Logical Agents – Lecture 7 (Oggie)

Please read this along with the lecture slides (he actually follows the slides here)

Deductive reasoning – use the example of Wumpus World to illustrate some points

Knowledge Base

- Set of sentences in a formal language – domain specific content

Inference Engine

- Domain-independent
- How to combine different sentences to arrive at new sentences

Thus, gather information → Update the representation of the world → Deduce a new conclusion

Modern example of the Wumpus World PEAS description

- 1972 (Oggie was being sarcastic…)
- There is a grid of cells and you can move from one cell to another
- (Read slide 5 for the rules of the Wumpus game, which is rather simple)
- The goal is to pick up as much gold

So how do you reason in this kind of a maze?

Oggie: Is this intelligence? No the Wumpus is just following the rules dictated by codes

The purpose of the Wumpus example is to make you think about the design (of what Oggie?)

- Observable? No, only local perception
- Deterministic? Yes, outcomes are exactly specified
- Episodic? No, sequential at the level of actions
- Static? Yes, Wumpus and Pits do not move
- Discrete? Yes, you have only set number of squares
- Single-agent? Yes

Exploring a Wumpus world

- From the current state of knowledge, you deduce that the two movements are okay (Please refer to page 14 of the slide)

Slides 14-16

- How you can deduce where Wumpus is (Perhaps no need to read this, it's rather simple)
- You can also deduce where there is a pit

Page 21

- Chances are uniformly distributed
  - i.e. equal chance in each square

Logic in General

- See page 22 for the definition

Logic in general

Logics are formal languages for representing information
such that conclusions can be drawn

Syntax defines the sentences in the language

Semantics define the "meaning" of sentences;
i.e., define truth of a sentence in a world

E.g., the language of arithmetic

$x+2 \geq y$ is a sentence; $x2+y>$ is not a sentence

$x+2 \geq y$ is true iff the number $x+2$ is no less than the number y

$x+2 \geq y$ is true in a world where $x=7,\ y=1$
$x+2 \geq y$ is false in a world where $x=0,\ y=6$

Entailment

Entailment means that one thing **follows from** another:

$KB \models \alpha$ Sequence of percepts entailed Wumpus in [1,3]

Knowledge base KB entails sentence α
if and only if
α is true in all worlds where KB is true

E.g., the KB containing "the Giants won" and "the Reds won"
entails "Either the Giants won or the Reds won"

