UBC Department of Computer Science
Undergraduate Events

More details @ https://my.cs.ubc.ca/students/development/events

Visier Info Session
Tues., Nov 3
12 – 1:30 pm
Kaiser 2020/2030

E-Portfolio Competition Info & Training Session
Wed., Nov 4
5:45 – 7:15 pm
DMP 310

Rakuten Info Session
Thurs., Nov 5
5:30 – 6:30 pm
DMP 110

Tri-mentoring/Townhall Event
Tues., Nov 10
5:45 – 7:30 pm
X860, ICICS/CS

Tableau Open House
Thurs., Nov 12
5 – 7 pm
http://waterviewvancouver.com
Lecture Overview

• Finish SAT: example
• First Order Logics
  • Language and Semantics
  • Inference
Satisfiability problems (SAT)

Consider a CNF sentence, e.g.,

\[(\neg D \lor \neg B \lor C) \land (B \lor \neg A \lor \neg C) \land (\neg C \lor \neg B \lor E) \land (E \lor \neg D \lor B) \land (B \lor E \lor \neg C)\]

Is there an interpretation in which this sentence is true (i.e., that is a model of this sentence)?

Many combinatorial problems can be reduced to checking the satisfiability of propositional sentences ......and returning a model
Encoding the Latin Square Problem in Propositional Logic

In combinatorics and in experimental design, a **Latin square** is an $n \times n$ array filled with $n$ different symbols, each occurring exactly once in each row and exactly once in each column. Here is an example:

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>A</td>
<td>B</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>C</td>
<td>A</td>
<td></td>
</tr>
</tbody>
</table>

Here is another one:
Encoding Latin Square in Propositional Logic

Variables must be binary! (They must be propositions)
Each variables represents a color assigned to a cell.
Assume colors are encoded as integers

\[
x_{ijk} \in \{0,1\}
\]

Assuming colors are encoded as follows
(black, 1) (red, 2) (blue, 3) (green, 4) (purple, 5)

\[
x_{233} = 0
\]

True or false, i.e. 0 or 1 with respect to the interpretation represented by the picture?

How many vars/propositions overall? \[n^3\]
Encoding Latin Square in Propositional Logic: Clauses

- Some color must be assigned to each cell (clause of length n);

\[ \forall_{ij} (x_{ij1} \lor x_{ij2} \ldots x_{ijn}) \quad \forall_{ik} (x_{ilk} \lor x_{i2k} \ldots x_{ink}) \]

A.

- No color is repeated in the same row (sets of negative binary clauses);

\[ \forall_{ik} (\neg x_{ilk} \lor \neg x_{i2k}) \land (\neg x_{ilk} \lor \neg x_{i3k}) \ldots (\neg x_{ilk} \lor \neg x_{ink}) \ldots (\neg x_{ink} \lor \neg x_{i(n-1)k}) \]

B.

How many clauses?
Encoding Latin Square in Propositional Logic: Clauses

• Some color must be assigned to each cell (clause of length n);

\[ \forall_{ij} (x_{ij1} \lor x_{ij2} \cdots x_{ijn}) \]

• No color is repeated in the same row (sets of negative binary clauses);

\[ \forall_{ik} (\neg x_{i1k} \lor \neg x_{i2k}) \land (\neg x_{i1k} \lor \neg x_{i3k}) \cdots (\neg x_{i1k} \lor \neg x_{ink}) \cdots (\neg x_{i(n-1)k} \lor \neg x_{ink}) \]

How many clauses?

\[ \sim O(n^4) \]
Encoding Latin Square Problems in Propositional Logic: FULL MODEL

Variables: $x_{ijk}$ cell $i, j$ has color $k$; $i, j, k = 1, 2, \ldots, n$. $x_{ijk} \in \{0, 1\}$

Each variable represents a color assigned to a cell.

