Propositional logic

Logical inference problem

Logical inference problem:
• Given:
  – a knowledge base KB (a set of sentences) and
  – a sentence $\alpha$ (called a theorem),
• Does a KB semantically entail $\alpha$? $KB \models \alpha$?

In other words:
• In all interpretations in which sentences in the KB are true, is $\alpha$ also true?
Logical inference problem

**Logical inference problem:**
- **Given:**
  - a knowledge base KB (a set of sentences) and
  - a sentence \( \alpha \) (called a theorem),
- **Does a KB semantically entail \( \alpha \) ?** \( KB \models \alpha \)

**Approaches to solve the logical inference problem:**
- Truth-table approach
- Inference rules
- Conversion to SAT
  - Resolution refutation

**Properties of inference solutions**
- **Truth-table approach**
  - Blind
  - Exponential in the number of variables
- **Inference rules**
  - More efficient
  - Many inference rules to cover logic
- **Conversion to SAT - Resolution refutation**
  - More efficient
  - Sentences must be converted into CNF
  - One rule – the resolution rule - is sufficient to perform all inferences
KB in restricted forms

If the sentences in the KB are restricted to some special forms, some of the sound inference rules may become complete.

Example:

- **Horn form (Horn normal form)**
  - a clause with **at most one positive literal**
    \[(A \lor \neg B) \land (\neg A \lor \neg C \lor D)\]

  Can be written also as:
  \[(B \Rightarrow A) \land ((A \land C) \Rightarrow D)\]

- **Two inference rules that are sound and complete for KBs in the Horn normal form:**
  - Resolution
  - Modus ponens

KB in Horn form

- **Horn form**: a clause with **at most one positive literal**
  \[(A \lor \neg B) \land (\neg A \lor \neg C \lor D)\]

- **Not all sentences in propositional logic can be converted into the Horn form**

- **KB in Horn normal form**:
  - Two types of propositional statements:
    - **Rules**
      \[
      \neg B_1 \lor \neg B_2 \lor \ldots \lor \neg B_k \lor A
      \equiv
      \neg (B_1 \land B_2 \land \ldots \land B_k) \lor A
      \equiv
      (B_1 \land B_2 \land \ldots \land B_k \Rightarrow A)
      \]
    - Propositional symbols: **facts**
      \[B\]
KB in Horn form

• Application of the resolution rule:
  – Infers new facts from previous facts
    
    \[
    \frac{A \lor \neg B, B}{A} \quad \frac{A \lor \neg B, \neg (B \lor \neg C)}{A \lor C}
    \]

  – Resolution is **sound and complete** for inferences on propositional symbols for KB in the Horn normal form (clausal form)

• Similarly, **modus ponens is sound and complete** when the HNF is written in the implicative form

Complexity of inferences for KBs in HNF

**Question:**
How efficient the inferences in the HNF can be?

**Answer:**
Inference on propositional symbols →
Procedures linear in the size of the KB in the Horn form exist.

• Size of a clause: the number of literals it contains.
• Size of the KB in the HNF: the sum of the sizes of its elements.

**Example:**

\[A, B, (A \land B \Rightarrow C), (C \Rightarrow D), (C \Rightarrow E), (E \land F \Rightarrow G)\]
or
\[A, B, (\neg A \lor \neg B \lor C), (\neg C \lor D), (\neg C \lor E), (\neg E \lor \neg F \lor G)\]

The size is: 12
Complexity of inferences for KBs in HNF

How to do the inference? If the HNF (is in the clausal form) we can apply resolution.

\[ A, B, (\neg A \lor \neg B \lor C), (\neg C \lor D), (\neg C \lor E), (\neg E \lor \neg F \lor G) \]

Inferred facts
Complexity of inferences for KBs in HNF

Features:
- Every resolution is a **positive unit resolution**; that is, a resolution in which **one clause is a positive unit clause** (i.e., a proposition symbol).

\[ A, B, (\neg A \lor \neg B \lor C), (\neg C \lor D), (\neg C \lor E), (\neg E \lor \neg F \lor G) \]

- At each resolution, the input clause which is not a unit clause is a logical consequence of the result of the resolution. (Thus, the input clause may be deleted upon completion of the resolution operation.)

\[ A, B, (\neg A \lor \neg B \lor C), (\neg C \lor D), (\neg C \lor E), (\neg E \lor \neg F \lor G) \]
Complexity of inferences for KBs in HNF

Features:
- At each resolution, the input clause which is not a unit clause is a logical consequence of the result of the resolution. (Thus, the input clause may be deleted upon completion of the resolution operation.)

\[
A, B, (\neg A \lor \neg B \lor C), (\neg C \lor D), (\neg C \lor E), (\neg E \lor \neg F \lor G)
\]

- Following this deletion, the size of the KB (the sum of the lengths of the remaining clauses) is one less than it was before the operation.

\[
A, B, (\neg A \lor \neg B \lor C), (\neg C \lor D), (\neg C \lor E), (\neg E \lor \neg F \lor G)
\]
Complexity of inferences for KBs in HNF

Features:
- If \( n \) is the size of the KB, then at most \( n \) positive unit resolutions may be performed on it.

