Lecture Overview

1. Recap

2. Datalog Syntax

3. Datalog Semantics
Top-down definite clause interpreter

To solve the query $\neg q_1 \land \ldots \land \neg q_k$:

$$ac := \text{"yes } \leftarrow \ q_1 \land \ldots \land q_k\text{"}$$

repeat

select atom $a_i$ from the body of $ac$;
choose clause $C$ from $KB$ with $a_i$ as head;
replace $a_i$ in the body of $ac$ by the body of $C$

until $ac$ is an answer.

- Don’t-care nondeterminism If one selection doesn’t lead to a solution, there is no point trying other alternatives. select
- Don’t-know nondeterminism If one choice doesn’t lead to a solution, other choices may. choose
Representational Assumptions of Datalog

bullet An agent’s knowledge can be usefully described in terms of *individuals* and *relations* among individuals.
bullet An agent’s knowledge base consists of *definite* and *positive* statements.
bullet The environment is *static*.
bullet There are only a finite number of individuals of interest in the domain. Each individual can be given a unique name.

⇒ Datalog
Example Domain for an RRS

\[\text{in}(\text{alan}, r123).\]
\[\text{part\_of}(r123, \text{cs\_building}).\]
\[\text{in}(X, Y) \leftarrow\]
\[\quad \text{part\_of}(Z, Y) \land\]
\[\quad \text{in}(X, Z).\]
Lecture Overview

1 Recap

2 Datalog Syntax

3 Datalog Semantics

Logic: Datalog Syntax and Semantics
Syntax of Datalog

**Definition (variable)**

A variable starts with upper-case letter.

**Definition (constant)**

A constant starts with lower-case letter or is a sequence of digits.

**Definition (term)**

A term is either a variable or a constant.

**Definition (predicate symbol)**

A predicate symbol starts with lower-case letter.
Syntax of Datalog (cont)

Definition (atom)
An atomic symbol (atom) is of the form $p$ or $p(t_1, \ldots, t_n)$ where $p$ is a predicate symbol and $t_i$ are terms.

Definition (definite clause)
A definite clause is either an atomic symbol (a fact) or of the form:

$$a \leftarrow b_1 \land \cdots \land b_m$$

where $a$ and $b_i$ are atomic symbols.

Definition (knowledge base)
A knowledge base is a set of definite clauses.
Example Knowledge Base

\[\begin{align*}
\text{in}(alan, R) & \leftarrow \\
& \text{teaches}(alan, \text{cs322}) \land \\
& \text{in}(\text{cs322}, R).
\end{align*}\]

\[\begin{align*}
\text{grandfather}(\text{william}, X) & \leftarrow \\
& \text{father}(\text{william}, Y) \land \\
& \text{parent}(Y, X).
\end{align*}\]

\[\begin{align*}
\text{slithy}(\text{toves}) & \leftarrow \\
& \text{mimsy} \land \text{borogroves} \land \\
& \text{outgrabe}(\text{mome}, \text{Raths}).
\end{align*}\]
Lecture Overview

1 Recap

2 Datalog Syntax

3 Datalog Semantics
Recall: a semantics specifies the meaning of sentences in the language.
- ultimately, we want to be able to talk about which sentences are true and which are false

In propositional logic, all we needed to do in order to come up with an interpretation was to assign truth values to atoms

For Datalog, an interpretation specifies:
- what objects (individuals) are in the world
- the correspondence between symbols in the computer and objects & relations in world
  - which constants denote which individuals
  - which predicate symbols denote which relations (and thus, along with the above, which sentences will be true and which will be false)
**Definition (interpretation)**

An **interpretation** is a triple $I = \langle D, \phi, \pi \rangle$, where

- $D$, the **domain**, is a nonempty set. Elements of $D$ are individuals.
- $\phi$ is a mapping that assigns to each constant an element of $D$. Constant $c$ denotes individual $\phi(c)$.
- $\pi$ is a mapping that assigns to each $n$-ary predicate symbol a relation: a function from $D^n$ into \{TRUE, FALSE\}. 
Example Interpretation

Constants: phone, pencil, telephone.
Predicate Symbol: noisy (unary), left_of (binary).

- $D = \{\text{phone}, \text{pencil}, \text{telephone}\}$.
- These are actually objects in the world, not symbols
- $\phi(\text{phone}) = \text{phone}, \phi(\text{pencil}) = \text{pencil}, \phi(\text{telephone}) = \text{telephone}$.

$\pi(\text{noisy}):$  

| $\langle \text{phone}, \text{phone} \rangle$ | FALSE | $\langle \text{pencil}, \text{pencil} \rangle$ | TRUE | $\langle \text{telephone}, \text{telephone} \rangle$ | FALSE |
| $\pi(\text{left_of}):$ |
| $\langle \text{phone}, \text{pencil} \rangle$ | FALSE | $\langle \text{pencil}, \text{telephone} \rangle$ | FALSE | $\langle \text{telephone}, \text{phone} \rangle$ | TRUE |
| $\langle \text{pencil}, \text{phone} \rangle$ | FALSE | $\langle \text{telephone}, \text{pencil} \rangle$ | FALSE | $\langle \text{phone}, \text{pencil} \rangle$ | TRUE |
| $\langle \text{pencil}, \text{pencil} \rangle$ | FALSE | $\langle \text{telephone}, \text{telephone} \rangle$ | FALSE | $\langle \text{telephone}, \text{pencil} \rangle$ | FALSE |
Important points to note

- The domain $D$ can contain **real objects**. (e.g., a person, a room, a course). $D$ can’t necessarily be stored in a computer.
- The constants do not have to match up one-to-one with members of the domain. Multiple constants can refer to the **same object**, and some objects can have **no constants** that refer to them.
- $\pi(p)$ specifies whether the relation denoted by the $n$-ary predicate symbol $p$ is true or false for each $n$-tuple of individuals.
- If predicate symbol $p$ has **no arguments**, then $\pi(p)$ is either **TRUE** or **FALSE**.
  - this was the situation in propositional logic
# Truth in an interpretation

**Definition (truth in an interpretation)**

- A constant $c$ denotes in $I$ the individual $\phi(c)$.

