Propositional and First-Order Logic

Chapter 7.4—7.8, 8.1—8.3, 8.5

Some material adopted from notes by Andreas Geyer-Schulz and Chuck Dyer
Logic roadmap overview

• Propositional logic (review)
• Problems with propositional logic
• First-order logic (review)
  – Properties, relations, functions, quantifiers, …
  – Terms, sentences, wffs, axioms, theories, proofs, …
• Extensions to first-order logic
• Logical agents
  – Reflex agents
  – Representing change: situation calculus, frame problem
  – Preferences on actions
  – Goal-based agents
Disclaimer

“Logic, like whiskey, loses its beneficial effect when taken in too large quantities.”

- Lord Dunsany
Propositional Logic: Review
Big Ideas

• Logic is a great knowledge representation language for many AI problems

• **Propositional logic** is the simple foundation and fine for some AI problems

• **First order logic** (FOL) is much more expressive as a KR language and more commonly used in AI

• There are many variations: horn logic, higher order logic, three-valued logic, probabilistic logics, etc.
Propositional logic

- Logical constants: true, false
- Propositional symbols: P, Q,... (aka atomic sentences)
- Wrapping parentheses: ( … )
- Sentences are combined by connectives:
  - \( \land \) and       [conjunction]
  - \( \lor \) or        [disjunction]
  - \( \Rightarrow \) implies [implication / conditional]
  - \( \Leftrightarrow \) is equivalent [biconditional]
  - \( \neg \) not       [negation]
- Literal: atomic sentence or negated atomic sentence: P, \( \neg \) P
Examples of PL sentences

- \((P \land Q) \rightarrow R\)  
  “If it is hot and humid, then it is raining”

- \(Q \rightarrow P\)  
  “If it is humid, then it is hot”

- \(Q\)  
  “It is humid.”

- We’re free to choose better symbols, btw:
  
  \(Ho = “It is hot”\)
  
  \(Hu = “It is humid”\)
  
  \(R = “It is raining”\)
Propositional logic (PL)

- Simple language for showing key ideas and definitions
- User defines set of propositional symbols, e.g., P, Q
- User defines **semantics** of each propositional symbol:
  - P means “It is hot”, Q means “It is humid”, etc.
- A sentence (well formed formula) is defined as follows:
  - A symbol is a sentence
  - If S is a sentence, then $\neg S$ is a sentence
  - If S is a sentence, then $(S)$ is a sentence
  - If S and T are sentences, then $(S \vee T), (S \wedge T), (S \rightarrow T)$, and $(S \leftrightarrow T)$ are sentences
  - A sentence results from a finite number of applications of the rules
Some terms

• The meaning or **semantics** of a sentence determines its **interpretation**

• Given the truth values of all symbols in a sentence, it can be “evaluated” to determine its **truth value** (True or False)

• A **model** for a KB is a *possible world* – an assignment of truth values to propositional symbols that makes each sentence in the KB True
**Model for a KB**

- Let the KB be \([P \land Q \rightarrow R, \ Q \rightarrow P]\)
- What are the possible models? Consider all possible assignments of T|F to P, Q and R and check truth tables

<table>
<thead>
<tr>
<th>P</th>
<th>Q</th>
<th>R</th>
<th>Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>OK</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>OK</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>NO</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>NO</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>OK</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
<td>OK</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>NO</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>OK</td>
</tr>
</tbody>
</table>

- If KB is \([P \land Q \rightarrow R, \ Q \rightarrow P, \ Q]\), the **only** model is TTT
More terms

• A **valid sentence** or **tautology** is a sentence that’s True under all interpretations, no matter what the world is actually like or what the semantics is. Example: “It's raining or it's not raining”

• An **inconsistent sentence** or **contradiction** is a sentence that is False under all interpretations. The world is never like what it describes, as in “It's raining and it's not raining.”

