Title: Logical Agents

AIMA: Chapter 6: Sections 6.3 and 6.4

Introduction to Artificial Intelligence

CSCE 476-876, Spring 2001

URL: www.cse.unl.edu/~choueiry/S01-476-876

Berthe Y. Choueiry (Shu-we-ri)

choueiry@cse.unl.edu, (402)472-5444

Glossary 8: *Due date: Mon, March* 26^{th}*, 2001*

Please check also books/dictionnaries/resources other than AIMA

- Antecedent
- Atomic sentence
- Boolean connectives
- Fuzzy logic
- Implication
- Literal
- Predicate logic, (first-order predicate logic)
- Premise
- Propositional logic
- Satisfiable
- Tautology
- Temporal logic
- Truth-preserving (inference procedure)

Glossary 9: *Due date: Mon, April* 2^{nd}*, 2001*

Please check also books/dictionnaries/resources other than AIMA

- Backus-Naur Form (BNR)
- Inference rule
- Locality
- Models (in logic. Sorry, might be difficult)
- Monotonicity
- Conjunctive Normal Form (CNF)
- Disjunctive Normal Form (DNF)
- Horn clauses (sentences)
- Horn Normal Form
- If-then statement

Outline

- Propositional (Boolean) logic
- Inference rules
- Normal forms

Inference

- Example of inference procedure: deduction
- Validity of a sentence: always true (i.e., under all possible interpretations)
  |The Earth is round or not round|
  $\rightarrow$ Tautology
- Satisfiability of a sentence: sometimes true (i.e., $\exists$ some interpretation(s) where it holds)
  |Alex is on campus|
- Insatisfiability of a sentence: never true (i.e., $\nexists$ any interpretation where it holds)
  |The Earth is round and the earth is not round|
  $\rightarrow$ useful for refutation, as we will see later

Beauty of inference:
Formal inference allows the computer to derive **valid** conclusions even when the computer does not know the interpretation you are using

A logic consists of:

1. A formal representation system:
   (a) Syntax: how to make sentences
   (b) Semantics: systematics constraints on how sentences relate to the states of affairs
2. Proof theory: a set of rules for deducing the entailment of a set of sentences

Example:

$\surd$ Propositional logic (or Boolean logic)

$\surd$ First-order logic FOL

Syntax of Propositional Logic

Propositional logic is the simplest logic—illustrates basic ideas

- Symbols represent whole propositions, sentences
  D says the Wumpus is dead
  The proposition symbols P_1, P_2, etc. are sentences

- Boolean connectives: $\wedge, \vee, \neg, \Rightarrow, \Leftrightarrow$, connect sentences
  If S_1 and S_2 are sentences, the following are sentences too:
  $\neg S_1$, $\neg S_2$, $S_1 \wedge S_2$, $S_1 \vee S_2$, $S_1 \Rightarrow S_2$, $S_1 \Leftrightarrow S_2$

Formal grammar of Propositional Logic: Backus-Naur Form, check Figure 6.8 page 167 in AIMA

Terminology

Atomic sentence: single symbol

Complex sentence: contains connectives, parentheses

Literal: atomic sentence or its negation (e.g., P, $\neg Q$)

Sentence $(P \land Q) \Rightarrow R$ is an implication, conditional, rule, if-then statement

$(P \land Q)$ is a premise, antecedent

R is a conclusion, consequence

Sentence $(P \land Q) \Leftrightarrow R$ is an equivalence, biconditional

Precedence order resolves ambiguity (highest to lowest):

$\neg, \land, \lor, \Rightarrow, \Leftrightarrow$ (e.g., $\neg P \lor Q \land R \Rightarrow S$?)

Syntax of First-order logic (Chapter 7)

First-Order Logic (FOL) is expressive enough to say almost anything of interest and has a sound and complete inference procedure

- Logical symbols:
  - parentheses
  - connectives ($\neg, \Rightarrow$, the rest can be regenerated)
  - variables
  - equality symbol (optional)
- Parameters:
  - quantifier $\forall$
  - predicate symbols
  - constant symbols
  - function symbols

Semantics of Propositional Logic

Semantics is defined by specifying:

— Interpretation of a proposition symbols and constants (T/F)

— Meaning of logical connectives

Proposition symbol means what ever you want:

D says the Wumpus is dead

Breeze says the agent is feeling a breeze

Stench says the agent is perceiving an unpleasant smell

Connectives are functions: complex sentences meaning derived from the meaning of its parts

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*

Note:

$P \Rightarrow Q$: if P is true, Q is true, otherwise I am making no claim

Models (I)

A model is a world in which a sentence is true under a particular interpretation

Logicians typically think in terms of <u>models</u>, which are formally <u>structured</u> worlds with respect to which truth can be evaluated

We say m is a <u>model</u> of a sentence α if α is true in m

Typically, a sentence can be true in many models

Models (II)

