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.

- $\phi$ 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 $\{TRUE, FALSE\}$. 
Example Interpretation

Constants: phone, pencil, telephone.

Predicate Symbol: noisy (unary), left_of (binary).

\[ D = \{ \text{📞, 📞, ✏️} \}. \]

\[ \phi(\text{phone}) = \text{📞}, \phi(\text{pencil}) = \text{✏️}, \phi(\text{telephone}) = \text{📞}. \]

\[ \pi(\text{noisy}): \begin{array}{ccc} \langle \text{📞} \rangle & \text{FALSE} & \langle \text{☎} \rangle & \text{TRUE} & \langle \text{✎} \rangle & \text{FALSE} \end{array} \]

\[ \pi(\text{left_of}): \]

\[ \begin{array}{ccc} \langle \text{📞, ✏️} \rangle & \text{FALSE} & \langle \text{📞, ☎} \rangle & \text{TRUE} & \langle \text{📞, ✏️} \rangle & \text{TRUE} \\ \langle \text{☎, ✏️} \rangle & \text{FALSE} & \langle \text{☎, ☎} \rangle & \text{FALSE} & \langle \text{☎, ✏️} \rangle & \text{TRUE} \\ \langle \text{✎, ✏️} \rangle & \text{FALSE} & \langle \text{✎, ☎} \rangle & \text{FALSE} & \langle \text{✎, ✏️} \rangle & \text{FALSE} \end{array} \]
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 \( \pi(p)(t'_1, \ldots, t'_n) = \text{TRUE} \), where \( t_i \) denotes \( t'_i \) in interpretation \( I \) and
- false in interpretation \( I \) if \( \pi(p)(t'_1, \ldots, t'_n) = \text{FALSE} \).

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:

\[ \text{noisy(phone)} \quad \text{true} \]
\[ \text{noisy(telephone)} \quad \text{true} \]
\[ \text{noisy(pencil)} \quad \text{false} \]
\[ \text{left_of(phone, pencil)} \quad \text{true} \]
\[ \text{left_of(phone, telephone)} \quad \text{false} \]
\[ \text{noisy(pencil)} \leftarrow \text{left_of(phone, telephone)} \quad \text{true} \]
\[ \text{noisy(pencil)} \leftarrow \text{left_of(phone, pencil)} \quad \text{false} \]
\[ \text{noisy(phone)} \leftarrow \text{noisy(telephone)} \land \text{noisy(pencil)} \quad \text{true} \]
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.
Simple Example

\[ KB = \begin{cases} 
  p \leftarrow q. \\
  q. \\
  r \leftarrow s. 
\end{cases} \]

<table>
<thead>
<tr>
<th></th>
<th>( \pi(p) )</th>
<th>( \pi(q) )</th>
<th>( \pi(r) )</th>
<th>( \pi(s) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( I_1 )</td>
<td>TRUE</td>
<td>TRUE</td>
<td>TRUE</td>
<td>TRUE</td>
</tr>
<tr>
<td>( I_2 )</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
</tr>
<tr>
<td>( I_3 )</td>
<td>TRUE</td>
<td>TRUE</td>
<td>FALSE</td>
<td>FALSE</td>
</tr>
<tr>
<td>( I_4 )</td>
<td>TRUE</td>
<td>TRUE</td>
<td>TRUE</td>
<td>FALSE</td>
</tr>
<tr>
<td>( I_5 )</td>
<td>TRUE</td>
<td>TRUE</td>
<td>FALSE</td>
<td>TRUE</td>
</tr>
</tbody>
</table>

\( KB \models p, \ KB \models q, \ KB \not\models r, \ KB \not\models s \)
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.