Logic, Meaning, Logical Reasoning

Andrew W . Moore

16-731/15-780 Spring 1997

Handout A WM1

Parts of this lecture f ollows Chapter 6 of Russell and Norvig

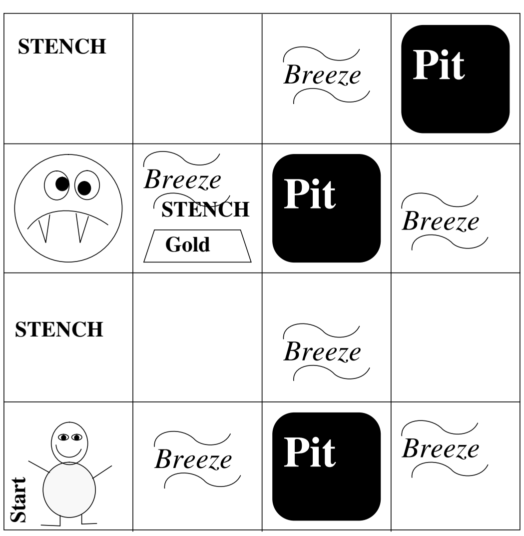
Overview

- The W umpus world
- How do you represent knowledge?
- Sentences
- The simplest F ormal Logic
- Syntax of a Logic
- Interpretations in a Logic
- Meaning (Semantics) of a Logic
- Brute force w ay of proving something is true .
- Efficiently showing something is true: logical inference .
- The W umpus W orld.
- Propositional Logic: The Dark Side .

Warning/Apology: Slides are not handcrafted with loving care! W ild la youts , hideous font choices and font inconsistencies abound!

The W umpus W orld

Here is a medium-complexity example of a problem in whic h a decision-making system needs to represent knowledge of the world and reason with it.



(Details , p153, R&N)

Reasoning in the W umpus world

If the rules of the world are presented formally to the decision-maker .

Then the decision-maker can use logical reasoning to safely explore this world.

A sentence

A sentence is a piece of information represented inside the system. (Very informal definition). Here are some sentences, some in formal languages, and a couple in natural language.

"If a professor gets tenure then they get fat".

$cos^2 x + sin^2 x = 1$

There's a stench at (2,1)

$a \vee b$

$a \vee \sim a$

$(a \wedge b) \Rightarrow \sim c$

$\exists x . Cat(x)$

$\forall x . Person(x) \Rightarrow (\exists y . y = Mother(x))$

Syntax

A syntax is a specification of which sentences can be constructed in a given formal logic.

Different kinds of formal logics have different syntaxes.

Example: The syntax of PROPOSITIONAL LOGIC:

Sentence ::= AtomicSentence | ComplexSentence

AtomicSentence ::= **True** | **False** | *Symbol*

ComplexSentence ::= (Sentence) |

Sentence Connective Sentence |

~Sentence

Connective ::= $\wedge$ | $\vee$ | <=> | =>

Symbol ::= P | Q | R | ....

Precedence (highest to lowest): ~ $\wedge$ $\vee$ => <=>

Semantics

How do specify what something MEANS?

That's the job of semantics.

A very importance concept.

Example:

In propositional logic, the semantics of the sentence $P \vee Q \Rightarrow R$ is

"The complete description of the world is one of the following:

- P is true , Q is True , R is True
- P is true , Q is False , R is True
- P is False , Q is False , R is True
- P is False , Q is False , R is True
- P is False , Q is False , R is False"

In general, in propositional logic, the meaning (i.e. semantics) of a sentence is a complete set of possible worlds.

Interpretations: the key to semantics

An interpretation is a logician's word for what I just called a "possible world".

Given 3 propositional symbols P, Q, R, there are 8 interpretations.

Given n propositional symbols $P_1, P_2 \dots P_n$, there are 2^n interpretations.

In propositional logic, an interpretation is a mapping from propositional symbols to truth values.

In general, in propositional logic, the meaning (i.e. semantics) of a sentence is the set of interpretations in which the sentence evaluates to True. (Read on for how to evaluate a sentence under a given interpretation).

Evaluating a sentence under interpretation I.

We'll define it by imagining we are writing a recursive function Eval(S,I) which takes a sentence and an interpretation. The behavior of Eval depends on the top-level form of the sentence.

Eval(True,I) = True

Eval(False,I) = False

Eval(P,I) = Look up the value of P in the interpretation I.

Eval($S_1 \vee S_2$,I) =

Call Eval(S_1,I), call Eval(S_2,I) then look up the result in the following truth table:

Eval(S_1,I)	Eval(S_2,I)	Eval($S_1 \vee S_2$,I)
True	True	True
True	False	True
False	True	True
False	False	False

Evaluating a sentence under interpretation I.

