Truth in a model
The *syntax* of a representation language tells us how we can write a sentence.

In *arithmetic*,

\[ x + y = 4 \] is a well-formed sentence.
\[ x 4 y + \] is not.

In *propositional logic*,

\[ p \lor \neg(q \Rightarrow r) \] is a well-formed sentence.
\[ \land p \neg q r \] is not.
The **semantics** of a formal representation tells us whether each sentence is true, with respect to some possible world.

In *arithmetic*, \( x + y = 4 \)

- is true in a world where \( x \) is 2 and \( y \) is 2
- is false in a world where \( x \) is 1 and \( y \) is 1

In *propositional logic*, \( p \lor q \)

- is true in a world where \( p \) is true and \( q \) is false
- is false in a world where \( p \) is false and \( q \) is false
These possible worlds are called **models**. When we write a truth table, each row corresponds to a model:

<table>
<thead>
<tr>
<th></th>
<th>p</th>
<th>q</th>
<th>p ∨ q</th>
</tr>
</thead>
<tbody>
<tr>
<td>a model</td>
<td>T</td>
<td>T</td>
<td><strong>T</strong></td>
</tr>
<tr>
<td></td>
<td>T</td>
<td>F</td>
<td><strong>T</strong></td>
</tr>
<tr>
<td></td>
<td>F</td>
<td>T</td>
<td><strong>T</strong></td>
</tr>
<tr>
<td></td>
<td>F</td>
<td>F</td>
<td><strong>F</strong></td>
</tr>
</tbody>
</table>

If a proposition \( \varphi \) is true in a model \( m \), we say that “\( m \) satisfies \( \varphi \)” or “\( m \) is a model of \( \varphi \)”. 

**does the model satisfy \( p \lor q \)?**
“Let’s talk film or let’s not talk film—I’m easy.”
\[ p: \quad \text{“The book is in the library.”} \]

\[ p \lor \neg p: \quad \text{“The book is in the library or the book is not in the library.”} \]
Statements of the form $\varphi \lor \neg \varphi$ are always true:

<table>
<thead>
<tr>
<th>$\varphi$</th>
<th>$\neg \varphi$</th>
<th>$\varphi \lor \neg \varphi$</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
</tbody>
</table>
A proposition that is true in all models is called valid.

Valid propositions are also known as tautologies.
IT WAS EITHER THE BEST OF TIMES OR THE WORST OF TIMES...
IT CAN'T BE BOTH!
BRING IT BACK WHEN YOU'VE MADE UP YOUR MIND!

Bill Whitehead,
2013
\( p \): "The book is in the library."

\( p \land \neg p \): "The book is in the library and the book is not in the library."
Statements of the form $\varphi \land \neg \varphi$ are always false:

<table>
<thead>
<tr>
<th>$\varphi$</th>
<th>$\neg \varphi$</th>
<th>$\varphi \land \neg \varphi$</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>F</td>
<td><strong>F</strong></td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td><strong>F</strong></td>
</tr>
</tbody>
</table>
A proposition that isn’t true in any model is called *unsatisfiable*. 

Unsatisfiable propositions are also known as *contradictions*. 
A prisoner was told that by making a statement, he could choose the method of his execution; if the statement was true, he would be shot, and if false, he would be hanged.

The prisoner made the statement, “I shall be hanged”.
Insolubles (paradoxes)

Georg Reisch, *Margarita phylosophica*, 1496
Any compound proposition that is neither a tautology nor a contradiction will be true in some models and false in others.

Such a proposition is called *satisfiable* or *contingent*. 
Entailment and inference
Imagine we know these propositions are all true:

\( p \land q \)

\( q \implies r \)

\( \neg r \lor s \)

Is \( s \) true?
Imagine we know these propositions are all true:

\[ p \land q \]  Then \( q \) must be true

\[ q \Rightarrow r \]

\[ \neg r \lor s \]

Is \( s \) true?
Imagine we know these propositions are all true:

\[ p \land q \] Then \( q \) must be true

\[ q \Rightarrow r \] Then \( r \) must be true

\[ \neg r \lor s \]

Is \( s \) true?
Imagine we know these propositions are all true:

\[ p \land q \]  Then \( q \) must be true

\[ q \Rightarrow r \]  Then \( r \) must be true

\[ \neg r \lor s \]  Then \( s \) must be true

Is \( s \) true? Yup!
Consider a set of premises $K = \{\varphi_1, ..., \varphi_n\}$ and conclusion $\psi$.

$K$ entails $\psi$ iff every model that makes $K$ true also makes $\psi$ true. This can be written as:

$$K \models \psi$$

or

$$\varphi_1, ..., \varphi_n \models \psi$$
Entailment is a relation that holds among logical sentences.

We use *inference methods* to find which propositions are entailed by a set of premises.

A *sound*, i.e., truth-preserving, inference algorithm is one that derives only entailed sentences.

An inference algorithm is *complete* if it can derive any sentence that is entailed.
A familiar example of a sound inference rule is *modus ponens*:

\[
\varphi \Rightarrow \psi, \varphi \\
\quad \\
\psi
\]

or

\[
\left\{ \varphi \Rightarrow \psi, \varphi \right\} \models \psi
\]
Resolution
Propositional resolution is a rule of inference that is *refutation complete*, i.e., using resolution, we can derive a contradiction whenever a set of propositions – a *knowledge base* – is unsatisfiable.
Using resolution is like doing a proof by contradiction.

To see if a proposition, called the goal, is entailed by a KB, we add its negation to the KB and apply resolution.

If we derive a contradiction, the goal is entailed by the KB.