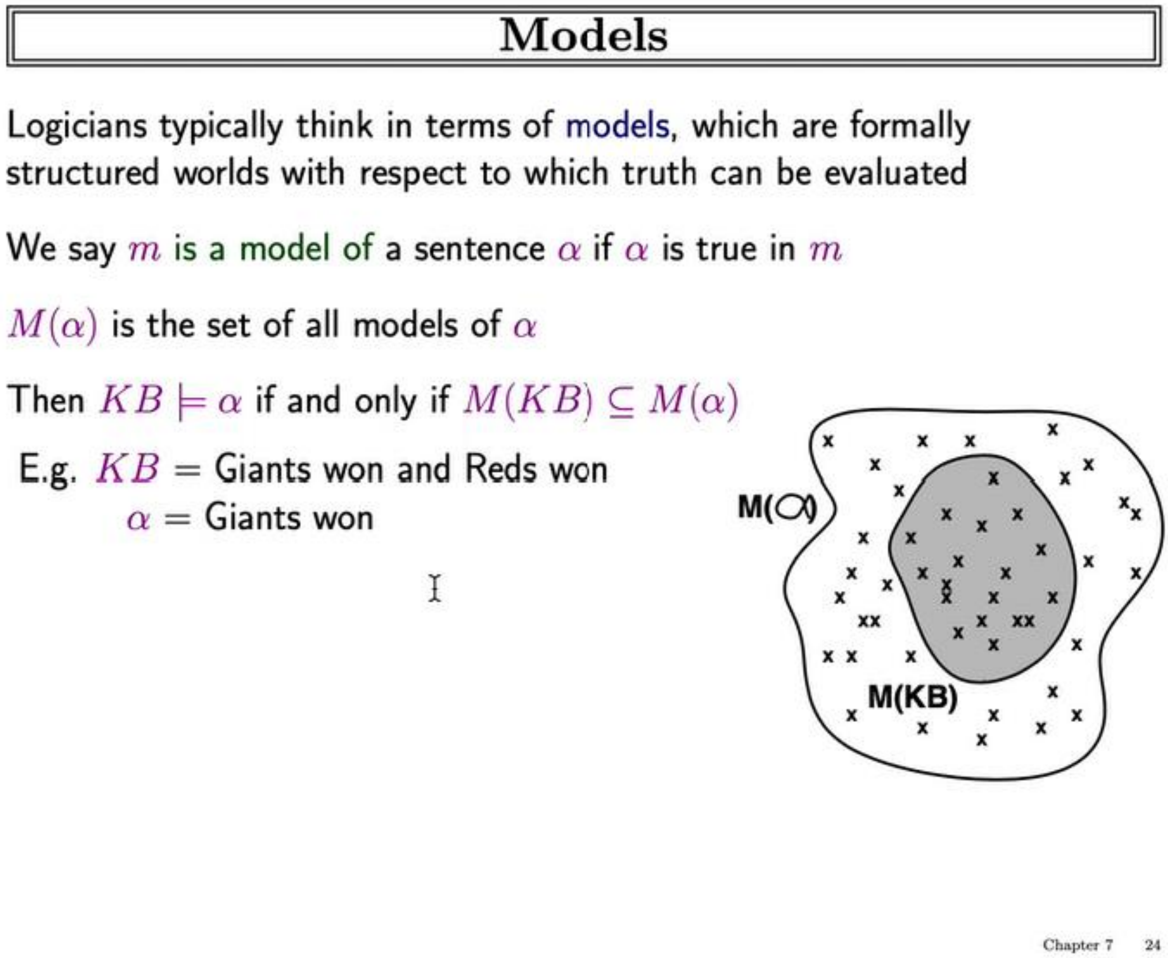
E.g., $x+y=4$ entails $4=x+y$

Entailment is a relationship between sentences (i.e., **syntax**)
that is based on **semantics**

Note: brains process **syntax** (of some sort)

= is syntax. The meaning "equals" is semantics. Could mean string concatenation in javascript, etc.

Oggie was reluctant to explain this slide so most likely not important but here we go:



Entailment in the Wumpus world

Page 27

KB = rule + observations

- Then you can expand this knowledge base with some inference rule or with new perceptions

So what is inference?

Soundness is important

- When we use inference to deduce from the knowledge base
    - It is true that KB entails A

Completeness

- Every time that something is entailed by the knowledge base,
    - It is true that A can be inferred from KB
    - So if the rules are entirely sound, but they can't derive everything, it's incomplete

Inference

$KB \vdash_i \alpha$ = sentence α can be derived from KB by procedure i

Consequences of KB are a haystack; α is a needle.
Entailment = needle in haystack; inference = finding it

Soundness: i is sound if
whenever $KB \vdash_i \alpha$, it is also true that $KB \models \alpha$

Completeness: i is complete if
whenever $KB \models \alpha$, it is also true that $KB \vdash_i \alpha$

Preview: we will define a logic (first-order logic) which is expressive enough to say almost anything of interest, and for which there exists a sound and complete inference procedure.

That is, the procedure will answer any question whose answer follows from what is known by the KB.

Conjunction (and), Disjunction (or), Biconditional (implies in both directions)

Propositional logic: Syntax

Propositional logic is the simplest logic—illustrates basic ideas

The proposition symbols P_1, P_2 etc are sentences

If S is a sentence, $\neg S$ is a sentence (negation)

If S_1 and S_2 are sentences, $S_1 \wedge S_2$ is a sentence (conjunction)

If S_1 and S_2 are sentences, $S_1 \vee S_2$ is a sentence (disjunction)

If S_1 and S_2 are sentences, $S_1 \Rightarrow S_2$ is a sentence (implication)

If S_1 and S_2 are sentences, $S_1 \Leftrightarrow S_2$ is a sentence (biconditional)

Page 33 and 34 are just an elaboration of the previous slide on the definitions.

It's not complicated.

Page 36 is an elaboration of these rules in the Wumpus world sentences. It's quite straightforward and Oggie didn't elaborate anything in addition to the contents of the slide.

