Overview

⇒ Representation and Reasoning System
• Syntax of Datalog
• Semantics of Datalog
• Adding Variables to Semantics
• Models
• Logical Consequence
• Two Views of Semantics

Previous Class

• Introduced a task domains: robot delivery and wiring
• Introduced the symbolic approach
  - Symbols have meaning to the knowledge engineer
  - Symbols used to build a knowledge base that computer is told about
    + Facts about the world
    + Rules about the world
  - Computer reasons with the facts and rules to make new conclusions
A Representation and Reasoning System (RRS) is made up of
- Formal language (syntax):
  + Specifies the legal sentences (the range of things that can be said)
- Semantics:
  + Specifies the meaning of the symbols (for your domain)
  + Specifies what is a correct conclusion
- Reasoning theory or proof procedure:
  + Specification of how an answer can be produced
  + Can be nondeterministic

Implementation of an RRS
- Reasoning procedure
  + Resolves nondeterminism of reasoning theory

Different RRS’s
- Different RRS’s
  - With different syntaxes
    + Actually different connectors: ways to build complex expressions
  - Or with different semantics for connectives
- Different RRS’s good for different domains
- The richer the syntax, the more difficult the reasoning procedure
  ⇒ Choose the simplest RRS possible for your application
Simplifying Assumptions of Initial RRS

• An agent’s knowledge can be usefully described in terms of individuals and relations among individuals
• An agent’s knowledge base consists of definite and positive statements
• The environment is static
• Only a finite number of individuals of interest in the domain
• Each individual can be given a unique name
⇒ Datalog

Overview

• Representation and Reasoning System
  ⇒ Syntax of Datalog
• Semantics of Datalog
• Adding Variables to Semantics
• Models
• Logical Consequence
• Two Views of Semantics
Syntax of Datalog

- Variable
  - starts with upper-case letter
- Constant
  - starts with lower-case letter or is a sequence of digits (numeral)
- Predicate symbol
  - starts with lower-case letter
- Term
  - either a variable or a constant
- Atomic symbol (atom)
  - of the form $p$ or $p(t_1, ..., t_n)$ where $p$ is a predicate symbol and $t_i$ are terms

More Syntax of Datalog

- Definite Clause
  - either an atomic symbol (a fact) or of the form
    \[ a \leftarrow b_1 \land ... \land b_m \]
- Query
  - of the form $\exists b_1 \land ... \land b_m$
- Knowledge Base
  - set of definite clauses

⇒ Syntax allows us to write sentences about the world
- Whether sentences are true or not depends on what the symbols mean,
  which will be specified by the semantics
Example

- Knowledge base
  - male(william)
  - male(george)
  - female(sally)
  - father(william,george)
  - father(george,sally)
  - person(X) ← female(X)
  - person(X) ← male(X)
  - parent(X,Y) ← father(X,Y)
  - grandfather(Z,X) ← father(Z,Y) ∧ parent(Y,X)

- What are the constants?
- What are the predicate symbols?
- What are the variables?
- Whether knowledge base is correct depends on semantics

Overview

- Representation and Reasoning System
- Syntax of Datalog
  ⇒ Semantics of Datalog
- Adding Variables to Semantics
- Models
- Logical Consequence
- Two Views of Semantics
Interpretation

An interpretation is a triple $I = (D, \phi, \pi)$ where

- $D$ the domain, is a nonempty set. Elements of $D$ are individuals
- $\phi$ maps each constant to an element of $D$
  - Constant $c$ denotes individual $\phi(c)$.
- $\pi$ maps each n-ary predicate symbol to subset of $D^n$
  - Alternatively, can think of $\pi$ as mapping each tuple $D^n$ to true or false
  - **NOTE**: it does not map it to a subset of constants
  - Common mistake, don’t make it on your homework
Example Interpretation

• D is the set of people
  - William, George, Sally
  - It is the actual people, not the names