If we run out of propositions to generate without deriving a contradiction, the goal is not entailed by the KB.
Propositional resolution only works for propositions that are in conjunctive normal form (CNF).
$(p \lor \neg q) \land r \land s$

This is in CNF 😊

$p \iff (q \lor (\neg r \Rightarrow s))$

This is not 😞
Every proposition can be converted into an equivalent proposition in CNF:

1. Replace any $\Rightarrow$ or $\Leftrightarrow$ by equivalent formulas using $\land$, $\lor$, and $\neg$:
   
   $a \Rightarrow \beta$ changes to $\neg a \lor \beta$
   $a \Leftrightarrow \beta$ changes to $(a \land \beta) \lor (\neg a \land \neg \beta)$

2. Move negation inward, using de Morgan’s laws and eliminating double negations:
   
   $\neg(a \land \beta)$ changes to $\neg a \lor \neg \beta$
   $\neg(a \lor \beta)$ changes to $\neg a \land \neg \beta$
   $\neg \neg a$ changes to $a$

3. Move conjunction upward, using this equivalence:
   
   $a \lor (\beta \land \gamma)$ changes to $(a \lor \beta) \land (a \lor \gamma)$

4. Collect terms:
   
   $a \lor a$ changes to $a$
Because CNF is entirely regular, we can omit the logical operators and write it as a set of clauses:

\[(p \lor \neg q) \land r \land s\]

\[
\{\{p, \neg q\}, \{r\}, \{s\}\}
\]
Resolution inference rule:

\[ A_1 \lor \cdots \lor A_n \lor B, \neg B \lor C_1 \lor \cdots \lor C_m \]

\[ \Rightarrow \]

\[ A_1 \lor \cdots \lor A_n \lor C_1 \lor \cdots \lor C_m \]

Each \( A_i \) or \( C_j \) is a propositional variable or its negation.
To determine if a knowledge base \( \Delta \) entails a proposition \( \psi \): (i.e., does \( \Delta \models \psi \)):

1. Put \( \Delta \) and the negation of the goal, \( \neg \psi \), into CNF to get a set of clauses, \( S \).
2. Check if \( \{\} \) is in \( S \). If so, \( \Delta \models \psi \).
   
   This means you found a contradiction, e.g., \( \alpha \) and \( \neg \alpha \), which you resolved to wind up with an empty clause.
3. Check if there are two clauses in \( S \) that resolve to produce a clause not already in \( S \). If not, \( \Delta \not\models \psi \).
4. Add the new clause to \( S \) and go to Step 2.
Exercise

Knowledge base:

\[ \text{Rain} \lor \text{Sun} \]

\[ \text{Sun} \Rightarrow \text{Mail} \]

\[ (\text{Rain} \lor \text{Sleet}) \Rightarrow \text{Mail} \]

Is there Mail?
Convert to CNF:

1. Replace any $\Rightarrow$ or $\Leftrightarrow$ by equivalent formulas using $\land$, $\lor$, and $\neg$:
   
   $a \Rightarrow \beta$ changes to $\neg a \lor \beta$
   
   $a \Leftrightarrow \beta$ changes to $(a \land \beta) \lor (\neg a \land \neg \beta)$

2. Move negation inward, using de Morgan's laws and eliminating double negations:
   
   $\neg(a \land \beta)$ changes to $\neg a \lor \neg \beta$
   
   $\neg(a \lor \beta)$ changes to $\neg a \land \neg \beta$
   
   $\neg \neg a$ changes to $a$

3. Move conjunction upward, using this equivalence:
   
   $a \lor (\beta \land \gamma)$ changes to $(a \lor \beta) \land (a \lor \gamma)$

4. Collect terms:
   
   $a \lor a$ changes to $a$
Convert to CNF:

1. Replace any ⇒ or ⇔ by equivalent formulas using ∧, ∨, and ¬:
   - \( a \Rightarrow \beta \) changes to \( \neg a \vee \beta \)
   - \( a \Leftrightarrow \beta \) changes to \( (a \land \beta) \lor (\neg a \land \neg \beta) \)

2. Move negation inward, using de Morgan’s laws and eliminating double negations:
   - \( \neg (a \land \beta) \) changes to \( \neg a \lor \neg \beta \)
   - \( \neg (a \lor \beta) \) changes to \( \neg a \land \neg \beta \)
   - \( \neg \neg a \) changes to \( a \)

3. Move conjunction upward, using this equivalence:
   - \( a \lor (\beta \land \gamma) \) changes to \( (a \lor \beta) \land (a \lor \gamma) \)

4. Collect terms:
   - \( a \lor a \) changes to \( a \)
Write as clauses:

\[
\begin{align*}
\text{Rain} & \lor \text{Sun} \\
\neg\text{Sun} & \lor \text{Mail} \\
\neg\text{Sleet} & \lor \text{Mail} \\
\neg\text{Rain} & \lor \text{Mail}
\end{align*}
\]

\[
\begin{align*}
\{\text{Rain, Sun}\} \\
\{\neg\text{Sun, Mail}\} \\
\{\neg\text{Sleet, Mail}\} \\
\{\neg\text{Rain, Mail}\}
\end{align*}
\]
Add negation of goal:

\{\neg \text{Mail}\}

\{\text{Rain, Sun}\}
\{\neg \text{Sun, Mail}\}
\{\neg \text{Sleet, Mail}\}
\{\neg \text{Rain, Mail}\}
Apply resolution until we get a contradiction:

Contradiction, so $Mail$ was entailed!
Acknowledgments

This lecture incorporates material from:

Brachman & Levesque