$M(\alpha)$ is the set of all models of α

Entailment: A sentence α is entailed by a KB if the models of the KB are all models ofα

KB $\models \alpha$ iff all models of KB are models of α (*i.e.*, $M(KB) \subseteq M(\alpha)$)

Then KB $\models \alpha$ if and only if $M(KB) \subseteq M(\alpha)$

Models (III)

In propositional logic:

- A model is a mapping from proposition symbols directly to truth or falsehood
- The models of a sentence are the mappings that make the sentence true

Example:

	α:	obj1 $\wedge$ obj2
$\surd$	Model1:	$1 \wedge 1$
$\surd$	Model2:	$1 \vee 1$
$\times$	Model3:	$0 \wedge 1$

Enumeration method in Propositional Logic

Let $\alpha = A \vee B$ and $KB = (A \ \vee C) \wedge (B \vee \neg C)$

Is it the case that KB $\models \alpha$?

Check all possible models—α must be true wherever KB is true

A	B	C	$A \vee C$	$B \vee \neg C$	KB	α
F	F	F				
F	F	T				
F	T	F				
F	T	T				
T	F	F				
T	F	T				
T	T	F				
T	T	T				

Validity and inference

A sentence is valid if it is true in all models
e.g., $A \vee \neg A$, $A \Rightarrow A$, $(A \wedge (A \Rightarrow B)) \Rightarrow B$

To establish validity, use truth tables:

P	H	$P \vee H$	$(P \vee H) \wedge \neg H$	$((P \vee H) \wedge \neg H) \Rightarrow P$
False	*False*	*False*	*False*	*True*
False	*True*	*True*	*False*	*True*
True	*False*	*True*	*True*	*True*
True	*True*	*True*	*False*	*True*

If every row is true, then then the conclusion, P, is entailed by the premises, $((P \vee H) \wedge \neg H)$

Validity is connected to inference via the Deduction Theorem:
KB $\models \alpha$ iff (KB $\Rightarrow \alpha$) is valid

Satisfiability and inference

A sentence is satisfiable if it is true in some model
e.g., $A \lor B$, C

A sentence is unsatisfiable if it is true in no models
e.g., $A \land \neg A$

Satisfiability is connected to inference via the following:
KB $\models \alpha$ iff (KB $\land \neg\alpha$) is unsatisfiable
i.e., prove α by *reductio ad absurdum*

Proof methods

Proof methods divide into (roughly) two kinds:

Model checking

Truth-table enumeration (sound and complete for propositional)

Heuristic search in model space (sound but incomplete)

e.g., the GSAT algorithm (Exercise AIMA 6.15)

Application of inference rules

Legitimate (sound) generation of new sentences from old

<u>Proof</u> = a sequence of inference rule applications, can use inference rules as operators in a standard search algorithm

Inference rules for Propositional logic (I)

- Modus Ponens (Implication-Elimination)

$$\frac{\alpha \Rightarrow \beta, \quad \alpha}{\beta}$$

- And-Elimination

$$\frac{\alpha_1 \wedge \alpha_2 \wedge \ldots \wedge \alpha_n}{\alpha_i}$$

- And-Introduction

$$\frac{\alpha_1, \alpha_2, \ldots, \alpha_n}{\alpha_1 \wedge \alpha_2 \wedge \ldots \wedge \alpha_n}$$

- Or-Introduction

$$\frac{\alpha_1, \alpha_2, \ldots, \alpha_n}{\alpha_1 \vee \alpha_2 \vee \ldots \vee \alpha_n}$$

Inference rules for Propositional logic (II)

- Double-Negation Elimination

$$\frac{\neg\neg\alpha}{\alpha}$$

- Unit resolution

$$\frac{\alpha \vee \beta, \quad \neg\beta}{\alpha}$$

- Resolution: $\frac{\alpha\vee\beta, \quad \neg\beta\vee\gamma}{\alpha\vee\gamma}$ or equivalently $\frac{\neg\alpha\Rightarrow\beta, \quad \beta\Rightarrow\gamma}{\neg\alpha\Rightarrow\gamma}$

Soundness of an inference rule

Premises are true $\rightarrow$ conclusion is true

To verify soundness, construct a truth table

Resolution rule: $\frac{\alpha\vee\beta,\quad \neg\beta\vee\gamma}{\alpha\vee\gamma}$ or equivalently $\frac{\neg\alpha\Rightarrow\beta,\quad \beta\Rightarrow\gamma}{\neg\alpha\Rightarrow\gamma}$

α	β	γ	$\alpha \vee \beta$	$\neg\beta \vee \gamma$	$\alpha \vee \gamma$
False	*False*	*False*	*False*	*True*	*False*
False	*False*	*True*	*False*	*True*	*True*
False	*True*	*False*	*True*	*False*	*False*
False	*True*	*True*	*True*	*True*	*True*
True	*False*	*False*	*True*	*True*	*True*
True	*False*	*True*	*True*	*True*	*True*
True	*True*	*False*	*True*	*False*	*True*
True	*True*	*True*	*True*	*True*	*True*

The job of an inference procedure is to construct proofs by finding appropriate sequences of applications of inference rules

starting with sentences initially in KB

and culminating in the generation of the sentence whose proof is desired

Complexity of propositional inference

Truth-table : Complete. 2^n rows: exponential, thus impractical

Inference rules : NP-complete in general

Two special cases:

1. Horn sentences: $P_1 \wedge P_2 \wedge \ldots P_n \Rightarrow Q$, where P_i and Q are non-negated atoms
   Apply Modus Ponens repeatedly: complete and polynomial time.

