Administrative

Midterm 1
Average = 41.5
Median = 42
Max = 58
Min = 18
Stdev = 8.1
Propositional Logic

- Advantages
  - Simple
  - Generic

- Disadvantage
  - All we have are propositions (facts)
  - Q: What else would we want?

PL for Vacuum World

- Let $D_i, i = 1, \ldots, n$, represent “room $i$ is dirty”
- How to say the following?
  - All the rooms are clean
  - The dog is in some room
  - Any room that the dog has been in is dirty
  - All the upstairs rooms are clean, except the bathrooms
  - Rooms next to dirty rooms have a dusty smell
  - No two adjacent rooms are both dirty
  - The vacuum should visit every dirty room
  - When vacuuming a room, check whether any nearby rooms are dirty
PL can be Cumbersome

- For example, introducing NextTo required $O(n^2)$ new propositional symbols
- Inference exponential in number of propositional symbols
  - Infinite domains... forget about it
- Solution: put more structure on facts

First Order Logic

- Create a richer language by adding additional structure.

<table>
<thead>
<tr>
<th>Propositional Logic</th>
<th>First Order Logic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logically reason over propositions</td>
<td>Logically reason about objects, their properties, and their relationships.</td>
</tr>
<tr>
<td>A superset of PL.</td>
<td></td>
</tr>
</tbody>
</table>
First Order Logic

- **Objects**
  - Rooms, dogs, vacuums, wumpus rooms...

- **Predicates**
  - Tests a property of one or more objects
    - IsBreezy(x), AreAdjacent(x,y), isCleanerThan(x, y)
  - Value (true or false) can be evaluated given a model
  - With no arguments, equivalent to PL propositions.

- **Functions**
  - Names an object as it relates to other objects
    - MotherOf(x), RoomEastOf(x), ChildOf(x,y)
  - By themselves, do not form legal sentences; used along with predicates.
  - With no arguments, is a constant object.... E.g., “John”.

Predicates and Functions

- These look like “function calls” from Java or C, but are *quite* different!

- Defined implicitly, not explicitly.
  - Consider:
    - Male(John) ^ Male(Arnold)
    - AreBrothers(John, Arnold) v AreCousins(John, Arnold)
    - FatherOf(John) = FatherOf(Arnold)

- Relationships are built up from sentences of FOL, in which relationships may appear anywhere.
  - The properties are a matter of *inference*. 
Syntax of First-Order Logic

Sentence $\rightarrow$ AtomicSentence | $\neg$Sentence  
| (Sentence Connective Sentence)  
| Quantifier Variable, ... Sentence

AtomicSentence $\rightarrow$ Predicate(Term,...) | Term = Term

Term $\rightarrow$ Function(Term,...) | Constant | Variable

Connective $\rightarrow$ $\Rightarrow$ | $\land$ | $\lor$ | $\Leftrightarrow$

Quantifier $\rightarrow$ $\forall$ | $\exists$

Constant $\rightarrow$ $A$ | $X_1$ | Room1500EECS | ...

Variable $\rightarrow$ $a$ | $x$ | $s$ | ...

Predicate $\rightarrow$ IsDirty | IsNextTo | IsComfy | ...

Function $\rightarrow$ RoomOnRight | FurnitureInRoom | ...

Equality

- Special built-in predicate symbol
- $Term = Term$
  - An atomic sentence
  - $=$: a binary predicate, equivalence reln
  - Written using infix notation
- Similarly, $\neq$. 
FOL Models

- Just like in PL, we can evaluate the *satisfiability* of a sentence *with respect to a model*.

- What was a model in PL?
  - Just an assignment of true/false to every proposition

- What is a FOL Model?

FOL Models

- A model must specify:
  - Which constants represent the same objects
    - IsCat(Felix) ^ IsDog(Fido)
    - Owner(Felix) = Robert
    - Owner(Fido) = Mr.Haroldson
    - HomeOf(Felix) = HomeOf(Fido)
    - Lives Alone(Robert)
  - Behavior of predicates
    - A truth table over all possible inputs
  - Behavior of functions
    - A map over all possible inputs
Making FOL even more powerful

- So far:
  - A neater way to write propositions
  - Captures object/relation structure, but still need to enumerate all cases
- Real FOL power comes from ability to quantify over objects using variables

Universal Quantifier

- “All the rooms are clean”
  - \( \forall x. \neg \text{Dirty}(x) \)
  - \( \forall x. \text{Room}(x) \Rightarrow \neg \text{Dirty}(x) \)

Equivalent to big conjunction, where \( x \) is replaced by every object in universe
Existential Quantifier

- “The dog is in some room”
  - $\exists x. \text{DogIn}(x)$
  - $\exists x. \text{Room}(x) \land \text{DogIn}(x)$

Existential quantifier

Equivalent to big disjunction, where $x$ is replaced by every object in universe

Sentences with Variables

- $\forall x. \Phi(x)$
  - True in model iff $\Phi(x)$ true no matter what object $x$ denotes

- $\forall x. \text{Room}(x) \Rightarrow \neg\text{Dirty}(x)$
  - True iff $\text{Room}(x) \Rightarrow \neg\text{Dirty}(x)$ is true for all objects $x$
  - Trivially true for any $x$ not a room

- $\exists x. \text{Room}(x) \land \text{DogIn}(x)$
  - True iff there is some $x$ s.t. $x$ is a room with a dog in it

- Compare and contrast:
  - $\exists x. \text{Room}(x) \Rightarrow \text{DogIn}(x)$
  - $\forall x. \text{Room}(x) \land \neg\text{Dirty}(x)$
Sentences about Rock Stars’ Houses

- Rock stars are rich.
- Everybody has a home.
- Rich people have big homes.
- Bruce is a rock star.

Rock Star House KB

\[
\begin{align*}
\forall p. \text{IsRockStar}(p) &\Rightarrow \text{IsRich}(p) \\
\forall p. \text{IsPerson}(p) &\Rightarrow \exists h. \text{HomeOf}(h,p) \\
\forall p. \text{IsRich}(p) &\Rightarrow \exists h. \text{HomeOf}(h,p) \land \text{IsBig}(h) \\
\text{IsRockStar}(Bruce)
\end{align*}
\]
Rock Star House Model

IsRich

IsRockStar

Bruce

IsBig

Bruce’s Home

HomeOf