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

A linear time resolution algorithm:
- The number of positive unit resolutions is limited to the size of the formula (\( n \))
- But to assure overall linear time we need to access each proposition in a constant time:
  - Data structures indexed by proposition names may be accessed in constant time. (This is possible if the proposition names are number in a range (e.g., 1..n), so that array lookup is the access operation.
  - If propositions are accessed by name, then a symbol table is necessary, and the algorithm will run in time \( O(n \cdot \log(n)) \).
Forward and backward chaining

Two inference procedures based on modus ponens for Horn KBs:

• **Forward chaining**
  **Idea:** Whenever the premises of a rule are satisfied, infer the conclusion. Continue with rules that became satisfied.

• **Backward chaining (goal reduction)**
  **Idea:** To prove the fact that appears in the conclusion of a rule prove the premises of the rule. Continue recursively.

Both procedures are complete for KBs in the Horn form !!!

---

Forward chaining example

• **Forward chaining**
  **Idea:** Whenever the premises of a rule are satisfied, infer the conclusion. Continue with rules that became satisfied.

Assume the KB with the following rules and facts:

**KB:**

R1: \( A \land B \Rightarrow C \)
R2: \( C \land D \Rightarrow E \)
R3: \( C \land F \Rightarrow G \)

**Facts:**

F1: \( A \)
F2: \( B \)
F3: \( D \)

**Theorem:** \( E \) ??
Forward chaining example

Theorem: $E$

KB:  
R1: $A \land B \Rightarrow C$
R2: $C \land D \Rightarrow E$
R3: $C \land F \Rightarrow G$

F1: $A$
F2: $B$
F3: $D$

Rule R1 is satisfied.

F4: $C$
**Forward chaining example**

**Theorem:** $E$

**KB:**

- **R1:** $A \land B \Rightarrow C$
- **R2:** $C \land D \Rightarrow E$
- **R3:** $C \land F \Rightarrow G$

<table>
<thead>
<tr>
<th>F1</th>
<th>F2</th>
<th>F3</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A$</td>
<td>$B$</td>
<td>$D$</td>
</tr>
</tbody>
</table>

- **Rule R1 is satisfied.**
- **F4:** $C$
- **Rule R2 is satisfied.**
- **F5:** $E$

---

**Forward chaining**

- Efficient implementation: linear in the size of the KB
- **Example:**

| $P \Rightarrow Q$ |
| $L \land M \Rightarrow P$ |
| $B \land L \Rightarrow M$ |
| $A \land P \Rightarrow L$ |
| $A \land B \Rightarrow L$ |
| $A$ |
| $B$ |
Forward chaining

- Count the number of facts in the antecedent of the rule

\[ P \Rightarrow Q \]
\[ L \land M \Rightarrow P \]
\[ B \land L \Rightarrow M \]
\[ A \land P \Rightarrow L \]
\[ A \land B \Rightarrow L \]
\[ A \]
\[ B \]

\( Q \)
\( P \)
\( M \)
\( L \)
\( A \)
\( B \)

Agenda (facts)

Inferred facts decrease the count

\[ P \Rightarrow Q \]
\[ L \land M \Rightarrow P \]
\[ B \land L \Rightarrow M \]
\[ A \land P \Rightarrow L \]
\[ A \land B \Rightarrow L \]
\[ A \]
\[ B \]

inferred
Forward chaining

- New facts can be inferred when the count associated with a rule becomes 0

\[ P \Rightarrow Q \]
\[ L \land M \Rightarrow P \]
\[ B \land L \Rightarrow M \]
\[ A \land P \Rightarrow L \]
\[ A \land B \Rightarrow L \]
\[ A \]
\[ B \]
Forward chaining

•

$P \Rightarrow Q$
$L \land M \Rightarrow P$
$B \land L \Rightarrow M$
$A \land P \Rightarrow L$
$A \land B \Rightarrow L$
$A$
$B$

Forward chaining

•

$P \Rightarrow Q$
$L \land M \Rightarrow P$
$B \land L \Rightarrow M$
$A \land P \Rightarrow L$
$A \land B \Rightarrow L$
$A$
$B$
Forward chaining

