Propositions

- An interpretation is an assignment of values to all variables.
- A model is an interpretation that satisfies the constraints.
- Often we don’t want to just find a model, but want to know what is true in all models.
- A proposition is statement that is true or false in each interpretation.
Why propositions?

- Specifying logical formulae is often more natural than filling in tables.
- It is easier to check correctness and debug formulae than tables.
- We can exploit the Boolean nature for efficient reasoning.
- We need a language for asking queries (of what follows in all models) that may be more complicated than asking for the value of a variable.
- It is easy to incrementally add formulae.
- It can be extended to infinitely many variables with infinite domains (using logical quantification).
Human’s view of semantics

Step 1 Begin with a task domain.
Step 2 Choose atoms in the computer to denote propositions. These atoms have meaning to the KB designer.
Step 3 Tell the system knowledge about the domain.
Step 4 Ask the system questions.
— the system can tell you whether the question is a logical consequence.
— You can interpret the answer with the meaning associated with the atoms.
Role of semantics

In computer:

\begin{align*}
\text{light1\_broken} & \leftarrow \text{sw\_up} \\
& \land \text{power} \land \text{unlit\_light1}. \\
\text{sw\_up}. \\
\text{power} & \leftarrow \text{lit\_light2}. \\
\text{unlit\_light1}. \\
\text{lit\_light2}. 
\end{align*}

In user’s mind:

- \text{light1\_broken}: light #1 is broken
- \text{sw\_up}: switch is up
- \text{power}: there is power in the building
- \text{unlit\_light1}: light #1 isn’t lit
- \text{lit\_light2}: light #2 is lit

Conclusion: \text{light1\_broken}

- The computer doesn’t know the meaning of the symbols
- The user can interpret the symbol using their meaning
Simple language: propositional definite clauses

- An **atom** is a symbol starting with a lower case letter.
- A **body** is an atom or is of the form $b_1 \land b_2$ where $b_1$ and $b_2$ are bodies.
- A **definite clause** is an atom or is a rule of the form $h \leftarrow b$ where $h$ is an atom and $b$ is a body.
- A **knowledge base** is a set of definite clauses.
Semantics

- An **interpretation** $I$ assigns a truth value to each atom.
- A body $b_1 \land b_2$ is true in $I$ if $b_1$ is true in $I$ and $b_2$ is true in $I$.
- A rule $h \leftarrow b$ is false in $I$ if $b$ is true in $I$ and $h$ is false in $I$. The rule is true otherwise.
- A knowledge base $KB$ is true in $I$ 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 = \{ p \leftarrow q, q, r \leftarrow s \}

<table>
<thead>
<tr>
<th></th>
<th>p</th>
<th>q</th>
<th>r</th>
<th>s</th>
</tr>
</thead>
<tbody>
<tr>
<td>$l_1$</td>
<td>true</td>
<td>true</td>
<td>true</td>
<td>true</td>
</tr>
<tr>
<td>$l_2$</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td>false</td>
</tr>
<tr>
<td>$l_3$</td>
<td>true</td>
<td>true</td>
<td>false</td>
<td>false</td>
</tr>
<tr>
<td>$l_4$</td>
<td>true</td>
<td>true</td>
<td>true</td>
<td>false</td>
</tr>
<tr>
<td>$l_5$</td>
<td>true</td>
<td>true</td>
<td>false</td>
<td>true</td>
</tr>
</tbody>
</table>
Simple Example

\[ KB = \left\{ \begin{array}{l}
p \leftarrow q. \\
q. \\
r \leftarrow s. 
\end{array} \right. \]

<table>
<thead>
<tr>
<th></th>
<th>( p )</th>
<th>( q )</th>
<th>( r )</th>
<th>( s )</th>
<th>model?</th>
</tr>
</thead>
<tbody>
<tr>
<td>( l_1 )</td>
<td>true</td>
<td>true</td>
<td>true</td>
<td>true</td>
<td>is a model of ( KB )</td>
</tr>
<tr>
<td>( l_2 )</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td>not a model of ( KB )</td>
</tr>
<tr>
<td>( l_3 )</td>
<td>true</td>
<td>true</td>
<td>false</td>
<td>false</td>
<td>is a model of ( KB )</td>
</tr>
<tr>
<td>( l_4 )</td>
<td>true</td>
<td>true</td>
<td>true</td>
<td>false</td>
<td>is a model of ( KB )</td>
</tr>
<tr>
<td>( l_5 )</td>
<td>true</td>
<td>true</td>
<td>false</td>
<td>true</td>
<td>not a model of ( KB )</td>
</tr>
</tbody>
</table>

