Logic

Virginia Tech CS 5804
Spring 2015
Outline

- Propositional logic syntax and semantics
- Inference in propositional logic
- (Horn clauses, forward/backward chaining)
Terminology
Terminology

• Knowledge base (KB): a set of logical sentences
Terminology

• **Knowledge base (KB):** a set of logical **sentences**
• **Axioms** are assumed true without **proof**
Terminology

- **Knowledge base** (KB): a set of logical *sentences*
- **Axioms** are assumed true without *proof*
  - New sentences can be found through *inference*
Terminology

• **Knowledge base** (KB): a set of logical **sentences**
• **Axioms** are assumed true without **proof**
  • New sentences can be found through **inference**
  • Logic has **syntax** and **semantics**
Terminology

• **Knowledge base** (KB): a set of logical **sentences**
• **Axioms** are assumed true without **proof**
  • New sentences can be found through **inference**
  • Logic has **syntax** and **semantics**
• **Logical** sentences have **truth**
Terminology

- **Knowledge base** (KB): a set of logical **sentences**
- **Axioms** are assumed true without **proof**
  - New sentences can be found through **inference**
  - Logic has **syntax** and **semantics**
- **Logical** sentences have **truth**
- A KB has **possible worlds** or **models**: instantiations of variables where all sentences are true, i.e., **satisfy**
Terminology

- Knowledge base (KB): a set of logical sentences
- Axioms are assumed true without proof
  - New sentences can be found through inference
  - Logic has syntax and semantics
- Logical sentences have truth
- A KB has possible worlds or models: instantiations of variables where all sentences are true, i.e., satisfy
- New sentences that all models also satisfy are entailed by KB
Terminology

- **Knowledge base** (KB): a set of logical sentences
- **Axioms** are assumed true without **proof**
  - New sentences can be found through **inference**
  - Logic has **syntax** and **semantics**
- **Logical** sentences have **truth**
- A KB has **possible worlds** or **models**: instantiations of variables where all sentences are true, i.e., **satisfy**
- New sentences that all models also satisfy are **entailed** by KB
- Inference methods can be **sound**: only infer truly entailed sentences
Terminology

- **Knowledge base** (KB): a set of logical **sentences**
- **Axioms** are assumed true without **proof**
  - New sentences can be found through **inference**
  - Logic has **syntax** and **semantics**
- **Logical** sentences have **truth**
- A KB has **possible worlds** or **models**: instantiations of variables where all sentences are true, i.e., **satisfy**
- New sentences that all models also satisfy are **entailed** by KB
- Inference methods can be **sound**: only infer truly entailed sentences
  - and it can be **complete**: able to infer any sentence that is entailed
Propositional Logic
Propositional Logic

• Atomic sentences are Boolean variables: X, Y, Z (atoms)
Propositional Logic

• Atomic sentences are Boolean variables: X, Y, Z (atoms)

• Complex sentences combine simpler sentences via connectives
Propositional Logic

• Atomic sentences are Boolean variables: X, Y, Z (atoms)

• Complex sentences combine simpler sentences via connectives

  • **not, negation**: \( \neg X, \neg Y \)
Propositional Logic

• Atomic sentences are Boolean variables: X, Y, Z (atoms)

• Complex sentences combine simpler sentences via connectives
  • not, negation: ¬X, ¬Y
  • and, conjunction: X ∧ Y
Propositional Logic

- Atomic sentences are Boolean variables: X, Y, Z (atoms)
- Complex sentences combine simpler sentences via connectives
  - not, negation: \( \neg X, \neg Y \)
  - and, conjunction: \( X \land Y \)
  - or, disjunction: \( X \lor Y \)
Propositional Logic

• Atomic sentences are Boolean variables: X, Y, Z (atoms)

• Complex sentences combine simpler sentences via connectives
  • not, negation: \( \neg X, \neg Y \)
  • and, conjunction: \( X \land Y \)
  • or, disjunction: \( X \lor Y \)
  • implication: \( X \Rightarrow Y \)
Propositional Logic

• Atomic sentences are Boolean variables: X, Y, Z (atoms)

• Complex sentences combine simpler sentences via connectives
  • not, negation: \( \neg X, \neg Y \)
  • and, conjunction: \( X \land Y \)
  • or, disjunction: \( X \lor Y \)
  • implication: \( X \Rightarrow Y \)
  • double implication, biconditional: \( X \Leftrightarrow Y \)
Propositional Logic

- Atomic sentences are Boolean variables: X, Y, Z (atoms)
- Complex sentences combine simpler sentences via connectives
  - not, negation: ¬X, ¬Y
  - and, conjunction: X ∧ Y
  - or, disjunction: X ∨ Y
  - implication: X ⇒ Y
  - double implication, biconditional: X ⇔ Y