- $P \Rightarrow Q$
- $L \land M \Rightarrow P$
- $B \land L \Rightarrow M$
- $A \land P \Rightarrow L$
- $A \land B \Rightarrow L$
- $A$
- $B$

Backward chaining example

- Backward chaining is more focused:
  - tries to prove the theorem only

KB:  R1: $A \land B \Rightarrow C$
     R2: $C \land D \Rightarrow E$
     R3: $C \land F \Rightarrow G$
F1:  $A$
F2:  $B$
F3:  $D$
Backward chaining example

KB:
- R1: $A \land B \Rightarrow C$
- R2: $C \land D \Rightarrow E$
- R3: $C \land F \Rightarrow G$
- F1: $A$
- F2: $B$
- F3: $D$

- Backward chaining is more focused:
  - tries to prove the theorem only

Backward chaining

- $P \Rightarrow Q$
- $L \land M \Rightarrow P$
- $B \land L \Rightarrow M$
- $A \land P \Rightarrow L$
- $A \land B \Rightarrow L$
- $A$
- $B$
Backward chaining

\[ P \Rightarrow Q \]
\[ L \land M \Rightarrow P \]
\[ B \land L \Rightarrow M \]
\[ A \land P \Rightarrow L \]
\[ A \land B \Rightarrow L \]

\[ A \]
\[ B \]
Backward chaining

\[
P \Rightarrow Q
\]

\[
L \land M \Rightarrow P
\]

\[
B \land L \Rightarrow M
\]

\[
A \land P \Rightarrow L
\]

\[
A \land B \Rightarrow L
\]

\[
A
\]

\[
B
\]
Backward chaining

\[ P \Rightarrow Q \]
\[ L \land M \Rightarrow P \]
\[ B \land L \Rightarrow M \]
\[ A \land P \Rightarrow L \]
\[ A \land B \Rightarrow L \]
\[ A \]
\[ B \]
Backward chaining

\[ P \Rightarrow Q \quad \leftarrow \]
\[ L \land M \Rightarrow P \quad \leftarrow \]
\[ B \land L \Rightarrow M \quad \leftarrow \]
\[ A \land P \Rightarrow L \quad \leftarrow \]
\[ A \land B \Rightarrow L \quad \leftarrow \]
\[ A \quad \leftarrow \]
\[ B \quad \leftarrow \]
Backward chaining

- $P \rightarrow Q$
- $L \land M \Rightarrow P$
- $B \land L \Rightarrow M$
- $A \land P \Rightarrow L$
- $A \land B \Rightarrow L$
- $A$
- $B$

Forward vs Backward chaining

- **FC is data-driven**, automatic, unconscious processing,
  - e.g., object recognition, routine decisions
- May do lots of work that is irrelevant to the goal
- **BC is goal-driven**, appropriate for problem-solving,
  - e.g., Where are my keys? How do I get into a PhD program?
- Complexity of BC can be much less than **linear in size of KB**
KB agents based on propositional logic

- Propositional logic allows us to build **knowledge-based agents** capable of answering queries about the world by inferring new facts from the known ones
- **Example:** an agent for diagnosis of a bacterial disease

**Facts:**
- The stain of the organism is gram-positive
- The growth conformation of the organism is chains

**Rules:**
- (If) The stain of the organism is gram-positive \(\land\) The morphology of the organism is coccus \(\land\) The growth conformation of the organism is chains
- (Then) \(\Rightarrow\) The identity of the organism is streptococcus

---

First-order logic
Limitations of propositional logic

World we want to represent and reason about consists of a number of objects with variety of properties and relations among them.

**Propositional logic:**
- Represents statements about the world without reflecting this structure and without modeling these entities explicitly.

**Consequence:**
- Some knowledge is hard or impossible to encode in the propositional logic.
- Two cases that are hard to represent:
  - Statements about similar objects, or relations
  - Statements referring to groups of objects.

- Statements about similar objects and relations needs to be enumerated
- Example: Seniority of people domain
  For inferences we need:
  
  \[
  \begin{align*}
  \text{John is older than Mary} & \land \text{Mary is older than Paul} \\
  \Rightarrow & \text{John is older than Paul} \\
  \text{Jane is older than Mary} & \land \text{Mary is older than Paul} \\
  \Rightarrow & \text{Jane is older than Paul}
  \end{align*}
  \]

- Problem: if we have many people and their age relations we need to represent many rules to support the inferences
- Possible solution: ??
Limitations of propositional logic

• **Statements about similar objects and relations needs to be enumerated**

• **Example:** Seniority of people domain
  For inferences we need:
  \[ \text{John is older than Mary} \land \text{Mary is older than Paul} \]
  \[ \Rightarrow \text{John is older than Paul} \]
  \[ \text{Jane is older than Mary} \land \text{Mary is older than Paul} \]
  \[ \Rightarrow \text{Jane is older than Paul} \]