Clauses: $O(n^4)$

- Some color must be assigned to each cell (clause of length $n$);
  $\forall_{ij} (x_{ij1} \lor x_{ij2} \ldots x_{ijn})$

- No color is repeated in the same row (sets of negative binary clauses);
  $\forall_{ik} (\neg x_{i1k} \lor \neg x_{i2k}) \land (\neg x_{i1k} \lor \neg x_{i3k}) \ldots (\neg x_{i1k} \lor \neg x_{ink})$

  $\ldots (\neg x_{i(n-1)k} \lor \neg x_{ink})$

- No color is repeated in the same column (sets of negative binary clauses);
  $\forall_{jk} (\neg x_{1jk} \lor \neg x_{2jk}) \land (\neg x_{1jk} \lor \neg x_{3jk}) \ldots (\neg x_{1jk} \lor \neg x_{njk})$

  $\ldots (\neg x_{(n-1)jk} \lor \neg x_{njk})$

  same as for rows
Logics in AI: Similar slide to the one for planning
Relationships between different Logics

(better with colors)

First Order Logic

\( \forall x \exists y p(x, y) \Leftrightarrow \neg q(y) \)

\( p(a_1, a_2) \)

\( \neg q(a_3) \)

Propositional Logic

\( \neg(p \lor q) \rightarrow (r \land s \land t) \)

\( p, r \)

DataLogic

\( p(x) \leftarrow q(x) \land r(x, y) \)

\( r(x, y) \leftarrow s(y) \)

\( s(a_1), q(a_2) \)

PDCL

\( p \leftarrow s \lor f \)

\( r \leftarrow s \lor q \lor p \)

r

p
Lecture Overview

• Finish SAT (example)
• First Order Logics
  • Language and Semantics
  • Inference
Representation and Reasoning in Complex domains (from 322)

- In complex domains expressing knowledge with propositions can be quite limiting.
- It is often natural to consider individuals and their properties.

There is no notion that

- \( up(s_2) \)
- \( up(s_3) \)
- \( ok(cb_1) \)
- \( ok(cb_2) \)
- \( live(w_1) \)
- \( connected(w_1, w_2) \)
By breaking propositions into relations applied to individuals?

- Express **knowledge** that holds for set of individuals (by introducing **variables**)
  
  \[\text{live}(W) \leftarrow \text{connected}_\text{to}(W, W1) \land \text{live}(W1) \land \text{wire}(W) \land \text{wire}(W1).\]

- We can **ask generic queries** (i.e., containing **variables**)
  
  \[? \text{connected}_\text{to}(W, w_1)\]
“Full” First Order Logics (FOL)

LIKE DATALOG: Whereas propositional logic assumes the world contains facts, FOL (like natural language) assumes the world contains

- **Objects**: people, houses, numbers, colors, baseball games, wars, …
- **Relations**: red, round, prime, brother of, bigger than, part of, comes between, …
- **Functions**: father of, best friend, one more than, plus, …

FURTHERMORE WE HAVE

- **More Logical Operators**:….
- **Equality**: coreference (two terms refer to the same object)
- **Quantifiers**
  - Statements about unknown objects
  - Statements about classes of objects
<table>
<thead>
<tr>
<th>Syntax of FOL</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Constants</strong></td>
</tr>
<tr>
<td><strong>Predicates</strong></td>
</tr>
<tr>
<td><strong>Functions</strong></td>
</tr>
<tr>
<td><strong>Variables</strong></td>
</tr>
<tr>
<td><strong>Connectives</strong></td>
</tr>
<tr>
<td><strong>Equality</strong></td>
</tr>
<tr>
<td><strong>Quantifiers</strong></td>
</tr>
</tbody>
</table>
Atomic sentences

Term is a \textit{function} \((\text{term}_1, \ldots, \text{term}_n)\)

or \textit{constant} or \textit{variable}

Atomic sentence is \textit{predicate} \((\text{term}_1, \ldots, \text{term}_n)\)

or \(\text{term}_1 = \text{term}_2\)

E.g.,

\begin{center}
\texttt{predicate(\textit{constant}, \textit{constant})}
\end{center}

• \texttt{Brother(KingJohn, RichardTheLionheart)}

\begin{center}
\texttt{\textit{function(function(\textit{constant}), \textit{function(function(\textit{constant}}))}}
\end{center}

• \texttt{> (Length(LeftLegOf(Richard)), Length(LeftLegOf(KingJohn)))}
Complex sentences

Complex sentences are made from atomic sentences using connectives

\[ \neg S, \quad S_1 \land S_2, \quad S_1 \lor S_2, \quad S_1 \Rightarrow S_2, \quad S_1 \Leftrightarrow S_2. \]

E.g.

\[ \text{Sibling}(\text{KingJohn}, \text{Richard}) \Rightarrow \text{Sibling}(\text{Richard}, \text{KingJohn}) \]
Truth in first-order logic

Like in Prop. Logic a sentence is true with respect to an interpretation.

\[ \neg A \land (B \implies C), \]

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>Truth</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>X</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>V</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>V</td>
</tr>
</tbody>
</table>

In FOL interpretations are much more complex but still same idea: possible configuration of the world.