Eval($S_1 \wedge S_2$,I) =

Call Eval(S_1,I), call Eval(S_2,I) then look up the result in the following truth table:

Eval(S_1,I)	Eval(S_2,I)	Eval($S_1 \wedge S_2$,I)
True	True	True
True	False	False
False	True	False
False	False	False

Eval(S_1 <=> S_2,I) =

Call Eval(S_1,I), call Eval(S_2,I) then look up the result in the following truth table:

Eval(S_1,I)	Eval(S_2,I)	Eval(S_1 <=> S_2,I)
True	True	True
True	False	False
False	True	False
False	False	True

Evaluating a sentence under interpretation I.

Eval(S_1 => S_2,I) =

Call Eval(S_1,I), call Eval(S_2,I) then look up the result in the following truth table:

Eval(S_1,I)	Eval(S_2,I)	Eval(S_1 => S_2,I)
True	True	True
True	False	False
False	True	True
False	False	True

This one takes a bit of getting used to.

"a implies b" means when a is true, b must also be true. But when a is not true, all bets are off and we don't care whether b is true.

The meaning of a sentence

The meaning of a sentence is the set of interpretations under which the sentence is true.

In propositional logic you could easily write a program to compute the meaning of a sentence defined this way.

Suppose we live in a world with three propositional symbols: P, Q and R.

What are the meanings of the following sentences?

- P
- $P \vee R$ => (Q <=> R)
- True
- False
- P ^ ~P
- $P \vee$ ~P

Validity and Satisfiability

Some sentences are very true! For example:

- True
- P => P
- (P ^ Q) => Q

If you're a sentence whose meaning includes every possible interpretation, then you're a **valid** sentence.

If you're a sentence whose meaning has no interpretation (P ^ ~P, False, etc) then you're an **unsatisfiable** sentence.

If you're a sentence whose meaning has at least one interpretation, you're a **satisfiable** sentence.

- You must be either satisfiable or unsatisfiable but you can't be both.
- If you're valid then you're satisfiable.
- If you're satisfiable then you may or may not be valid.

Meaning is important

We have done a lot of work to define what's a meaning.

Is it worth it?

You bet!

Now we can start talking about ways of automating logical reasoning. We'll use meaning to clearly specify this problem, and to clearly understand what it does.

Knowledge Base

A knowledge base is a collection of sentences, representing our knowledge about the world.

Here's an example knowledge base (using propositional logic)

- AndrewLectures => (TodayIsTuesday ∨ TodayIsThursday)
- ~TodayIsThursday
- TodayIsSaturday => SleepLate
- Rainy => GrassIsWet
- AndrewLectures ∨ TodayIsSaturday
- ~SleepLate

Entailment and the meaning of a knowledge base

The meaning of a knowledge base is the set of all interpretations in which all the sentences in the knowledge base are true.

Entailment:

Given our current knowledge base, we may be able to deduce that other sentences must be true.

(More formally: some other sentences may be true in all the interpretations in which the knowledge base is true).

Define

KB |= S (read as "Knowledge base KB entails S")

to mean "in all interpretations where KB is true, so is S".

Entailment examples

KB =

- AndrewLectures => (TodayIsTuesday ∨ TodayIsThursday)
- ~TodayIsThursday
- TodayIsSaturday => SleepLate
- Rainy => GrassIsWet
- AndrewLectures ∨ TodayIsSaturday
- ~SleepLate.

Then which of these are correct entailments?

KB |= ~SleepLate

KB |= GrassIsWet

KB |= ~SleepLate ∨ GrassIsWet

KB |= TodayIsTuesday

KB |= True

Logical Inference

The logical inference problem is simply stated.

The computer has a knowledge base KB. The user inputs a sentence. The computer tells the user whether the sentence is entailed by the knowledge base.

Given N propositional symbols in the system, can you invent an inference algorithm that costs $O(2^N)$ time complexity?

Efficient Logical Inference

- Humans who are doing proofs almost never do the brute-force approach of the previous slide.
- And we can often arrange that automatic reasoning doesn't either.

A **Proof Theory** gives a set of general purpose rules that allow efficient determination of entailment.

- We use the symbol KB |-- b to mean that our proof theory can infer sentence b from knowledge base KB.

How good is a proof theory?

- The proof theory is **sound** if

  KB|= b whenever KB |-- b

- The proof theory is **complete** if

  KB |-- b whenever KB |= b

Some Inference Rules from a Proof Theory for Propositional Calculus

In these rules, a and b can stand for any sentences.

(1) Modus Ponens or Implication Elimination:

$$a \Rightarrow b,\ a \ \ |\text{--}\ b$$

(2) And-Elimination

$$a_1 \wedge a_2 \wedge .. a_n \ \ |\text{--}\ a_i \quad (\textit{for any } 1 <= i <= n)$$

(3) Unit Resolution

$$a \vee b,\ \sim b \ \ |\text{--}\ a$$

(4) Implication-Elimination

$$a \Rightarrow b \ \ |\text{--}\ \sim a \vee b$$

How do you show the soundness of a proof rule?

Do a truth table. Show that all interpretations in which the left hand side of the |-- is true, the right hand side is also true.