• **Problem:** if we have many people and their age relations we need to represent many rules to support the inferences

• **Possible solution:** introduce variables

\[
\begin{align*}
\text{Pers}_A & \text{ is older than } \text{Pers}_B & \text{Pers}_B & \text{ is older than } \text{Pers}_C \\
\Rightarrow & \text{Pers}_A & \text{ is older than } \text{Pers}_C
\end{align*}
\]

Limitations of propositional logic

• **Statements referring to groups of objects require exhaustive enumeration of objects**

• **Example:**
  Assume we want to express \( \text{Every student likes vacation} \)

  Doing this in propositional logic would require to include statements about every student

  \[ \text{John likes vacation} \land \text{Mary likes vacation} \land \text{Ann likes vacation} \land \ldots \]

• **Solution:** Allow quantification in statements
First-order logic (FOL)

- More expressive than propositional logic

- **Eliminates deficiencies of PL by:**
  - Representing objects, their properties, relations and statements about them;
  - Introducing variables that refer to an arbitrary objects and can be substituted by a specific object
  - Introducing quantifiers allowing us to make statements over groups objects without the need to represent each of them separately

Logic

Logic is defined by:

- **A set of sentences**
  - A sentence is constructed from a set of primitives according to syntax rules.

- **A set of interpretations**
  - An interpretation gives a semantic to primitives. It associates primitives with objects, values in the real world.

- **The valuation (meaning) function** $V$
  - Assigns a truth value to a given sentence under some interpretation

  $V : \text{sentence} \times \text{interpretation} \rightarrow \{\text{True}, \text{False}\}$
First-order logic. Syntax.

**Term** – a syntactic entity for representing objects

**Terms in FOL:**
- **Constant symbols:** represent specific objects
  - E.g. *John, France, car89*
- **Variables:** represent objects of a certain type (type = domain of discourse)
  - E.g. *x, y, z*
- **Functions** applied to one or more terms
  - E.g. *father-of*(John)
    
**First order logic. Syntax.**

**Sentences in FOL:**
- **Atomic sentences:**
  - A **predicate symbol** applied to 0 or more terms
    
**Examples:**

- *Red(car12),*
- *Sister(Amy, Jane);*
- *Manager(father-of(John));*

- t1 = t2 **equivalence** of terms

**Example:**

- *John = father-of(Peter)*
First order logic. Syntax.

Sentences in FOL:
- **Complex sentences:**
- Assume $\phi, \psi$ are sentences in FOL. Then:
  - $(\phi \land \psi)$, $(\phi \lor \psi)$, $(\phi \Rightarrow \psi)$, $(\phi \Leftrightarrow \psi)$, $\neg \psi$
  and
- $\forall x \phi$ and $\exists y \phi$
  are sentences

Symbols $\exists, \forall$
- stand for the **existential** and the **universal** quantifier

Semantics. Interpretation.

An interpretation $I$ is defined by a **mapping** constants, predicates and functions to the **domain of discourse** $D$ or **relations on $D$**
- **domain of discourse:** a set of objects in the world we represent and refer to;

**An interpretation $I$ maps:**
- Constant symbols to objects in $D$
  $I(\text{John}) = \text{John}$
- Predicate symbols to relations, properties on $D$
  $I(\text{brother}) = \left\{ \langle \text{Juan} \enspace \text{Pedro} \rangle ; \langle \text{Pedro} \enspace \text{Juan} \rangle ; \ldots \right\}$
- Function symbols to functional relations on $D$
  $I(\text{father-of}) = \left\{ \langle \text{Juan} \rightarrow \text{Pedro} \rangle ; \langle \text{Pedro} \rightarrow \text{Juan} \rangle ; \ldots \right\}$
Semantics of sentences.

Meaning (evaluation) function:

\[
V : \text{sentence} \times \text{interpretation} \rightarrow \{True, False\}
\]

A **predicate** \(\text{predicate}(\text{term-1, term-2, term-3, term-n})\) is true for the interpretation \(I\), iff the objects referred to by \(\text{term-1, term-2, term-3, term-n}\) are in the relation referred to by \(\text{predicate}\)

\[
\begin{align*}
I(\text{John}) &= \text{\textbullet} \\
I(\text{Paul}) &= \text{\textbullet} \\
I(\text{brother}) &= \{\text{\textbullet; \textbullet; \textbullet} \ }; \text{ in } I(\text{brother}) \\
\text{brother(John, Paul)} &= \text{\textbullet} \quad \text{in } I(\text{brother}) \\
V(\text{brother(John, Paul), I}) &= \text{True}
\end{align*}
\]