- 2 objects \( \text{△} \) \( \square \)
- 2 constant symbols \( \{c_1, c_2\} \)
- 1 unary predicate \( P \)
- 1 binary predicate \( Q \)

Is \( \forall x \ P(x) \) true?

A. yes
B. no
Truth in first-order logic

Like in Prop. Logic a sentences is true with respect to an interpretation

In FOL interpretations are much more complex but still same idea: possible configuration of the world

2 objects △ □
2 constant symbols \{C_1, C_2\} \rightarrow \{△, □\}
1 unary predicate P \rightarrow \{△\}
1 binary predicate Q \rightarrow \{\{△, △\}\}

Is \(\forall x \ P(x)\) TRUE? No Yes!
Interpretations for FOL: Example

- Binary relations: 2
- Unary relations: 3
- Functions: 1

5 objects

5 constant symbols

CPSC 422, Lecture 22
Same interpretation with sets

Since we have a one to one mapping between symbols and objects we can use symbols to refer to objects

- \{R, J, RLL, JLL, C\}

Property Predicates
- Person = \{R, J\}
- Crown = \{C\}
- King = \{J\}

Relational Predicates
- Brother = \{<R,J>, <J,R>\}
- OnHead = \{<C,J>\}

Functions
- LeftLeg = \{<R, RLL>, <J, JLL>\}
How many Interpretations with....

5 Objects and 5 symbols
- \{R, J, RLL, JLL, C\}

3 Property Predicates (Unary Relations)
- Person
- Crown
- King

2 Relational Predicates
- Brother
- OnHead

1 Function
- LeftLeg

A. \(2^5\)  B. \(2^{25}\)  C. \(25^2\)

\[5!\times(2^5)^3\times(2^{25})^2\times5^5\]
To summarize: Truth in first-order logic

- Sentences are true with respect to an **interpretation**
- World contains objects (**domain elements**)  
- Interpretation specifies referents for  
  - constant symbols $\rightarrow$ objects  
  - predicate symbols $\rightarrow$ relations  
  - function symbols $\rightarrow$ functional relations

- An atomic sentence $\text{predicate}(\text{term}_1,\ldots,\text{term}_n)$ is true iff the **objects** referred to by $\text{term}_1,\ldots,\text{term}_n$ are in the **relation** referred to by $\text{predicate}$
Quantifiers

Allows us to express

• **Properties of collections of objects** instead of enumerating objects by name

• **Properties of an unspecified object**

Universal: “for all” $\forall$

Existential: “there exists” $\exists$
Universal quantification

\[ \forall \langle \text{variables} \rangle \langle \text{sentence} \rangle \]

Everyone at UBC is smart:
\[ \forall x \text{ At}(x, \text{UBC}) \Rightarrow \text{Smart}(x) \]

\( \forall x \ P \) is true in an interpretation \( I \) iff \( P \) is true with \( x \) being each possible object in \( I \)

Equivalent to the conjunction of instantiations of \( P \)

\[ \text{At}(\text{KingJohn}, \text{UBC}) \Rightarrow \text{Smart}(\text{KingJohn}) \]
\[ \land \text{At}(\text{Richard}, \text{UBC}) \Rightarrow \text{Smart}(\text{Richard}) \]
\[ \land \text{At}(\text{Ralphie}, \text{UBC}) \Rightarrow \text{Smart}(\text{Ralphie}) \]
\[ \land \ldots \]
Existential quantification

\[ \exists \langle \text{variables} \rangle \langle \text{sentence} \rangle \]

Someone at UBC is smart:

\[ \exists x \text{At}(x, \text{UBC}) \land \text{Smart}(x) \]

\[ \exists x \ P \text{ is true in an interpretation I iff } P \text{ is true with } x \text{ being some possible object in I} \]

Equivalent to the disjunction of instantiations of \( P \)

\[ \text{At}(\text{KingJohn}, \text{UBC}) \land \text{Smart}(\text{KingJohn}) \]

\[ \lor \text{At}(\text{Richard}, \text{UBC}) \land \text{Smart}(\text{Richard}) \]

\[ \lor \text{At}(\text{Ralphie}, \text{UBC}) \land \text{Smart}(\text{Ralphie}) \]

\[ \lor \ldots \]
Properties of quantifiers

\( \exists x \ \forall y \) is not the same as \( \forall y \ \exists x \)