Page 37 – Truths tables for inference based on the previous slides

Oggie makes a good point that the real world is extremely complex

- Very selcom can you represent it in truth tables

Truth tables for inference

$B_{1,1}$	$B_{2,1}$	$P_{1,1}$	$P_{1,2}$	$P_{2,1}$	$P_{2,2}$	$P_{3,1}$	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*

Enumerate rows (different assignments to symbols),
if KB is true in row, check that α is too

7 Boolean variables so 2^(7) rows in the truth table
In three of these the KB is true
In those three, there is no pit in [1,2]
So we have derived no pit in [1,2] from the KB by exhaustive case analysis

Oggie skipped page 38 entirely

Page 39

- According to Oggie, the only non-intuitive ones are the De Morgan*
- Which I disagree, I think they are quite intuitive

Page 40

- Oggie basically just read through these
- They are not unimportant though

Reductio ad Absurdum

- Proof by contradiction

Mathematicians don’t like proof by contradiction because it doesn’t really give you much insight as to why something is true.

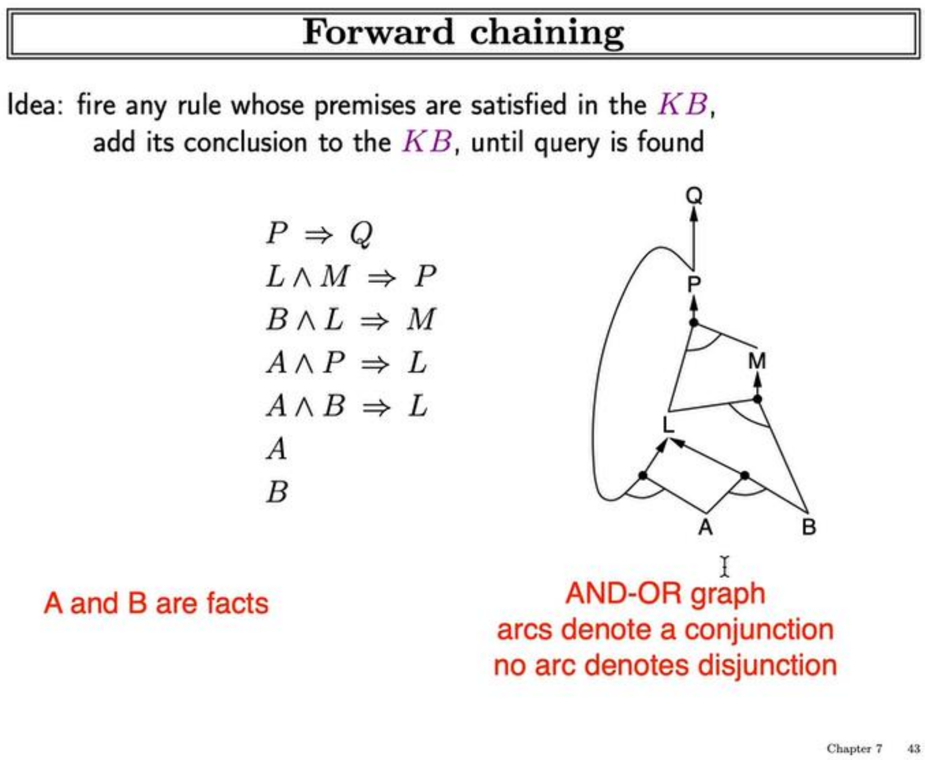
Page 41 – Proof Methods (Oggie sort of skipped this slide, so likely not important)

Modus Ponens is important

- If B implies Q
- If B is true, then we can imply Q
- Sounds very trivial but important

Forward Chaining

- Start with something you know and go forward to derive new things



Forward chaining logic explained from slides p 43 onwards

Need graphical explanation – Oggie didn’t really add anything, the slides are enough

Page 53 when forward chaining is done – Proof of completeness

Proof of completeness

FC derives every atomic sentence that is entailed by KB

atomic means one variable or its negation

1. FC reaches a fixed point where no new atomic sentences are derived
2. Consider the final state as a model m, assigning true/false to symbols
3. Every clause in the original KB is true in m
   Proof: Suppose a clause $a_1 \wedge \ldots \wedge a_k \Rightarrow b$ is false in m
   Then $a_1 \wedge \ldots \wedge a_k$ is true in m and b is false in m
   Therefore the algorithm has not reached a fixed point!
4. Hence m is a model of KB
5. If $KB \models q$, q is true in **every** model of KB, including m

General idea: construct any model of KB by sound inference, check α

FC is data-driven reasoning: start with what we know and build forwards

Exercises 7.4, 7.5, 7.6

Chapter 7 53

Intelligent modelling involves thinking about a theorem first.

And work backwards perhaps…

Rather than starting with axioms

Oggie says