• **P entails Q**, written P |= Q, means that whenever P is True, so is Q
  – In all models in which P is true, Q is also true
Truth tables

- Truth tables are used to define logical connectives
- And to determine when a complex sentence is true given the values of the symbols in it

Truth tables for the five logical connectives

<table>
<thead>
<tr>
<th>$P$</th>
<th>$Q$</th>
<th>$\neg P$</th>
<th>$P \land Q$</th>
<th>$P \lor Q$</th>
<th>$P \Rightarrow Q$</th>
<th>$P \Leftrightarrow Q$</th>
</tr>
</thead>
<tbody>
<tr>
<td>False</td>
<td>False</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>False</td>
<td>True</td>
<td>True</td>
<td>False</td>
<td>True</td>
<td>True</td>
<td>False</td>
</tr>
<tr>
<td>True</td>
<td>False</td>
<td>False</td>
<td>False</td>
<td>True</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>True</td>
<td>True</td>
<td>False</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
</tbody>
</table>

Example of a truth table used for a complex sentence

<table>
<thead>
<tr>
<th>$P$</th>
<th>$H$</th>
<th>$P \lor H$</th>
<th>$(P \lor H) \land \neg H$</th>
<th>$((P \lor H) \land \neg H) \Rightarrow P$</th>
</tr>
</thead>
<tbody>
<tr>
<td>False</td>
<td>False</td>
<td>False</td>
<td>False</td>
<td>True</td>
</tr>
<tr>
<td>False</td>
<td>True</td>
<td>True</td>
<td>False</td>
<td>True</td>
</tr>
<tr>
<td>True</td>
<td>False</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>True</td>
<td>True</td>
<td>True</td>
<td>False</td>
<td>True</td>
</tr>
</tbody>
</table>
On the implies connective: \( P \rightarrow Q \)

- Note that \( \rightarrow \) is a logical connective
- So \( P \rightarrow Q \) is a logical sentence and has a truth value, i.e., is either true or false
- If we add this sentence to the KB, it can be used by an inference rule, *Modes Ponens*, to derive/infer/prove \( Q \) if \( P \) is also in the KB
- Given a KB where \( P=\text{True} \) and \( Q=\text{True} \), we can also derive/infer/prove that \( P \rightarrow Q \) is True
P → Q

• When is $P \rightarrow Q$ true? Check all that apply
  - P=Q=true
  - P=Q=false
  - P=true, Q=false
  - P=false, Q=true
P → Q

• When is \( P \rightarrow Q \) true? Check all that apply
  
  ✓ P=Q=true
  ✓ P=Q=false
  ✓ P=true, Q=false
  ✓ P=false, Q=true

• We can get this from the truth table for →

• Note: in FOL it's much harder to prove that a conditional true

  –Consider proving \( \text{prime}(x) \rightarrow \text{odd}(x) \)
Inference rules

• **Logical inference** creates new sentences that logically follow from a set of sentences (KB)
• An inference rule is **sound** if every sentence $X$ it produces when operating on a KB logically follows from the KB
  – i.e., inference rule creates no contradictions
• An inference rule is **complete** if it can produce every expression that logically follows from (is entailed by) the KB.
  – Note analogy to complete search algorithms
Sound rules of inference

- Here are some examples of sound rules of inference
- Each can be shown to be sound using a truth table

<table>
<thead>
<tr>
<th>RULE</th>
<th>PREMISE</th>
<th>CONCLUSION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modus Ponens</td>
<td>A, A → B</td>
<td>B</td>
</tr>
<tr>
<td>And Introduction</td>
<td>A, B</td>
<td>A ∧ B</td>
</tr>
<tr>
<td>And Elimination</td>
<td>A ∧ B</td>
<td>A</td>
</tr>
<tr>
<td>Double Negation</td>
<td>¬¬A</td>
<td>A</td>
</tr>
<tr>
<td>Unit Resolution</td>
<td>A ∨ B, ¬B</td>
<td>A</td>
</tr>
<tr>
<td><strong>Resolution</strong></td>
<td><strong>A ∨ B, ¬B ∨ C</strong></td>
<td><strong>A ∨ C</strong></td>
</tr>
</tbody>
</table>
## Soundness of modus ponens