\( \exists x \ \forall y \) Loves(x,y)
  • “There is a person who loves everyone in the world”

\( \forall y \ \exists x \) Loves(x,y)
  • “Everyone in the world is loved by at least one person”

**Quantifier duality:** each can be expressed using the other

\( \forall x \) Likes(x,IceCream) \( \rightarrow \exists x \) \( \neg \)Likes(x,IceCream)

\( \exists x \) Likes(x,Broccoli) \( \rightarrow \forall x \) \( \neg \)Likes(x,Broccoli)
Lecture Overview

• Finish SAT (example)
• First Order Logics
  • Language and Semantics
  • Inference
FOL: Inference

Resolution Procedure can be generalized to FOL

- Every formula can be rewritten in logically equivalent CNF
  - Additional rewriting rules for quantifiers

- Similar Resolution step, but variables need to be unified (like in DATALOG)

\[
\{ \neg \text{In}(x, y) \lor \neg \text{Charged}(x) \} \\
\{ \neg \text{In}(z, y) \lor \text{Connected}(z) \} \\
\Rightarrow \neg \text{Charged}(x) \lor \text{Connected}(x)
\]

\[\Theta = \{z/x, y/y\}\]
NLP Practical Goal for FOL: the ultimate Web question-answering system?

Map NL queries into FOPC so that answers can be effectively computed

What African countries are not on the Mediterranean Sea?

\[ \exists c \text{ Country}(c) \land \neg \text{Borders}(c, \text{Med. Sea}) \land \text{In}(c, \text{Africa}) \]

- Was 2007 the first El Nino year after 2001?
  \[ \text{ElNino}(2007) \land \neg \exists y \text{ Year}(y) \land \text{After}(y, 2001) \land \text{Before}(y, 2007) \land \text{ElNino}(y) \]
Learning Goals for today’s class

You can:

• Explain differences between Proposition Logic and First Order Logic
• Compute number of interpretations for FOL
• Explain the meaning of quantifiers
• Describe application of FOL to NLP: Web question answering
Next class Wed

- Ontologies (e.g., Wordnet, Probase), Description Logics…

Assignment-3 out today
Categories & Events

• Categories:
  - `VegetarianRestaurant (Joe's)` - relation vs. object
  - `MostPopular(Joe's, VegetarianRestaurant)`
  - `ISA (Joe's, VegetarianRestaurant)`
  - `AKO (VegetarianRestaurant, Restaurant)`

Events: can be described in NL with different numbers of arguments...

• I ate
• I ate a turkey sandwich
• I ate a turkey sandwich at my desk
• I ate at my desk
• I ate lunch
• I ate a turkey sandwich for lunch
• I ate a turkey sandwich for lunch at my desk
Reification Again

"I ate a turkey sandwich for lunch"

\[ \exists w: \text{Isa}(w, \text{Eating}) \land \text{Eater}(w, \text{Speaker}) \land \text{Eaten}(w, \text{TurkeySandwich}) \land \text{MealEaten}(w, \text{Lunch}) \land \text{Eaten}(w, \text{apple}) \]

Reification Advantage:

• No need to specify fixed number of arguments to represent a given sentence in NL
On October 30, 1989, one civilian was killed in a reported FMLN attack in El Salvador.
Representing Time

Events are associated with **points** or **intervals** in time.

We can impose an ordering on distinct events using the notion of **precedes**.

- **Temporal logic notation:**
  \((\exists w,x,t)\) \text{Arrive}(w,x,t)\)

- **Constraints on variable** \(t\)
  
  \(I\) arrived in New York
  \((\exists t)\) \text{Arrive}(I,\text{New York},t) \land \text{precedes}(t,\text{Now})\)
Interval Events

Need $t_{\text{start}}$ and $t_{\text{end}}$

“*She was driving to New York until now*”

$$\exists t_{\text{start}}, t_{\text{end}}, e, i \quad \text{ISA}(e, \text{Drive}) \land \text{Driver}(e, \text{She}) \land$$
$$\text{Dest}(e, \text{NewYork}) \land \text{IntervalOf}(e, i) \land$$
$$\text{Endpoint}(i, t_{\text{end}}) \land \text{Startpoint}(i, t_{\text{start}}) \land$$
$$\text{Precedes}(t_{\text{start}}, \text{Now}) \land$$
$$\text{Equals}(t_{\text{end}}, \text{Now})$$
Review --- Syntactic Ambiguity