• $\phi$ maps constants of syntax
to objects in the domain
  - $\phi(william) = \text{William}$

• Knowledge Engineer decides D and mapping of all constants to D

Example Continued

• William and George are male, Sally is female

• Lets have $\pi$ map
  - $\pi$ maps $\text{male}$ to $\{<\text{William}>,<\text{George}>\}$
  - $\pi$ maps $\text{female}$ to $\{<\text{Sally}>\}$

• Knowledge Engineer decides on mapping of predicates
  - Must decide on the mapping for all predicates
  - Hence, must do mapping for $\text{male}$, even if no facts in $\mathcal{KB}$ about $\text{male}$

• This is an example of an intended interpretation:
  - The interpretation that the knowledge engineer has in mind when coming up with language and knowledge base
Second Example

- Example: (focus on all interpretations, not just intended one)
  - Language with constants $a$ and $b$ and 1-ary predicate $\text{female}(\_)$
  - Domain with $D = \{x, y, z\}$
  - How many different $\phi$’s?

<table>
<thead>
<tr>
<th>$\phi_i(a)$</th>
<th>$\phi_i(b)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\phi_1$</td>
<td></td>
</tr>
<tr>
<td>$\phi_2$</td>
<td></td>
</tr>
<tr>
<td>$\phi_3$</td>
<td></td>
</tr>
<tr>
<td>$\phi_4$</td>
<td></td>
</tr>
<tr>
<td>$\phi_5$</td>
<td></td>
</tr>
<tr>
<td>$\phi_6$</td>
<td></td>
</tr>
<tr>
<td>$\phi_7$</td>
<td></td>
</tr>
<tr>
<td>$\phi_8$</td>
<td></td>
</tr>
<tr>
<td>$\phi_9$</td>
<td></td>
</tr>
</tbody>
</table>

Example Continued

- How many $\pi$’s?

<table>
<thead>
<tr>
<th>$x \in \pi_i(\text{female})$</th>
<th>$y \in \pi_i(\text{female})$</th>
<th>$z \in \pi_i(\text{female})$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\pi_1$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\pi_2$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\pi_3$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\pi_4$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\pi_5$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\pi_6$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\pi_7$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\pi_8$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- How many different interpretations are there altogether (different combinations of $\phi$ and $\pi$)?
Determining Truth of Ground Atoms in $I$

- Ground atom has no variables
- $p(t_1, ..., t_n)$ maps to true if $(\phi(t_1), ..., \phi(t_n)) \in \pi(p)$ otherwise to false
- What does $male(george)$ map to?
  - $\phi(george) = George$
  - $\pi(male) = \{<William>,<George>\}$
  - $<George> \in \{<William>,<George>\}$
  - So it maps to true
- For predicates without arguments
  - $\pi(p)$ is either the set with the empty tuple $\{<>\}$ or it is empty $\{\}$
  - Semantics of Ground Atoms comes from interpretation

Semantics of Connectives

- Still need to specify what $\land$ and $\leftarrow$ mean

<table>
<thead>
<tr>
<th>$p$</th>
<th>$q$</th>
<th>$p \land q$</th>
<th>$p \leftarrow q$</th>
</tr>
</thead>
<tbody>
<tr>
<td>true</td>
<td>true</td>
<td>true</td>
<td>true</td>
</tr>
<tr>
<td>true</td>
<td>false</td>
<td>false</td>
<td>true</td>
</tr>
<tr>
<td>false</td>
<td>true</td>
<td>false</td>
<td>false</td>
</tr>
<tr>
<td>false</td>
<td>false</td>
<td>false</td>
<td>true</td>
</tr>
</tbody>
</table>

- Nota bene!
  + $p \leftarrow q$ is true when both $p$ and $q$ are false
  + $p \land q$ doesn’t always correspond to ‘english’ meaning
- Thus $h \leftarrow b_1 \land ... \land b_m$ is false in interpretation $I$
  if $h$ is false in $I$ and each $b_i$ is true in $I$
  - Semantics of $\land$ and $\leftarrow$ part of Datalog
Limitations of Datalog