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>A → B</th>
<th>OK?</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td>True</td>
<td>True</td>
<td>√</td>
</tr>
<tr>
<td>True</td>
<td>False</td>
<td>False</td>
<td>√</td>
</tr>
<tr>
<td>False</td>
<td>True</td>
<td>True</td>
<td>√</td>
</tr>
<tr>
<td>False</td>
<td>False</td>
<td>True</td>
<td>√</td>
</tr>
</tbody>
</table>
Resolution

• Resolution is a valid inference rule producing a new clause implied by two clauses containing complementary literals
  – A literal is an atomic symbol or its negation, i.e., P, ~P

• Amazingly, this is the only interference rule you need to build a sound and complete theorem prover
  – Based on proof by contradiction and usually called resolution refutation

• The resolution rule was discovered by Alan Robinson (CS, U. of Syracuse) in the mid 1960s
Resolution

- A KB is actually a set of sentences all of which are true, i.e., a conjunction of sentences.
- To use resolution, put KB into conjunctive normal form (CNF) where each is a disjunction of (one or more) literals (positive or negative atoms)
- Every KB can be put into CNF, it's just a matter of rewriting its sentences using standard tautological rules

\[
\begin{align*}
-(A \rightarrow B) & \iff (\neg A \lor B) \\
-(A \lor (B \land C)) & \iff (A \lor B) \land (A \lor C) \\
-A \land B & \rightarrow A \\
-A \land B & \rightarrow B
\end{align*}
\]
Resolution Example

- KB: \([P \rightarrow Q, Q \rightarrow R \land S]\)
- KB in CNF: \([\neg P \lor Q, \neg Q \lor R, \neg Q \lor S]\)
- Resolve KB(1) and KB(2) producing:
  \(\neg P \lor R\) (i.e., \(P \rightarrow R\))
- Resolve KB(1) and KB(3) producing:
  \(\neg P \lor S\) (i.e., \(P \rightarrow S\))
- New KB: \([\neg P \lor Q, \neg Q \lor R, \neg Q \lor S, \neg P \lor R, \neg P \lor S]\)

Tautologies

\[(A \rightarrow B) \leftrightarrow (\neg A \lor B)\]
\[(A \lor (B \land C)) \leftrightarrow (A \lor B) \land (A \lor C)\]
Soundness of the resolution inference rule

From the rightmost three columns of this truth table, we can see that

\[ (\alpha \lor \beta) \land (\neg \beta \lor \gamma) \leftrightarrow (\alpha \lor \gamma) \]

is valid (i.e., always true regardless of the truth values assigned to \(\alpha\), \(\beta\) and \(\gamma\))
Proving things

• A **proof** is a sequence of sentences, where each is a premise (i.e., a given) or is derived from earlier sentences in the proof by an inference rule
• The last sentence is the **theorem** (also called goal or query) that we want to prove
• Example for the “weather problem”

1. Hu premise “It's humid”
2. Hu → Ho premise “If it's humid, it's hot”
3. Ho modus ponens(1,2) “It's hot”
4. (Ho ∧ Hu) → R premise “If it's hot & humid, it's raining”
5. Ho ∧ Hu and introduction(1,3) “It's hot and humid”
6. R modus ponens(4,5) “It's raining”
Horn* sentences

• A **Horn sentence** or **Horn clause** has the form:

\[ P_1 \land P_2 \land P_3 \ldots \land P_n \rightarrow Q_m \text{ where } n \geq 0, m \in \{0, 1\} \]

• Note: a conjunction of 0 or more symbols to left of \( \rightarrow \) and 0-1 symbols to right

• Special cases:
  – \( n=0, \ m=1: \ P \) (assert \( P \) is true)
  – \( n>0, \ m=0: \ P \land Q \rightarrow \) (constraint: both \( P \) and \( Q \) can’t be true)
  – \( n=0, \ m=0: \) (well, there is nothing there!)