• provable: X ⊢ Y
Propositional Logic

• Atomic sentences are Boolean variables: X, Y, Z (atoms)

• Complex sentences combine simpler sentences via connectives
  • not, negation: \( \neg X, \neg Y \)
  • and, conjunction: \( X \land Y \)
  • or, disjunction: \( X \lor Y \)
  • implication: \( X \Rightarrow Y \)
  • double implication, biconditional: \( X \Leftrightarrow Y \)

• provable: \( X \vdash Y \)
• entails: \( X \vDash Z \)
Propositional Logic

- Atomic sentences are Boolean variables: X, Y, Z (atoms)
- Complex sentences combine simpler sentences via connectives
  - not, negation: ¬X, ¬Y, !X
  - and, conjunction: X ∧ Y, X && Y
  - or, disjunction: X ∨ Y, X || Y
  - implication: X ⇒ Y
  - double implication, biconditional: X ⇔ Y
  - provable: X ⊢ Y
  - entails: X ⊨ Z
Propositional Logic

• Atomic sentences are Boolean variables: X, Y, Z (atoms)

• Complex sentences combine simpler sentences via connectives
  • not, negation: \( \neg X, \neg Y \)
  • and, conjunction: \( X \land Y \)
  • or, disjunction: \( X \lor Y \)
  • implication: \( X \Rightarrow Y \)
  • double implication, biconditional: \( X \Leftrightarrow Y \)

  literal: X, \( \neg X \), Y, \( \neg Y \), Z, \( \neg Z \)

  • provable: \( X \vdash Y \)
  • entails: \( X \models Z \)
Inference

• Given KB with prop. logic sentences, check if sentence $\alpha$ is true

• Naive approach: evaluate full truth table
  - complete and sound inference but expensive

• Better approach: execute inference rules on KB: theorem proving

<table>
<thead>
<tr>
<th>KB: $X \land Y$</th>
<th>$X = F$</th>
<th>$X = T$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Y = F$</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>$Y = T$</td>
<td>F</td>
<td>T</td>
</tr>
</tbody>
</table>
Inference Rules
Inference Rules

• Inference rules preserve **logical equivalence**
Inference Rules

- Inference rules preserve logical equivalence
- **Validity**: a sentence is valid if it is true for all possible worlds
Inference Rules

- Inference rules preserve **logical equivalence**

- **Validity**: a sentence is valid if it is true for all possible worlds

- **Satisfiability**: a sentence is satisfiable if it is true for *at least one* possible world
Inference Rules

• Inference rules preserve logical equivalence

• Validity: a sentence is valid if it is true for all possible worlds

• Satisfiability: a sentence is satisfiable if it is true for at least one possible world

• Modus ponens: \( \alpha \Rightarrow \beta, \ \alpha \Rightarrow \beta \)
Inference Rules

• Inference rules preserve logical equivalence

• Validity: a sentence is valid if it is true for all possible worlds

• Satisfiability: a sentence is satisfiable if it is true for at least one possible world

• Modus ponens: \[ \alpha \Rightarrow \beta, \alpha \quad \text{in KB} \]

\[ \beta \]
Inference Rules

- Inference rules preserve **logical equivalence**

- **Validity**: a sentence is valid if it is true for all possible worlds

- **Satisfiability**: a sentence is satisfiable if it is true for *at least one* possible world

- **Modus ponens**:
  
  $\alpha \Rightarrow \beta, \alpha \quad \text{in KB}$

  $\beta \quad \text{new sentence}$
Inference rules preserve logical equivalence

- **Validity**: a sentence is valid if it is true for all possible worlds

- **Satisfiability**: a sentence is satisfiable if it is true for *at least one* possible world

- **Modus ponens**: \[ \alpha \Rightarrow \beta, \quad \alpha \implies \beta \]
Inference rules preserve logical equivalence

- **Validity**: a sentence is valid if it is true for all possible worlds

- **Satisfiability**: a sentence is satisfiable if it is true for *at least one* possible world

- **Modus ponens**: \( \alpha \Rightarrow \beta, \alpha \quad \frac{}{\beta} \)

- **And-elimination**: \( \alpha \land \beta \quad \frac{}{\alpha} \)
Inference rules preserve logical equivalence

- **Validity**: a sentence is valid if it is true for all possible worlds
- **Satisfiability**: a sentence is satisfiable if it is true for *at least one* possible world
- **Modus ponens**: $\alpha \Rightarrow \beta, \alpha \quad \Rightarrow \beta$
- **And-elimination**: $\alpha \land \beta \quad \Rightarrow \alpha$
Inference rules preserve logical equivalence

- **Validity**: a sentence is valid if it is true for all possible worlds

- **Satisfiability**: a sentence is satisfiable if it is true for at least one possible world

- **Modus ponens**:

\[
\frac{\alpha \Rightarrow \beta, \ \alpha}{\beta}
\]

- **And-elimination**:

\[
\frac{\alpha \land \beta}{\alpha} \text{ in KB}
\]

\[
\text{new sentence}
\]
Category: Rules of inference

The main article for this category is Rules of inference.

