Example

- KB: female(sally)
  - person(X) ← female(X)

- Prove KB | = person(sally)

What does KB | = person(sally) mean?

- Means that if interpretation I models KB then it models person(sally)

- Could prove this by checking all interpretations

- Let's do proof by contradiction instead

  - Assume that I is a model of KB but not a model of person(sally)

- Then KB | does not = person(sally) (contradiction)

- Prove KB | = person(sally) (if KB | = person(sally) then I is a model)

- Then interpretation I models every clause in KB

- KB | = true

- Model I of KB | is an interpretation

- If I is a complete mapping

- Then interpretation maps any clause to either true or false

Review

- Bottom-up Ground Proof Procedure
- Top-down Ground Proof Procedure
- Proof Procedures
- Queries
- Semantics

Overview
Overview

• Semantics ⇒ Queries
• Proof Procedures
  • Bottom-up Ground Proof Procedure
  • Top-down Ground Proof Procedure

More on Variables in Clauses (pg. 42)

• Say \( \text{parent}(X,Y) \leftarrow \text{father}(X,Y) \) is in KB
  - Implicit universal quantifiers around it
  - Anytime that \( \text{father}(X,Y) \) is true, so must \( \text{parent}(X,Y) \)

• Say \( \text{grandfather}(X,Y) \leftarrow \text{father}(X,Z) \land \text{parent}(Z,Y) \) is in KB
  - This clause is true for all \( X, Y, Z \)
  - \( \forall X \, Y \, Z \, (\text{grandfather}(X,Y) \leftarrow \text{father}(X,Z) \land \text{parent}(Z,Y)) \).

Proof by Contradiction: A Semantic Proof

• Assume \( I = \{ D, \phi, \pi \} \) is a model of \( \text{KB} = \{ \text{female}(\text{sally}) \land \text{person}(X) \leftarrow \text{female}(X) \} \)
  - So \( <\phi>(\text{sally}) \in \pi(\text{female}) \)
  - Say \( \phi(\text{sally}) = F \), so \( <F> \in \pi(\text{female}) \) (1)
  - \( \text{person}(X) \leftarrow \text{female}(X) \) must be true for \( I \rho \) for any var. assignment \( \rho \) (2)
  - Assume variable assignment \( \rho \) where \( \rho(X) = F \)

  - If \( \text{female}(X) \) is true for \( I \rho \), then \( \text{person}(X) \) must be true for \( I \rho \) (from (2)) (4)
  - \( \rho(X) = F \) so \( <F> \in \pi(\text{female}) \) and \( \phi(\text{sally}) = F \)
  - \( <F> \in \pi(\text{female}) \) and \( \phi(\text{sally}) = F \) are both true for \( I \rho \) (5)
  - \( \text{person}(X) \leftarrow \text{female}(X) \) is in KB

  - Consider variable assignment \( \delta \) where \( \delta(X) = F \)

  - Assume \( \text{person}(\text{Sally}) \) is not true under \( I \)

  - \( \text{person}(\text{Sally}) \) is not true under \( I \)

  - \( \text{person}(\text{Sally}) \) must be false for any var. assignment \( \rho \) (7)

  - Assume \( \{ \phi \} = \text{a model of KB} \)

  - \( \text{Assume by contradiction: A Semantic Proof} \)
Overview

Top-down Ground Proof Procedure
- Bottom-up Ground Proof Procedure

Queries
- Proof Procedures

Semantics

Queries with Variables
- You might not only want to check if something is true or false, but what value makes it true
- KB:
  father(william,ted)

- Example:
  parent(X,Y) ← father(X,Y)

  - Question:
    parent(X,ted) - Who is Ted's parent?

  - Solution:
    yes(X) ← parent(X,ted)

  - An answer is either
    - instance of 'yes' that is a logical consequence of KB
    - no if no instance is a logical consequence of KB

  - We do not distinguish between it being false in all models or just some

Ground Queries
- A query is a way to ask if a body is a logical consequence of the KB:
  ? b₁ ∧ ... ∧ bₘ

- Ground query (no variables) has the answer
  - "yes" if the body is a logical consequence of the KB
  - "no" if the body is not a logical consequence of the KB

- Can do query-answering by:
  - Transform goal into a query (KB)
  - Add (temporarily)
  - Check if 'yes' is a logical consequence of KB

- Ground Queries (no variables) has the answer:
  - "yes" if KB entailment
Two Types of Proof Procedures

- **Top-Down (Forward-Chaining)**
  - KB ⊢ g means g can be derived from KB with the proof procedure
  - Properties of Proof Procedure:
    - Soundness: if KB ⊢ g then KB |= g
    - Completeness: if KB |= g then KB ⊢ g

- **Bottom-Up (Backward-Chaining)**
  - Query ⊢ KB
  - Properties of Proof Procedure:
    - Soundness: if KB ⊢ g then KB |= g
    - Completeness: if KB |= g then KB ⊢ g

Proof Procedures

Semantics Is Not Enough

- We have KB
- But, don't have a mechanical way of checking if KB |= g
- Can, in fact check the user's intended information
- Can extract this so user can ask questions with variables as well
- Can extract this so user can ask questions with variables as well
- Can check this so user can ask questions with variables as well

Terminology:

- Semantic proof: |=, logically follows, logically entails, models
- Syntactic proof: ⊢, derives
Non-deterministic Specification

- Haven't specified the exact order that things should be done in KB
- What order should we pick clauses from KB to try?