• Put in CNF: each sentence is a disjunction of literals with at most one non-negative literal

\[ \neg P_1 \lor \neg P_2 \lor \neg P_3 \ldots \lor \neg P_n \lor Q \]

\[ (P \rightarrow Q) = (\neg P \lor Q) \]

* After Alfred Horn*
Significance of Horn logic

• We can also have horn sentences in FOL
• Reasoning with horn clauses is much simpler
  – Satisfiability of a propositional KB (i.e., finding values for a symbols that will make it true) is NP complete
  – Restricting KB to horn sentences, satisfiability is in P
• For this reason, FOL Horn sentences are the basis for many rule-based languages, including Prolog and Datalog
• Horn logic gives up handling, in a general way, (1) negation and (2) disjunctions
Entailment and derivation

• **Entailment: KB |= Q**
  
  – Q is entailed by KB (set sentences) iff there is no logically possible world where Q is false while all the sentences in KB are true
  
  – Or, stated positively, Q is entailed by KB iff the conclusion is true in every logically possible world in which all the premises in KB are true

• **Derivation: KB |- Q**
  
  – We can derive Q from KB if there's a proof consisting of a sequence of valid inference steps starting from the premises in KB and resulting in Q
Two important properties for inference

**Soundness:** If $\text{KB} \vdash Q$ then $\text{KB} \models Q$

- If $Q$ is derived from $\text{KB}$ using a given set of rules of inference, then $Q$ is entailed by $\text{KB}$
- Hence, inference produces only real entailments, or any sentence that follows deductively from the premises is valid

**Completeness:** If $\text{KB} \models Q$ then $\text{KB} \vdash Q$

- If $Q$ is entailed by $\text{KB}$, then $Q$ can be derived from $\text{KB}$ using the rules of inference
- Hence, inference produces all entailments, or all valid sentences can be proved from the premises
Problems with Propositional Logic
Propositional logic: pro and con

• **Advantages**
  – Simple KR language sufficient for some problems
  – Lays the foundation for higher logics (e.g., FOL)
  – Reasoning is decidable, though NP complete, and efficient techniques exist for many problems

• **Disadvantages**
  – Not expressive enough for most problems
  – Even when it is, it can very “un-concise”
PL is a weak KR language

- Hard to identify “individuals” (e.g., Mary, 3)
- Can’t directly talk about properties of individuals or relations between individuals (e.g., “Bill is tall”)
- Generalizations, patterns, regularities can’t easily be represented (e.g., “all triangles have 3 sides”)
- First-Order Logic (FOL) is expressive enough to represent this kind of information using relations, variables and quantifiers, e.g.,
  - Every elephant is gray: $\forall x \ (\text{elephant}(x) \rightarrow \text{gray}(x))$
  - There is a white alligator: $\exists x \ (\text{alligator}(X) \land \text{white}(X))$
PL Example

• Consider the problem of representing the following information:
  – Every person is mortal.
  – Confucius is a person.
  – Confucius is mortal.

• How can these sentences be represented so that we can infer the third sentence from the first two?
PL Example

• In PL we have to create propositional symbols to stand for all or part of each sentence, e.g.:
  P = “person”; Q = “mortal”; R = “Confucius”

• The above 3 sentences are represented as:
  P → Q; R → P; R → Q

• The 3rd sentence is entailed by the first two, but we need an explicit symbol, R, to represent an individual, Confucius, who is a member of the classes person and mortal

• Representing other individuals requires introducing separate symbols for each, with some way to represent the fact that all individuals who are “people” are also “mortal”
Hunt the Wumpus domain

- Some atomic propositions:
  - $S_{12}$ = There is a stench in cell (1,2)
  - $B_{34}$ = There is a breeze in cell (3,4)
  - $W_{22}$ = Wumpus is in cell (2,2)
  - $V_{11}$ = We’ve visited cell (1,1)
  - $OK_{11}$ = Cell (1,1) is safe
  ...

