Propositional Logic, Part 2

Lecture 15

Chapter 7, Sections 7.4

Jim Rehg

College of Computing

Georgia Tech

February 15, 2016

Elements of Propositional Logic

Syntax – Defines the allowable sentences

Atomic sentences – Single proposition $P, Q, \ldots$ either True or False

Complex sentences – Built using connectives, e.g. $P \wedge Q$

Semantics – Identifies sentences that are True in a model

Truth tables for connectives

Logical Reasoning – Entailment: $\mathrm{KB} \vDash \alpha$

α is entailed by KB if in every model where KB is True, α is True

Inference Algorithm – Method for deriving sentences

$\mathrm{KB} \vdash \alpha$ (α is derived from KB)

Must be sound and complete

Soundness and Completeness

Soundness

Inference engine only derives entailed sentences (truth-preserving)

$\mathrm{KB} \vdash \alpha$ implies $\mathrm{KB} \models \alpha$

Completeness

Inference engine can derive any sentence that is entailed

$\mathrm{KB} \models \alpha$ implies $\mathrm{KB} \vdash \alpha$

These properties are important because they allow us to ignore the semantics and just manipulate symbols

Prepositional Logic: Syntax

Propositional symbols $P, Q, R, S, \ldots$

Logical Connectives

- Negation – If P is sentence, $\neg P$ is also sentence
- Conjunction (AND) – If P and Q sentences, $P \land Q$ is sentence
- Disjunction (OR) – If P and Q sentences, $P \lor Q$ is sentence
- Implication – If P and Q sentences, $P \Rightarrow Q$ is sentence
- Biconditional – If P and Q sentences, $P \Leftrightarrow Q$ is sentence

Propositional Logic: Semantics

Model (possible world) m:

Assignment of truth values to symbols (e.g. $m = \{P = T, Q = F\}$)

If sentence α is true in model m, then m satisfies α and $m \in M(\alpha)$

e.g. if $\alpha = P \wedge Q$ then $M(\alpha) = \{\{P = T, Q = T\}\}$

Sentence α:

α is valid (or is a tautology) if it is true in every model (e.g $P \vee \neg P$)

α is a contradiction if it is false in every model (e.g. $P \wedge \neg P$)

α is satisfiable if it is true in at least one model

Entailment

$\beta \vDash \alpha$ if and only if $M(\beta) \subseteq M(\alpha)$

Propositional Logic: Truth Tables

P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \Rightarrow Q$	$P \Leftrightarrow Q$
False	False	True	False	False	True	True
False	True	True	False	True	True	False
True	False	False	False	True	False	False
True	True	False	True	True	True	True

Example Knowledge Base (KB)

$$\text{KB} = (A \vee B) \wedge (\neg C \vee A)$$

A	B	C	KB
F	F	F	F
F	F	T	F
F	T	F	T
F	T	T	F
T	F	F	T
T	F	T	T
T	T	F	T
T	T	T	T

Entailment

$$\mathrm{KB} = (A \vee B) \wedge (\neg C \vee A)$$

$S = A \wedge C$

Does $\mathrm{KB} \vDash S$?

A	B	C	KB	S
F	F	F	F	F
F	F	T	F	F
F	T	F	$\boxed{T}$	$\boxed{F}$
F	T	T	F	F
T	F	F	T	F
T	F	T	T	T
T	T	F	T	F
T	T	T	T	T

KB $\not\vDash$ S because KB is true but *S* is false

Entailment

$$\text{KB} = (A \vee B) \wedge (\neg C \vee A)$$

$S = A \vee B \vee C$

Does $\text{KB} \vDash S$?

A	B	C	KB	S
F	F	F	F	F
F	F	T	F	T
F	T	F	T	T
F	T	T	F	T
T	F	F	T	T
T	F	T	T	T
T	T	F	T	T
T	T	T	T	T

KB \|= *S* because *S* is true for all the assignments for which KB is true

Sentences in Wumpus World

Let $P_{i,j}$ be True if there is pit in (i,j)

Let $B_{i,j}$ be True if there is breeze in (i,j)