The concepts described in articles in this category may be also expressed in terms of arguments, or theorems. Very often the same concept is in more than one of these categories, expressed a different way and sometimes with a different name.

Pages in category "Rules of inference"

The following 40 pages are in this category, out of 40 total. This list may not reflect recent changes (learn more).

D cont.
- Disjunction elimination
- Disjunction introduction
- Disjunctive syllogism
- Distributive property
- Double negative elimination

M cont.
- Modus ponens
- Modus tollens

N
- Negation as failure
- Negation introduction

R
- Resolution (logic)
- Rule of replacement

A
- Absorption (logic)
- Admissible rule
- Associative property

E
- Existential generalization
- Existential instantiation
Inference via Search
Inference via Search

• States: knowledge bases
  Actions: apply inference rule, add new sentence to KB
  Goal: add target sentence $\alpha$ to KB
Inference via Search

- States: knowledge bases
  Actions: apply inference rule, add new sentence to KB
  Goal: add target sentence $\alpha$ to KB

- Sound?
Inference via Search

- States: knowledge bases
  Actions: apply inference rule, add new sentence to KB
  Goal: add target sentence $\alpha$ to KB

- Sound?

- Complete?
Inference via Search

• States: knowledge bases
Actions: apply inference rule, add new sentence to KB
Goal: add target sentence $\alpha$ to KB

• Sound? Yes!

• Complete?
Inference via Search

• States: knowledge bases
  Actions: apply inference rule, add new sentence to KB
  Goal: add target sentence $\alpha$ to KB

• Sound? Yes!

• Complete? Not necessarily
Resolution

\[ X_1 \lor X_2 \lor \ldots \lor X_N \lor \beta, \quad Y_1 \lor Y_2 \lor \ldots \lor Y_M \lor \neg \beta \]

\[ \frac{X_1 \lor X_2 \lor \ldots \lor X_N \lor Y_1 \lor Y_2 \lor \ldots \lor Y_M}{X_1 \lor X_2 \lor \ldots \lor X_N \lor Y_1 \lor Y_2 \lor \ldots \lor Y_M} \]
Resolution

\[ X_1 \lor X_2 \lor \ldots \lor X_N \lor \beta, \quad Y_1 \lor Y_2 \lor \ldots \lor Y_M \lor \neg \beta \]

\[ X_1 \lor X_2 \lor \ldots \lor X_N \lor Y_1 \lor Y_2 \lor \ldots \lor Y_M \]
Resolution

\[ X_1 \lor X_2 \lor \ldots \lor X_N \lor \beta, \quad Y_1 \lor Y_2 \lor \ldots \lor Y_M \lor \neg \beta \]

\[ X_1 \lor X_2 \lor \ldots \lor X_N \lor Y_1 \lor Y_2 \lor \ldots \lor Y_M \]
Resolution

\[ X_1 \lor X_2 \lor \ldots \lor X_N \lor \beta, \quad Y_1 \lor Y_2 \lor \ldots \lor Y_M \lor \neg \beta \]

\[ X_1 \lor X_2 \lor \ldots \lor X_N \lor Y_1 \lor Y_2 \lor \ldots \lor Y_M \]
Resolution

\[ X_1 \lor X_2 \lor \ldots \lor X_N \lor \beta, \quad Y_1 \lor Y_2 \lor \ldots \lor Y_M \lor \neg \beta \]
Resolution

\[ X_1 \lor X_2 \lor \ldots \lor X_N \lor \beta, \quad Y_1 \lor Y_2 \lor \ldots \lor Y_M \lor \neg \beta \]

\[ X_1 \lor X_2 \lor \ldots \lor X_N \lor Y_1 \lor Y_2 \lor \ldots \lor Y_M \]
Resolution

\[ X_1 \lor X_2 \lor \ldots \lor X_N \lor \beta, \quad Y_1 \lor Y_2 \lor \ldots \lor Y_M \lor \neg \beta \]
Resolution