- Some rules:
  - $\neg S_{22} \rightarrow \neg W_{12} \land \neg W_{23} \land \neg W_{32} \land \neg W_{21}$
  - $S_{22} \rightarrow W_{12} \lor W_{23} \lor W_{32} \lor W_{21}$
  - $B_{22} \rightarrow P_{12} \lor P_{23} \lor P_{32} \lor P_{21}$
  - $W_{22} \rightarrow S_{12} \land S_{23} \land S_{23} \land W_{21}$
  - $W_{22} \rightarrow \neg W_{11} \land \neg W_{21} \land \ldots \land \neg W_{44}$
  - $A_{22} \rightarrow V_{22}$
  - $A_{22} \rightarrow \neg W_{11} \land \neg W_{21} \land \ldots \land \neg W_{44}$
  - $V_{22} \rightarrow OK_{22}$
Hunt the Wumpus domain

• Eight variables for each cell: e.g., A11, B11, G11, OK11, P11, S11, V11, W11
• The lack of variables requires us to give similar rules for each cell!
• Ten rules (I think) for each cell:
  A11 → ...
  V11 → ...
  P11 → ...
  ¬P11 → ...
  W11 → ...
  ¬W11 → ...
  S11 → ...
  ¬S11 → ...
  B11 → ...
  ¬B11 → ...
After third move

- We can prove that the Wumpus is in (1,3) using these four rules
- See R&N section 7.5

\[(R1) \neg S_{11} \rightarrow \neg W_{11} \land \neg W_{12} \land \neg W_{21}\]
\[(R2) \neg S_{21} \rightarrow \neg W_{11} \land \neg W_{21} \land \neg W_{22} \land \neg W_{31}\]
\[(R3) \neg S_{12} \rightarrow \neg W_{11} \land \neg W_{12} \land \neg W_{22} \land \neg W_{13}\]
\[(R4) S_{12} \rightarrow W_{13} \lor W_{12} \lor W_{22} \lor W_{11}\]
Proving W13

Applying MP with \( \neg S_{11} \) and \( R_1 \):
\[ \neg W_{11} \land \neg W_{12} \land \neg W_{21} \]

Applying And-Elimination to this, yielding 3 sentences:
\[ \neg W_{11}, \neg W_{12}, \neg W_{21} \]

Applying MP to \( \neg S_{21} \) and \( R_2 \), then applying And-elimination:
\[ \neg W_{22}, \neg W_{21}, \neg W_{31} \]

Applying MP to \( S_{12} \) and \( R_4 \) to obtain:
\[ W_{13} \lor W_{12} \lor W_{22} \lor W_{11} \]

Applying Unit Resolution on \( (W_{13} \lor W_{12} \lor W_{22} \lor W_{11}) \) and \( \neg W_{11} \):
\[ W_{13} \lor W_{12} \lor W_{22} \]

Applying Unit Resolution with \( (W_{13} \lor W_{12} \lor W_{22}) \) and \( \neg W_{22} \):
\[ W_{13} \lor W_{12} \]

Applying Unit Resolution with \( (W_{13} \lor W_{12}) \) and \( \neg W_{12} \):
\[ W_{13} \]

QED
Propositional Wumpus hunter problems

• Lack of variables prevents stating more general rules
  • $\forall x, y \ V(x,y) \rightarrow OK(x,y)$
  • $\forall x, y \ S(x,y) \rightarrow W(x-1,y) \lor W(x+1,y) \ldots$

• Change of the KB over time is difficult to represent
  – In classical logic, a fact is true or false for all time
  – A standard technique is to index dynamic facts with the time when they’re true
    • A(1, 1, t0)
  – Thus we have a separate KB for every time point
Propositional logic summary

- Inference: process of deriving new sentences from old
  - **Sound** inference derives true conclusions given true premises
  - **Complete** inference derives all true conclusions from a set of premises

- **Valid sentence:** true in all worlds under all interpretations

- If an implication sentence can be shown to be valid, then, given its premise, its consequent can be derived

- Different logics make different **commitments** about what the world is made of and the kind of beliefs we can have

- **Propositional logic** commits only to the existence of facts that may or may not be the case in the world being represented
  - Simple syntax and semantics suffices to illustrate the process of inference
  - Propositional logic can become impractical, even for very small worlds