Knowledge Base

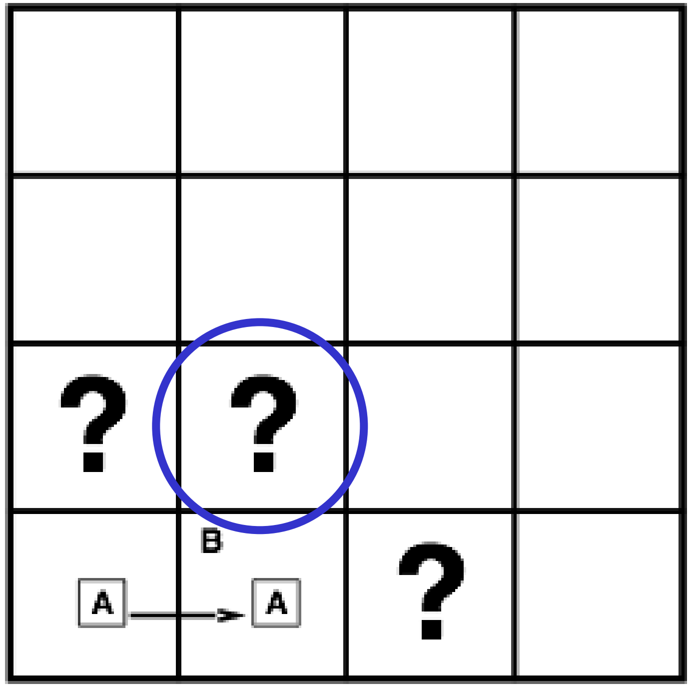
Pits cause breezes in adjacent squares

R_2: $B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$

R_3: $B_{2,1} \Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{3,1})$

Percepts

$R_1: \neg P_{1,1}$ $R_4: \neg B_{1,1}$ $R_5: B_{2,1}$



What does our KB say about $P_{2,2}$?

Entailment by Enumeration

Model | KB Sentences

$B_{1,1}$	$B_{2,1}$	$P_{1,1}$	$P_{1,2}$	$P_{2,1}$	$P_{2,2}$	$P_{3,1}$	R_1	R_2	R_3	R_4	R_5	KB
false	*false*	*false*	*false*	*false*	*false*	*false*	*true*	*true*	*true*	*true*	*false*	*false*
false	*false*	*false*	*false*	*false*	*false*	*true*	*true*	*true*	*false*	*true*	*false*	*false*
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
false	*true*	*false*	*false*	*false*	*false*	*false*	*true*	*true*	*false*	*true*	*true*	*false*
false	*true*	*false*	*false*	*false*	*false*	*true*	*true*	*true*	*true*	*true*	*true*	*true*
false	*true*	*false*	*false*	*false*	*true*	*false*	*true*	*true*	*true*	*true*	*true*	*true*
false	*true*	*false*	*false*	*false*	*true*	*true*	*true*	*true*	*true*	*true*	*true*	*true*
false	*true*	*false*	*false*	*true*	*false*	*false*	*true*	*false*	*false*	*true*	*true*	*false*
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
true	*true*	*true*	*true*	*true*	*true*	*true*	*false*	*true*	*true*	*false*	*true*	*false*

$\neg P_{1,2}$ entailed

$\neg P_{2,1}$ entailed

$\neg P_{2,2}$ *is not* entailed

Entailment by Enumeration

Sound: Yes, if it says S is entailed then it is

Complete: Yes, enumerating everything will find all entailed sentences

Complexity: $O(2^N)$ in number of symbols

Exponential in the size of the input, so not practical

Another Example: Autonomous Car

Knowledge-base describing when the car should brake?

$$
\begin{aligned}
&(\text{PersonInFrontOfCar} \Rightarrow \text{Brake}) \\
&\wedge (((\text{YellowLight} \wedge \text{Policeman}) \wedge (\neg\text{Slippery})) \Rightarrow \text{Brake}) \\
&\wedge (\text{Policecar} \Rightarrow \text{Policeman}) \\
&\wedge (\text{Snow} \Rightarrow \text{Slippery}) \\
&\wedge (\text{Slippery} \Rightarrow \neg\text{Dry}) \\
&\wedge (\text{RedLight} \Rightarrow \text{Brake})
\end{aligned}
$$

Observation from sensors:

$$
\begin{aligned}
&\text{YellowLight} \\
&\wedge \neg\text{RedLight} \\
&\wedge \neg\text{Snow} \\
&\wedge \text{Dry} \\
&\wedge \text{Policecar} \\
&\wedge \neg\text{PersonInFrontOfCar}
\end{aligned}
$$

Model Checking

- **Idea:**
  - To test whether $\alpha \models \beta$, enumerate all models and check truth of α and β.
  - α entails β if no model exists in which α is true and β is false (i.e. $(\alpha \wedge \neg\beta)$ is unsatisfiable)
- **Proof by Contradiction:**

  $\alpha \models \beta$ if and only if the sentence $(\alpha \wedge \neg\beta)$ is unsatisfiable.
- **Model Checking:**
  - Variables: One for each propositional symbol
  - Domains: {true, false}
  - Objective Function: $(\alpha \wedge \neg\beta)$
  - Which search algorithm works best?

Questions?