- FOPC provides many ways to represent the same thing.
- E.g., “Ball-5 is red.”
  - HasColor(Ball-5, Red)
    - Ball-5 and Red are objects related by HasColor.
  - Red(Ball-5)
    - Red is a unary predicate applied to the Ball-5 object.
  - HasProperty(Ball-5, Color, Red)
    - Ball-5, Color, and Red are objects related by HasProperty.
  - ColorOf(Ball-5) = Red
    - Ball-5 and Red are objects, and ColorOf() is a function.
  - HasColor(Ball-5(), Red())
    - Ball-5() and Red() are functions of zero arguments that both return an object, which objects are related by HasColor.
  - ...

- This can GREATLY confuse a pattern-matching reasoner.
  - Especially if multiple people collaborate to build the KB, and they all have different representational conventions.
Review --- Syntactic Ambiguity --- Partial Solution

- FOL can be TOO expressive, can offer TOO MANY choices

- Likely confusion, especially for teams of Knowledge Engineers

- Different team members can make different representation choices
  - E.g., represent “Ball43 is Red.” as:
    - a predicate (= verb)? E.g., “Red(Ball43)”?
    - an object (= noun)? E.g., “Red = Color(Ball43))”?
    - a property (= adjective)? E.g., “HasProperty(Ball43, Red)”?

- PARTIAL SOLUTION:
  - An upon-agreed ontology that settles these questions
  - Ontology = what exists in the world & how it is represented
  - The Knowledge Engineering teams agrees upon an ontology BEFORE they begin encoding knowledge
Relation Between Tenses and Time

Relation between simple verb tenses and points in time is not straightforward

Present tense used like future:

- *We fly from Baltimore to Boston at 10*

Complex tenses:

- *Flight 1902 arrived late*
- *Flight 1902 had arrived late*

Representing them in the same way seems wrong....
Reference Point

Reichenbach (1947) introduced the notion of Reference point \((R)\), separated out from Utterance time \((U)\) and Event time \((E)\).

Example:

- When Mary's flight departed, I ate lunch
- When Mary's flight departed, I had eaten lunch

*Departure* event specifies the reference point.
Today Feb 7

Semantics / Meaning / Meaning Representations
Linguistically relevant Concepts in FOPC / FOL
Semantic Analysis
Limited expressiveness of propositional logic

KB contains "physics" sentences for every single square

For every time $t$ and every location $[x,y],
\quad L_{x,y} \land FacingRight^t \land Forward^t \Rightarrow L_{x+1,y}

Rapid proliferation of clauses.

First order logic is designed to deal with this through the introduction of variables.
Logical agents apply inference to a knowledge base to derive new information and make decisions.

Basic concepts of logic:
- syntax: formal structure of sentences
- semantics: truth of sentences wrt models
- entailment: necessary truth of one sentence given another
- inference: deriving sentences from other sentences
- soundness: derivations produce only entailed sentences
- completeness: derivations can produce all entailed sentences

Resolution is complete for propositional logic.

Forward, backward chaining are linear-time, complete for Horn clauses.

Propositional logic lacks expressive power.
Resolution Algorithm

- The resolution algorithm tries to prove: \( KB \models \alpha \) equivalent to \( KB \wedge \neg \alpha \) unsatisfiable
- \( KB \wedge \neg \alpha \) is converted in CNF
- Resolution is applied to each pair of clauses with complementary literals
- Resulting clauses are added to the set (if not already there)
- Process continues until one of two things can happen:

1. Two clauses resolve in the empty clause. i.e. query is entailed
   \[ p \rightarrow \emptyset \rightarrow KB \models \neg \alpha \Rightarrow KB \models \alpha \]

2. No new clauses can be added: We find no contradiction, there is a model that satisfies the sentence and hence we cannot entail the query.
   \[ KB \models \neg \alpha \Rightarrow KB \not\models \alpha \]
Resolution example

\[ KB = (A \iff (B \lor C)) \land \neg A \]

\[ \alpha = \neg B \]

True!

