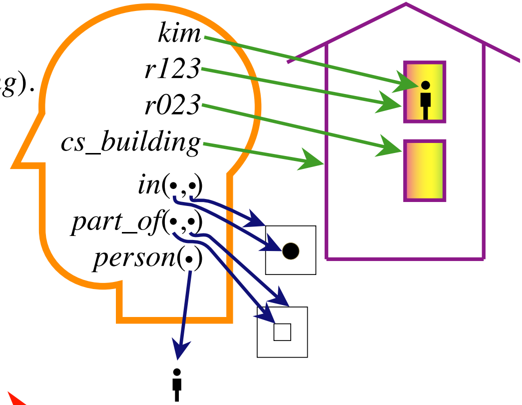
Computer's view of semantics

- 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.

Role of Semantics in an RRS

in(*kim*,*r123*).
part_of(*r123*,*cs_building*).
in(*X*,*Y*) ←
  part_of(*Z*,*Y*) ∧
  in(*X*,*Z*).





in(*kim*,*cs_building*)