©D. Poole and A. Mackworth 2010  
Artificial Intelligence, Lecture 5.1, Page 9
Simple Example

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

<table>
<thead>
<tr>
<th></th>
<th>( p )</th>
<th>( q )</th>
<th>( r )</th>
<th>( s )</th>
<th>model?</th>
</tr>
</thead>
<tbody>
<tr>
<td>( I_1 )</td>
<td>true</td>
<td>true</td>
<td>true</td>
<td>true</td>
<td>is a model of ( KB )</td>
</tr>
<tr>
<td>( I_2 )</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td>not a model of ( KB )</td>
</tr>
<tr>
<td>( I_3 )</td>
<td>true</td>
<td>true</td>
<td>false</td>
<td>false</td>
<td>is a model of ( KB )</td>
</tr>
<tr>
<td>( I_4 )</td>
<td>true</td>
<td>true</td>
<td>true</td>
<td>false</td>
<td>is a model of ( KB )</td>
</tr>
<tr>
<td>( I_5 )</td>
<td>true</td>
<td>true</td>
<td>false</td>
<td>true</td>
<td>not a model of ( KB )</td>
</tr>
</tbody>
</table>

Which of \( p, q, r, q \) logically follow from \( KB \)?
Simple Example

\[ KB = \left\{ \begin{array}{l}
p \leftarrow q.
p.
r \leftarrow s.\end{array} \right\} \]

<table>
<thead>
<tr>
<th></th>
<th>p</th>
<th>q</th>
<th>r</th>
<th>s</th>
<th>model?</th>
</tr>
</thead>
<tbody>
<tr>
<td>( l_1 )</td>
<td>true</td>
<td>true</td>
<td>true</td>
<td>true</td>
<td>is a model of KB</td>
</tr>
<tr>
<td>( l_2 )</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td>not a model of KB</td>
</tr>
<tr>
<td>( l_3 )</td>
<td>true</td>
<td>true</td>
<td>false</td>
<td>false</td>
<td>is a model of KB</td>
</tr>
<tr>
<td>( l_4 )</td>
<td>true</td>
<td>true</td>
<td>true</td>
<td>false</td>
<td>is a model of KB</td>
</tr>
<tr>
<td>( l_5 )</td>
<td>true</td>
<td>true</td>
<td>false</td>
<td>true</td>
<td>not a model of KB</td>
</tr>
</tbody>
</table>

Which of \( p, q, r, q \) logically follow from \( KB \)?

\[ KB \models p, \ KB \models q, \ KB \not\models r, \ KB \not\models s \]
1. Choose a task domain: intended interpretation.
2. Associate an atom with each proposition you want to represent.
3. Tell the system clauses that are true in the intended interpretation: axiomatizing the domain.
4. Ask questions about the intended interpretation.
5. If $KB \models g$, then $g$ must be true in the intended interpretation.
6. Users can interpret the answer using their intended interpretation of the symbols.
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.
Representing the Electrical Environment

\[
\begin{align*}
light_{l1} & \quad \text{lit}_{l1} \leftarrow \text{live}_{w0} \land \text{ok}_{l1} \\
light_{l2} & \quad \text{live}_{w0} \leftarrow \text{live}_{w1} \land \text{up}_{s2} \\
down_{s1} & \quad \text{live}_{w0} \leftarrow \text{live}_{w2} \land \text{down}_{s2} \\
up_{s2} & \quad \text{live}_{w1} \leftarrow \text{live}_{w3} \land \text{up}_{s1} \\
up_{s3} & \quad \text{live}_{w2} \leftarrow \text{live}_{w3} \land \text{down}_{s1} \\
ok_{l1} & \quad \text{lit}_{l2} \leftarrow \text{live}_{w4} \land \text{ok}_{l2} \\
ok_{l2} & \quad \text{live}_{w4} \leftarrow \text{live}_{w3} \land \text{up}_{s3} \\
ok_{cb1} & \quad \text{live}_{p1} \leftarrow \text{live}_{w3} \\
ok_{cb2} & \quad \text{live}_{w3} \leftarrow \text{live}_{w5} \land \text{ok}_{cb1} \\
\text{live}_{outside} & \quad \text{live}_{p2} \leftarrow \text{live}_{w6} \\
& \quad \text{live}_{w6} \leftarrow \text{live}_{w5} \land \text{ok}_{cb2} \\
& \quad \text{live}_{w5} \leftarrow \text{live}_{outside} \\
\end{align*}
\]