asst 5 is posted
Introduction to Knowledge Representation and Reasoning
What is Knowledge Representation?

- Representing facts
- Reasoning with facts

Can computers be meaningful?
Philo of Megara (5C BC): truth tables
Aristotle (322BC): tautologies of proper arguments
Gottfried Leibniz (1646-1716): inference as math-like (bogus) logic
George Boole (1854): *The Laws of Thought* (almost propositional logic)
Gottlob Frege (1879): Conceptual Notation (propositional and first-order logic)
Dartmouth Conference (1956): ‘AI’ coined
Advice Taker (1959): manifesto for declarative knowledge
CYCorp (1984-, www.cyc.com): slightly more complicated than first-order logic
The Advice Taker (1959)

What is KR?
- History of Logic
- Advice Taker
- The PSSH

Prop. Logic
Reasoning

John McCarthy: “AI”, Lisp, time-sharing
Empirical Philosophy = Science

The Physical Symbol System Hypothesis: A physical symbol system has the necessary and sufficient means for general intelligent action. (Newell and Simon)

where a

Symbol is a designating pattern that can be combined with others to form another designating pattern

and

Designation means standing in for something in the world
Propositional Logic
**Propositional Logic**

**What is KR?**

**Prop. Logic**

- **Logic**
- **An Example**
- **Semantics**
- **Break**

**Reasoning**

---

### An Example

**Semantics**

\[
\text{itisraining} \\
\text{iamwet} \\
\text{itisraining} \rightarrow \text{iamwet}
\]

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>(x \wedge y)</th>
<th></th>
<th>x \rightarrow y</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
</tbody>
</table>

\[
\begin{array}{c|c}
 x & x \rightarrow y \\
\hline 
 T & \text{modus ponens} \\
 \hline 
 F & \wedge, \lor, \neg, \rightarrow (\supset, \Rightarrow), \leftrightarrow
\end{array}
\]

---

Wheeler Ruml (UNH)
A logic is a formal system:

- **syntax**: defines sentences
- **semantics**: relation to world
- **inference rules**: reaching new conclusions

three layers: proof, models, reality

flexible, general, principled
If the unicorn is mythical, then it is immortal, but if it is not mythical, then it is a mortal mammal. If the unicorn is either immortal or a mammal, then it is horned. The unicorn is magical if it is horned.
Interpretation: possible world = state of affairs = truth value for each proposition
Meaning: values across all interpretations
Model of $P$: an interpretation in which $P$ is true
Satisfiable: $\exists$ a model
Entailment: if $\alpha$ is true in every model of $KB$, then $KB \models \alpha$
Valid: true in any interpretation

<table>
<thead>
<tr>
<th>$x$</th>
<th>$y$</th>
<th>$(x \land \neg y)$</th>
<th>$z$</th>
<th>$(x \land \neg y) \rightarrow z$</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
</tbody>
</table>
What is KR?

Prop. Logic
- Prop. Logic
- Logic
- An Example
- Semantics

Break

Reasoning

- asst 4
- asst 5
Entailment: if $\beta$ is true in every model of $\alpha$, then $\alpha \models \beta$

computing entailment
soundness, completeness

$\alpha \models \beta$ iff $\alpha \rightarrow \beta$ is valid

$\alpha \models \beta$ iff $\alpha \land \neg \beta$ is unsatisfiable

determining satisfiability is NP-complete
eg, easy to test proof of yes!
Given a formula of boolean logic, is there any assignment of T/F to its variables that makes the entire formula true?

\[(a \lor b \lor c) \land (\neg a \lor b \lor \neg c) \land (\neg a \lor \neg b \lor c) \land (\neg a \lor \neg b \lor \neg c)\]
The Davis-Logemann-Loveland Algorithm (1962)

\textbf{DLL}(\phi): \\
UnitPropagate(\phi) \\
[ PureLiterals(\phi) ] \\
if \phi \text{ is empty, return SAT}
if \phi \text{ contains empty clause, return UNSAT}
\begin{align*}
v & \leftarrow \text{choose a variable} \\
\text{if DLL(SetVariable}(\phi \text{ with } v = \text{true})) = \text{SAT, return SAT} \\
\text{else, return DLL(SetVariable}(\phi \text{ with } v = \text{false}))
\end{align*}

\textbf{UnitPropagate}(\phi):
\begin{align*}
\text{as long as there is a unit clause} \\
\text{SetVariable according to the literal}
\end{align*}

\textbf{SetVariable}(\phi \text{ with } v = \text{value}):
\begin{align*}
\text{remove clauses where } v \text{ appears as } \text{value} \\
\text{remove } v \text{ from clauses where it appears as } \neg\text{value}
\end{align*}
WalkSAT/SKC (1994)

for 1 to \( \text{maxTries} \)
  assign all variables randomly
  from 1 to \( \text{maxFlips} \)
    randomly choose an unsatisfied clause \( c \)
    if one or more of \( c \)'s variables can be flipped while
      breaking nothing,
      randomly choose among those
    else
      with probability \( p \)
      randomly choose one of \( c \)'s variables
    else
      randomly choose among those of \( c \)'s variables that
        minimize breaks
    flip the variable
  if formula satisfied, terminate

\[ p \approx 0.5? \]
DPLL: 50 vars = 1.4 secs, 100 vars = 2.8 min, 140 vars = 4.7 hrs
Local Search for SAT

DPLL: 50 vars = 1.4 secs, 100 vars = 2.8 min, 140 vars = 4.7 hrs

GSAT: 100 vars = 6 secs, 140 vars = 14 secs, 500 vars = 1.6 hrs
Please write down the most pressing question you have about the course material covered so far and put it in the box on your way out.

*Thanks!*