- Ground (variable-free) atom $p(t_1, \ldots, t_n)$ is
  - *true in interpretation* $I$ if $\pi(p)(t'_1, \ldots, t'_n) = TRUE$, where $t_i$ denotes $t'_i$ in interpretation $I$ and
  - *false in interpretation* $I$ if $\pi(p)(t'_1, \ldots, t'_n) = FALSE$.

- Ground clause $h \leftarrow b_1 \land \ldots \land b_m$ is
  - *false in interpretation* $I$ if $h$ is *false in* $I$ and each $b_i$ is *true in* $I$, and item *true in interpretation* $I$ otherwise.
  - A knowledge base, $KB$, is *true in* interpretation $I$ if and only if every clause in $KB$ is *true in* $I$.

- Notice that truth values are only associated with *predicates* (atomic symbols; clauses), not variables and constants!
Example Truths

In the interpretation given before:

\[ \text{noisy}(\text{phone}) \]
Example Truths

In the interpretation given before:

\[
\text{noisy}(\text{phone}) \\
\text{noisy}(\text{telephone})
\]

\[\text{true}\]
Example Truths

In the interpretation given before:

\[
\begin{align*}
\text{noisy(\text{phone})} & \quad \text{true} \\
\text{noisy(\text{telephone})} & \quad \text{true} \\
\text{noisy(\text{pencil})} & \quad \text{false}
\end{align*}
\]
Example Truths

In the interpretation given before:

\[ \text{noisy(phone)} \quad \text{true} \]
\[ \text{noisy(telephone)} \quad \text{true} \]
\[ \text{noisy(pencil)} \quad \text{false} \]
\[ \text{left_of(phone, pencil)} \]
Example Truths

In the interpretation given before:

\[\text{noisy(phone)} \quad \text{true}\]
\[\text{noisy(telephone)} \quad \text{true}\]
\[\text{noisy(pencil)} \quad \text{false}\]
\[\text{left_of(phone, pencil)} \quad \text{true}\]
\[\text{left_of(phone, telephone)} \quad \text{true}\]
Example Truths

In the interpretation given before:

\[
\begin{align*}
\text{noisy(phone)} & \quad \text{true} \\
\text{noisy(telephone)} & \quad \text{true} \\
\text{noisy(pencil)} & \quad \text{false} \\
\text{left_of(phone, pencil)} & \quad \text{true} \\
\text{left_of(phone, telephone)} & \quad \text{false} \\
\text{noisy(pencil)} & \leftarrow \text{left_of(phone, telephone)}
\end{align*}
\]
Example Truths

In the interpretation given before:

\[
\begin{align*}
\textit{noisy(phone)} & : \text{true} \\
\textit{noisy(telephone)} & : \text{true} \\
\textit{noisy(pencil)} & : \text{false} \\
\textit{left_of(phone, pencil)} & : \text{true} \\
\textit{left_of(phone, telephone)} & : \text{false} \\
\textit{noisy(pencil)} & \leftarrow \textit{left_of(phone, telephone)} : \text{true} \\
\textit{noisy(pencil)} & \leftarrow \textit{left_of(phone, pencil)} : \text{false}
\end{align*}
\]
Example Truths

In the interpretation given before:

\[
\begin{align*}
\text{noisy}(\text{phone}) & \quad \text{true} \\
\text{noisy}(\text{telephone}) & \quad \text{true} \\
\text{noisy}(\text{pencil}) & \quad \text{false} \\
\text{left}_{\text{of}}(\text{phone}, \text{pencil}) & \quad \text{true} \\
\text{left}_{\text{of}}(\text{phone}, \text{telephone}) & \quad \text{false} \\
\text{noisy}(\text{pencil}) & \leftarrow \text{left}_{\text{of}}(\text{phone}, \text{telephone}) \quad \text{true} \\
\text{noisy}(\text{pencil}) & \leftarrow \text{left}_{\text{of}}(\text{phone}, \text{pencil}) \quad \text{false} \\
\text{noisy}(\text{phone}) & \leftarrow \text{noisy}(\text{telephone}) \land \text{noisy}(\text{pencil}) \quad \text{true}
\end{align*}
\]
Example Truths

In the interpretation given before:

- noisy(phone)  
- noisy(telephone)  
- noisy(pencil)  
- left_of(phone, pencil)  
- left_of(phone, telephone)  
- noisy(pencil) ← left_of(phone, telephone)  
- noisy(pencil) ← left_of(phone, pencil)  
- noisy(phone) ← noisy(telephone) ∧ noisy(pencil)
How do we determine the truth value of a clause that includes variables?

**Definition (variable assignment)**

A variable assignment is a function from variables into the domain.

- Given an interpretation and a variable assignment, each term denotes an individual and each clause is either true or false.
- A clause containing variables is true in an interpretation if it is true for all variable assignments.
  - Variables are universally quantified in the scope of a clause.
Models and logical consequences

**Definition (model)**

A **model** of a set of clauses is an interpretation in which all the clauses are *true*.

**Definition (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 $KB$.

- That is, $KB \models g$ if there is no interpretation in which $KB$ is *true* and $g$ is *false*. 