- Even if every object is male or female, both predicates needed
  - Datalog does not include an operator that means negation
- Cannot write a rule that ensures just one of \textit{male} and \textit{female} is true for any person
  - Up to knowledge engineer to ensure each person is just one of them
  - More expressive formalisms can handle this (negative knowledge)

Example

- Is \textit{male}(george) \land \textit{female}(sally) true in I?
- Is \textit{male}(george) \leftarrow \textit{female}(sally) true in I?
- Is \textit{male}(george) \leftarrow \textit{female}(william) true in I?
- Is \textit{female}(george) \leftarrow \textit{male}(william) true in I?
Overview

• Representation and Reasoning System
• Syntax of Datalog
• Semantics of Datalog
  ⇒ Adding Variables to Semantics
• Models
• Logical Consequence
• Two Views of Semantics

Semantics & Variables

• How do we interpret clauses such as
  \( \text{person}(X) \leftarrow \text{female}(X) \)

• Clause is true if it is true for all values of \( X \)
  - \( \text{person}(X) \) must be true whenever \( \text{female}(X) \) is true
  - Remember, knowledge engineer had to specify mapping for all predicates, even room
    - \( \pi(\text{female}) \subseteq \pi(\text{person}) \)

• It really has a universal quantifier
  - For all \( X \) \( \text{female}(X) \leftarrow \text{person}(X) \)

• So, variables have an implicit universal quantifier over the clause
Variable Assignment: Formal Definition

- Define a variable assignment $\rho$
  - Maps each variable to some object in the domain
- Together $\rho$ and $\phi$ assign each term to some object in the domain
- Together $\rho$ and interpretation $I$ map every clause to true or false
  + Even ungrounded ones
- Now we can say:
  - A clause is true in an interpretation if it is true for all variable assignments

Example

- Interpretation $I$
  - $\pi(male) = \{<\text{William}>,<\text{George}>\}$
  - $\pi(female) = \{<\text{Sally}>\}$
  - $\pi(person) = \{<\text{William}>,<\text{George}>,<\text{Sally}>\}$
- Are the following true?
  - $\text{person}(X) \leftarrow \text{male}(X)$
  - $\text{person}(X) \leftarrow \text{female}(X)$
  - $\text{male}(X) \land \text{female}(X)$
  - $\text{male}(X) \lor \text{female}(X)$
  - $\text{person}(X) \leftarrow \text{female}(X) \land \text{male}(\text{William})$
Overview

- Representation and Reasoning System
- Syntax of Datalog
- Semantics of Datalog
- Adding Variables to Semantics
  ⇒ Models
- Logical Consequence
- Two Views of Semantics

Sets of Clauses

- A set of clauses is true in an interpretation if each clause is true in the interpretation
  - Note that we universally quantify for the variables over each clause
  - In other words, if two clauses use the same variables, it is the same as if they used different variables
    
    \[
    \begin{align*}
    \text{person}(X) & \leftarrow \text{male}(X) \\
    \text{parent}(X,Y) & \leftarrow \text{father}(X,Y) \\
    \text{grandfather}(Z,X) & \leftarrow \text{father}(Z,Y) \land \text{parent}(Y,X)
    \end{align*}
    \]
Models

- A model of a set of clauses is an interpretation in which all the clauses are true.
  - Start with KB and look at what interpretations can be true.