Logical Inference Example

KB =

- AndrewLectures => (TodayIsTuesday ∨ TodayIsThursday)
- ~TodayIsThursday
- TodayIsSaturday => SleepLate
- Rainy => GrassIsWet
- AndrewLectures ∨ TodayIsSaturday
- ~SleepLate

Proof Theory Inference Rules:

(1) Modus Ponens or Implication Elimination:
a => b , a |-- b

(2) And-Elimination
$a_1 \wedge a_2 \wedge .. a_n$ |-- a_i *(for any 1 <= i <= n)*

(3) Unit Resolution
a ∨ b , ~b |-- a

(4) Implication-Elimination
a => b |-- ~a ∨ b

Can we use our rules to infer:

KB |-- ~TodayIsSaturday ?

Logical Inference Example

KB =

- AndrewLectures => (TodayIsTuesday ∨ TodayIsThursday)
- ~TodayIsThursday
- TodayIsSaturday => SleepLate
- Rainy => GrassIsWet
- AndrewLectures ∨ TodayIsSaturday
- ~SleepLate

Proof Theory Inference Rules:

(1) Modus Ponens or Implication Elimination:
a => b , a |-- b

(2) And-Elimination
$a_1 \wedge a_2 \wedge .. a_n$ |-- a_i *(for any 1 <= i <= n)*

(3) Unit Resolution
a ∨ b , ~b |-- a

(4) Implication-Elimination
a => b |-- ~a ∨ b

Can we use our rules to infer:

KB |-- TodayIsTuesday ?

Wumpus Proof

14	24	34	44
13	23	33	43
12 STENCH	22 *Visited, found nothing*	32	42
11 *Visited, found nothing*	21 *Breeze*	31	41

Imagine we're at a stage in the game where we've explored four cells, and had the above experiences. What can we deduce about the Wumpus? Let Sij be a symbol denoting whether there's a stench in cell ij. Let Bij denote whether there's a breeze. Let Wij denote whether there's a wumpus. What's in our knowledge base?

- Our experiences so far.

- The rules of the game

Our experiences so far

14	24	34	44
13	23	33	43
12 STENCH	22 *Visited, found nothing*	32	42
11 *Visited, found nothing*	21 *Breeze*	31	41

(Let Sij be a symbol denoting whether there's a stench in cell ij. Let Bij denote whether there's a breeze. Let W ij denote whether there's a wumpus.)

Our experiences so far:

S12 , ~S11 , ~S21 , ~S22 ,

B21 ~B11 , ~B12 , ~B22

Rules of the game

14	24	34	44
13	23	33	43
12 STENCH	22 *Visited, found nothing*	32	42
11 *Visited, found nothing*	21 *Breeze*	31	41

"If there's no stench in a cell, there is no Wumpus in an adjacent cell". Produces 16 sentences, including:

R1 : ~S11 => ~W11 ^ ~W12 ^ ~W21

R2 : ~S21 => ~W11 ^ ~W21 ^ ~W22 ^ ~W31

R3 : ~S12 => ~W11 ^ ~W12 ^ ~W22 ^ ~W13

"If there's a stench in a cell, there's a wumpus in a neighboring cell." Produces 16 sentences, including:

R4 : S12 => W11 ∨ W12 ∨ W22 ∨ W13

Wumpus Proof

KB =

S12 ,
~S11 ,
~S21 ,
~S22 ,
B21
~B11
~B12
~B22
R1 : ~S11 => ~W11 ^ ~W12 ^ ~W21
R2 : ~S21 => ~W11 ^ ~W21 ^ ~W22 ^ ~W31
R3 : ~S12 => ~W11 ^ ~W12 ^ ~W22 ^ ~W13
R4 : S12 => W11 ∨ W12 ∨ W22 ∨ W13

Inference Rules:

(1) Modus Ponens or Implication Elimination:

$a \Rightarrow b,\ a \ \vdash\ b$

(2) And-Elimination

$a_1 \wedge a_2 \wedge .. a_n \ \vdash\ a_i \quad (\text{for any } 1 \le i \le n)$

(3) Unit Resolution

$a \vee b,\ \sim b \ \vdash\ a$

(4) Implication-Elimination

$a \Rightarrow b \ \vdash\ \sim a \vee b$

Can we prove KB |-- W13 ?

Modus Ponens with ~S11 and R1. And Elimination to Result. Modus Ponens to ~S21, R2 then And-elim to result. Apply Modus ponens to S12 and R4. Apply Unit Resolution with b as W11. Do it again with B as W22. Do it again with b as W12.

Let's moan about propositional logic

Propositional logic cannot express general-purpose knowledge succinctly.

We needed 32 sentences to describe the relation between wumpusses and stenches. We'd need another 32 for pits and breezes.

Then when we describe the effects of actions we'll need at least 64 more. (How would we do that?)

Then how about expressing the fact that there's only one wumpus?