Assumptions behind 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.
  An individual can be named.

Datalog syntax

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

\[ a \leftarrow b_1 \land \cdots \land b_m \]

where \( a \) and \( b_i \) are atomic symbols.

- **query** is of the form \( ?b_1 \land \cdots \land b_m \).
- **knowledge base** is a set of definite clauses.

---

**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
An interpretation is a triple \( I = \langle D, \phi, \pi \rangle \), where

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

Example Interpretation

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

- \( D = \{ \text{\ding{115}}, \text{\ding{112}}, \text{\ding{113}} \} \).
- \( \phi(\text{phone}) = \text{\ding{112}}, \phi(\text{pencil}) = \text{\ding{113}}, \phi(\text{telephone}) = \text{\ding{115}} \).
- \( \pi(\text{noisy}): \begin{array}{ccc} \langle \text{\ding{115}}, \text{\ding{112}} \rangle & \text{FALSE} & \langle \text{\ding{112}}, \text{\ding{112}} \rangle & \text{TRUE} & \langle \text{\ding{113}}, \text{\ding{112}} \rangle & \text{FALSE} \\ \langle \text{\ding{115}}, \text{\ding{113}} \rangle & \text{FALSE} & \langle \text{\ding{112}}, \text{\ding{113}} \rangle & \text{FALSE} & \langle \text{\ding{113}}, \text{\ding{113}} \rangle & \text{TRUE} \\ \langle \text{\ding{113}}, \text{\ding{115}} \rangle & \text{FALSE} & \langle \text{\ding{113}}, \text{\ding{112}} \rangle & \text{FALSE} & \langle \text{\ding{113}}, \text{\ding{113}} \rangle & \text{FALSE} \end{array} \)
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 $\pi(p)(\langle \phi(t_1), \ldots, \phi(t_n) \rangle) = TRUE$ in interpretation $I$ and
- false otherwise.
Ground clause $h \leftarrow b_1 \land \ldots \land b_m$ 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)`  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**

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

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

kim

r123

r023

cs_building

in(•,•)

part_of(•,•)

person(•)

in(kim, cs_building)