Example $KB$:

\[ p \leftarrow q. \]
\[ q. \]

<table>
<thead>
<tr>
<th>$\pi(p)$</th>
<th>$\pi(q)$</th>
<th>$\pi(p \leftarrow q)$</th>
<th>Model of KB?</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRUE</td>
<td>TRUE</td>
<td>TRUE</td>
<td>TRUE</td>
</tr>
<tr>
<td>TRUE</td>
<td>TRUE</td>
<td>FALSE</td>
<td>FALSE</td>
</tr>
<tr>
<td>FALSE</td>
<td>TRUE</td>
<td>TRUE</td>
<td>TRUE</td>
</tr>
<tr>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
</tr>
</tbody>
</table>

Example with constants

- Example: (focus on all interpretations, not just intended one)
  + Language with constants $a$ and $b$ and 1-ary predicate $\text{girl}(\_)$.
  + Domain with $D = \{x, y, z\}$.
  + 9 $\phi$’s and 8 $\pi$’s, so 72 interpretations.

- How many models of $KB = \{ \text{girl}(a), \text{girl}(b) \}$?
  (Checking each would take too long, so let's break down into subcases)

  - Case 1: $\phi_i(a) = \phi_i(b)$
    + How many of the 9 $\phi_i$’s have $\phi_i(a) = \phi_i(b)$?
    + When $\phi_i(a) = \phi_i(b) = x$, which $\pi_i$’s make $KB$ true?
    + So how many models with $\phi_i(a) = \phi_i(b)$?

  - Case 2: $\phi_i(a) \neq \phi_i(b)$
    + How many of the 9 $\phi$’s?
    + When $\phi_i(a) = x$ and $\phi_i(b) = y$, which $\pi$’s make the $KB$ true?
    + So how many models with $\phi_i(a) \neq \phi_i(b)$?
Logical Consequence

• 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 $KB$.
  - This tells us that our $KB$, by its definition, always forces $g$ to be true
  - Other terms that mean same thing:
    - $g$ logically follows from $KB$
    - $KB$ entails $g$
• That is, $KB \models g$ if there is no interpretation in which $KB$ is true and $g$ is false.
• $KB \not\models g$ if $g$ is not a logical consequence of $KB$
Example Revisited

- **KB:**
  
  \[ p \leftarrow q. \]
  
  \[ q. \]

<table>
<thead>
<tr>
<th>[ \pi(p) ]</th>
<th>[ \pi(q) ]</th>
<th>[ \pi(p \leftarrow q) ]</th>
<th>model of KB?</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRUE</td>
<td>TRUE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TRUE</td>
<td>FALSE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FALSE</td>
<td>TRUE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FALSE</td>
<td>FALSE</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Does **KB** \( \models p \)?

Overview

- Representation and Reasoning System
- Syntax of Datalog
- Semantics of Datalog
- Adding Variables to Semantics
- Models
- Logical Consequence
  \( \Rightarrow \) Two Views of Semantics
User’s View of Semantics

• Choose a task domain: intended interpretation
• Associate constants with individuals you want to name
• For each relation you want to represent, associate a predicate symbol in the language
• Tell the system clauses that are true in the intended interpretation: axiomatizing the domain
  - hopefully you tell it enough knowledge about the domain so that it can conclude everything you want it to
• Ask questions about your domain

Computer’s view of semantics

• Computer given the knowledge base
  - Computer doesn’t have access to the intended interpretation
• User asks it a question $g$
  - Computer should answer true if $KB \models g$
    + $g$ is true in all models, so is true in user’s intended interpretation
  - Otherwise, computer should answer “I don’t know”
    + There is at least one model in which $g$ is false
    + Note $g$ might have been true in user’s intended interpretation. In this case, user didn’t have enough clauses in the KB to sufficiently narrow down the models
• Aside: computer could answer the question by enumerating over all of the possible interpretations (model checking)
  - But number of interpretations grows quickly!!
Summary of Semantics

• User has intended interpretation
  But just tells the computer a small set of facts that hopefully adequately captures the user’s intended interpretation

• Computer answers true if all interpretations that make KB true (models) make the question true
  - Now we have specs for the computer’s reasoning algorithm
  - It should answer yes if $KB \models q$, other answer don’t know