False in all worlds
Resolution algorithm

Proof by contradiction, i.e., show $KB \land \neg \alpha$ unsatisfiable

function PL-RESOLUTION($KB, \alpha$) returns true or false
    inputs: $KB$, the knowledge base, a sentence in propositional logic
             $\alpha$, the query,
    clauses $\leftarrow$ the set of clauses in the CNF representation of $KB \land \neg \alpha$
    new $\leftarrow \{\}$
    loop do
        for each $C_i, C_j$ in clauses do
            resolvents $\leftarrow$ PL-RESOLVE($C_i, C_j$)
            if resolvents contains the empty clause then return true
            new $\leftarrow$ new $\cup$ resolvents
            if new $\subseteq$ clauses then return false
            clauses $\leftarrow$ clauses $\cup$ new
Requirements for Meaning Representations

Answer questions Y/N

Inference

Pres/ Arguments
Common Meaning Representations

I have a car

\[ \exists x, y \text{Having}(x) \land \text{Haver}(\text{Speaker}, x) \land \text{HadThing}(y, x) \land \text{Car}(y) \]

FOPC

Semantic Nets

Frames

Common foundation: structures composed of symbols that correspond to objects and relationships
Today Feb 7

Semantics / Meaning / Meaning Representations

Linguistically relevant Concepts in FOPC/FOL

Semantic Analysis
Linguistically Relevant Concepts in FOPC

Categories & Events (Reification)

Representing Time

Beliefs (optional, read if relevant to your project)

Aspects (optional, read if relevant to your project)

Description Logics (optional, read if relevant to your project)

restrict FOL

Semantic Web
Variables

A big part of using FOL involves keeping track of all the variables while reasoning. **Substitution lists** are the means used to track the value, or binding, of variables as processing proceeds.

\[
\{ \text{var/var/term, var/var/term, var/var/term...} \}
\]
Examples

\( \forall x \text{Cat}(x) \rightarrow \text{Annoying}(x) \)

\( \{x / \text{Felix}\} \)

\( \text{Cat(\text{Felix})} \rightarrow \text{Annoying(\text{Felix})} \)
\[ \forall x, y \: \text{Near}(x, y) \rightarrow \text{Near}(y, x) \]
\[ \{x / \text{McCoy}, y / \text{ChemE}\} \]
\[ \text{Near}(\text{McCoy}, \text{ChemE}) \rightarrow \text{Near}(\text{ChemE}, \text{McCoy}) \]
Semantics: Worlds

The world consists of objects that have properties.

- There are relations and functions between these objects.
- Objects in the world, individuals: people, houses, numbers, colors, baseball games, wars, centuries
  - Clock A, John, 7, the-house in the corner, Tel-Aviv, Ball43
- Functions on individuals:
  - father-of, best friend, third inning of, one more than
- Relations:
  - brother-of, bigger than, inside, part-of, has color, occurred after
An interpretation of a sentence (wff) is an assignment that maps
• Object constant symbols to objects in the world,
• n-ary function symbols to n-ary functions in the world,
• n-ary relation symbols to n-ary relations in the world

Given an interpretation, an atomic sentence has the value “true” if it denotes a relation that holds for those individuals denoted in the terms. Otherwise it has the value “false.”
An interpretation satisfies a wff (sentence) if the wff has the value “true” under the interpretation.

Model: A domain and an interpretation that satisfies a wff is a model of that wff

Validity: Any wff that has the value “true” under all interpretations is valid

Any wff that does not have a model is inconsistent or unsatisfiable

If a wff w has a value true under all the models of a set of sentences KB then KB logically entails w
Interpretations for FOL: Example

- Person R
- Person J
- Brother R and J
- Left leg of R and J
- Crown on the head of J
- Person king

Diagram shows relationships and attributes between individuals and objects.
Pros and cons of propositional logic

😊 Propositional logic is declarative

😊 Propositional logic allows partial/disjunctive/negated information
   • (unlike most data structures and databases)

😊 Propositional logic is compositional:
   • meaning of $B_{1,1} \land P_{1,2}$ is derived from meaning of $B_{1,1}$ and of $P_{1,2}$

😊 Meaning in propositional logic is context-independent
   • (unlike natural language, where meaning depends on context)
   •

😊 Propositional logic has very limited expressive power
   • (unlike natural language)
   • E.g., cannot say "pits cause breezes in adjacent squares"
     ✓ except by writing one sentence for each square