Bottom-up Ground Proof Procedure

- For now, only consider ground facts and ground rules
- Bottom-up or forward chaining procedure: starts from KB and works towards the query
- Forward chaining rule:
  - If \( h \leftarrow b_1 \land \ldots \land b_m \) is a clause in the KB
  - and each \( b_i \) has been derived
  - then \( h \) can be derived

General forward chaining rule:

- For now, only consider ground facts and ground rules
- Top-down Ground Proof Procedure

For now, only consider ground facts and ground rules
Is it Complete?

• Does C have every ground atom that logically follows from KB?

We need to prove something about consequence sets

Let C be the final consequent set generated by the algorithm
-

- Will stop because finite number of constants and predicate symbols
- Will stop with same C, no matter what order C was generated

Define I such that for atom h -

- I(h) is true if h ∈ C
- Otherwise, I(h) is false

I is an interpretation because it defines a subset of ground atoms as being true, and the rest as false

Is it Sound?

• Does everything in C logically follow from KB?

Proof by contradiction: assume KB ⊢ g but KB ̸|= g
-

- g is the result of a finite number of derivations
-

- Without loss of generality, assume b is first one in derivation such that KB ̸|= b

Proof by contradiction: assume KB ⊢ b but KB ̸|= b
-

- b is the result of a finite number of derivations
-

- b ̸|= KB

Does every ground atom that logically follows from KB? Does C have every ground atom that logically follows from KB?

Example

\[ \begin{align*}
\text{a} & \leftarrow \text{b} \land \text{c}. \\
\text{b} & \leftarrow \text{d} \land \text{e}. \\
\text{b} & \leftarrow \text{g} \land \text{e}. \\
\text{c} & \leftarrow \text{e}. \\
\text{d}. \\
\text{e}. 
\end{align*} \]
Overview

• Semantics
• Queries
• Proof Procedures

⇒ Bottom-up Ground Proof Procedure

Final Step in Completeness Proof

• Let \( g \) be atomic and \( KB \models g \)
• Need to make sure that \( KB \vdash g \)
• Since \( KB \models g \), \( g \) must be in every model of \( KB \)
• So, it is in the interpretation defined by the Consequence set
• Since \( g \) is atomic and it is true in the interpretation, it must be in consequence set
• So \( KB \vdash g \)

Proof that Consequence Set is a Model

• Proof by Contradiction: Let \( g \in KB \) but where \( I(g) \) is false
• Since \( g \in KB \), \( g \) must have the form \( h \leftarrow b_1 \land \ldots \land b_m \)
• So \( h \leftarrow b_1 \land \ldots \land b_m \) is false in \( I \)
• Since \( b_1 \land \ldots \land b_m \) is true in \( I \)
• Each individually must be true in \( I \)
• Remember, definition of \( \land \) comes from Datalog, not \( I \)
• So, all \( b_i \) in \( C \) and \( h \leftarrow b_1 \land \ldots \land b_m \) is in \( KB \)
• Since all \( b_i \) in \( C \) and \( h \leftarrow b_1 \land \ldots \land b_m \) is in \( KB \)
• \( h \leftarrow b_1 \land \ldots \land b_m \) is true in \( I \)

Proof by Contradiction: Let \( g \in KB \) where \( I(g) \) is false
Example

\[ a \rightarrow f \]
\[ \neg c \rightarrow \neg e \]
\[ p \rightarrow c \]
\[ g \rightarrow q \]
\[ \neg p \rightarrow q \]
\[ \neg q / q \rightarrow a \]

KB

Now for some definitions

- **Answer clause**: Yes \( \leftarrow a_1 \land \ldots \land a_m \)
- **Answer**: An answer clause with \( m = 0 \)
- **Derivation of a query** \( \chi_1 \land \ldots \land \chi_k \) from KB is a sequence of answer clauses \( \gamma_0, \gamma_1, \ldots, \gamma_n \) such that
  - \( \gamma_0 \) is the answer clause corresponding to the original query
  - \( \gamma_i \) is obtained by resolving \( \gamma_{i-1} \) with a clause in KB
  - \( \gamma_n \) is the answer

Nondeterminism - In choosing which clause from KB to resolve with,
- Can find all derivations by systematically considering all different choices (see Chapter 4)

Top-Down Ground Proof Procedure

- **Alternative to bottom-up (forward-chaining)**
- **Top-down (backward-chaining)**:
  - Start with goal, work toward facts in KB
  - Define Clause Resolution for Ground Case
  - Top-down (backwards-chaining)
  - Alternative to Bottom-up (forward-chaining)
Bottom-up versus Top-down

- Both top-down and bottom-up techniques are sufficient for datalog.
- We will explore some of these later in the course.
- There are many other ways of doing proofs.

Some top-down proofs may be easier to follow, while bottom-up proofs may be more efficient.

- Any top-down proof can be converted to a bottom-up proof.
- Any bottom-up proof can be converted to a top-down proof.

**Bottom-Up**

```
C = \{ e, c, d, b, a \}
C = \{ e, c, d \}
C = \{ e, c \}
C = \{ e \}
```

**Top-Down**

```
\{ a \} \models \top
\{ c, e \} \models \top
\{ e, c, d \} \models \top
\{ e, c, d, b, a \} \models \top
```

**KB Rule**

- Bottom-Up
- Top-Down

K.B.