Bob is a cat or lazy       Bob is a bird or he is not lazy

Bob is a cat or a bird

\[
\begin{align*}
X_1 \lor X_2 \lor \ldots \lor X_N \lor \beta, & \quad Y_1 \lor Y_2 \lor \ldots \lor Y_M \lor \neg \beta \\
X_1 \lor X_2 \lor \ldots \lor X_N \lor \beta, & \quad Y_1 \lor Y_2 \lor \ldots \lor Y_M
\end{align*}
\]
Resolution

\[ A \Rightarrow B \equiv \neg A \lor B \]
Resolution

\[ A \Rightarrow B \equiv \neg A \lor B \]

\[ A \Rightarrow B, \quad A \]

\[ \frac{B}{B} \]

\[ X_1 \lor X_2 \lor \ldots \lor X_N \lor \beta, \quad X_1 \lor X_2 \lor \ldots \lor X_N \lor \beta \]

\[ Y_1 \lor Y_2 \lor \ldots \lor Y_M \lor \neg \beta, \quad Y_1 \lor Y_2 \lor \ldots \lor Y_M \lor \neg \beta \]
Resolution

\[ A \Rightarrow B \equiv \neg A \lor B \]

\[
\begin{align*}
A \Rightarrow B, & \quad A \\
\hline
\Rightarrow & \quad B
\end{align*}
\]

\[
\begin{align*}
\neg A \lor B, & \quad A \\
\hline
\Rightarrow & \quad B
\end{align*}
\]

\[
\begin{align*}
X_1 \lor X_2 \lor \ldots \lor X_N \lor \beta, & \quad Y_1 \lor Y_2 \lor \ldots \lor Y_M \lor \neg \beta \\
X_1 \lor X_2 \lor \ldots \lor X_N \lor \beta, & \quad Y_1 \lor Y_2 \lor \ldots \lor Y_M
\end{align*}
\]
Conjunctive Normal Form

- Conjunction of **clauses** (disjunctions of literals)

\[(X_1 \lor X_2 \lor X_3) \land (Y_1 \lor Y_2 \lor Y_3) \land (Z_1 \lor Z_2 \lor Z_3)\]
Conjunctive Normal Form

• Conjunction of **clauses** (disjunctions of literals)

\[
(X_1 \lor X_2 \lor X_3) \land (Y_1 \lor Y_2 \lor Y_3) \land (Z_1 \lor Z_2 \lor Z_3)
\]

\[
\alpha \leftrightarrow \beta \equiv (\alpha \Rightarrow \beta) \land (\beta \Leftarrow \alpha)
\]
\[
\alpha \Rightarrow \beta \equiv \neg \alpha \lor \beta
\]
\[
\neg (\neg \alpha) \equiv \alpha
\]
\[
\neg (\alpha \land \beta) \equiv (\neg \alpha \lor \neg \beta)
\]
\[
\neg (\alpha \lor \beta) \equiv (\neg \alpha \land \neg \beta)
\]
\[
(\alpha \land (\beta \lor \gamma)) \equiv (\alpha \land \beta) \lor (\alpha \land \gamma)
\]
\[
(\alpha \lor (\beta \land \gamma)) \equiv (\alpha \lor \beta) \land (\alpha \lor \gamma)
\]
Horn clauses

• Restricted logic. Sentences must be disjunctions with at most one positive literal

• equivalent to conjunction implying an atom

\[ \neg X_1 \lor \neg X_2 \lor \neg X_3 \lor Y \]

\[ X_1 \land X_2 \land X_3 \Rightarrow Y \]
Horn clauses

- Restricted logic. Sentences must be disjunctions with at most one positive literal

- equivalent to conjunction implying an atom

\[
\neg X_1 \lor \neg X_2 \lor \neg X_3 \lor Y
\]

\[
X_1 \land X_2 \land X_3 \Rightarrow Y
\]
Horn clauses

• Restricted logic. Sentences must be disjunctions with at most one positive literal

• equivalent to conjunction implying an atom

\neg X_1 \lor \neg X_2 \lor \neg X_3 \lor Y

\text{forward-chaining}
Horn clauses

- Restricted logic. Sentences must be disjunctions with at most one positive literal

- equivalent to conjunction implying an atom

\[ \neg X_1 \lor \neg X_2 \lor \neg X_3 \lor Y \]
\[ X_1 \land X_2 \land X_3 \Rightarrow Y \]

forward-chaining
Horn clauses

• Restricted logic. Sentences must be disjunctions with at most one positive literal

• equivalent to conjunction implying an atom

\[ \neg X_1 \lor \neg X_2 \lor \neg X_3 \lor Y \]
\[ X_1 \land X_2 \land X_3 \Rightarrow Y \]

forward-chaining  backward-chaining
Summary

• Propositional logic syntax and semantics
• Inference in propositional logic: table, inference rules, resolution
• Horn clauses, forward/backward chaining