2. Use locality and exploit monotonicity
   Locality: Modus Ponens
   Monotonicity: if $\mathrm{KB}_1 \models \alpha \Rightarrow KB_1 \cup KB_2 \models \alpha$
   PL, FOL are monotonic, probability theory is not
   Monotonicity essential for soundness of inference

Wumpus world in Propositional Logic

Sentences:

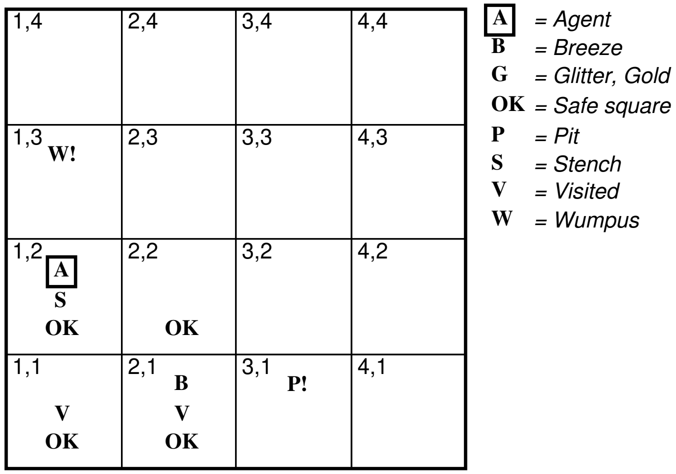
$S_{i,j}$: there is a stench in square $[i, j]$

$\neg S_{i,j}$: there is not a stench in square $[i, j]$

$B_{i,j}$: there is a breeze in square $[i, j]$...

$W_{i,j}$: there is a Wumpus in square $[i, j]$...

Wumpus world starting position



Sentence of KB:

$$\neg S_{1,1},\ \neg S_{2,1},\ S_{1,2},\ \neg B_{1,1},\ B_{2,1},\ B_{1,2},\ \text{etc.}$$

$$R_1:\quad \neg S_{1,1} \quad \Rightarrow \neg W_{1,1} \wedge \neg W_{1,2} \wedge \neg W_{2,1}$$

$$R_2:\quad \neg S_{2,1} \quad \Rightarrow \neg W_{1,1} \wedge \neg W_{2,1} \wedge \neg W_{2,2} \wedge \neg W_{3,1}$$

$$R_3:\quad \neg S_{1,2} \quad \Rightarrow \neg W_{1,1} \wedge \neg W_{1,2} \wedge \neg W_{2,2} \wedge \neg W_{1,3}$$

$$R_4:\quad S_{1,2} \quad \Rightarrow W_{1,3} \vee W_{1,2} \vee W_{2,2} \vee \neg W_{1,1}$$

etc.

Agent infers $W_{1,3}$

Process: First show Wumpus is not in one of the other squares, then conclude by elimination that it must be in [1, 3].

1. Apply Modus Ponens with $\neg S_{1,1}$ and R_1
2. Apply And-elimination to result: three separate sentences
3. Apply Modus Ponens with $\neg S_{2,1}$ and R_2, then apply And-elimination to result
4. Apply Modus Ponens with $\neg S_{1,2}$ and R_4
5. Apply Unit resolution rule to result of (4) and $\neg W_{1,2}$, derived in (2)
6. Apply unit resolution again with $\neg W_{2,2}$, derived in (3)
7. Apply unit resolution again with $\neg W_{1,2}$, derived in (2)

Agent uses inference to act

Add all rules that relate current state of the world to the actions agent should take

Example: $A_{1,1} \wedge \text{East}_A \wedge W_{1,2} \Rightarrow \neg forward$

Propositional logic:

- can answer: 'Should I go forward?', 'Should I go right?'
- cannot answer: 'What should I do?'

```
function PROPOSITIONAL-KB-AGENT(percept) returns an action
   static: KB, a knowledge base
           t, a counter, initially 0, indicating time

   TELL(KB, MAKE-PERCEPT-SENTENCE(percept, t))
   for each action in the list of possible actions do
       if ASK(KB, MAKE-ACTION-QUERY(t, action)) then
           t ← t + 1
           return action
   end
```

Problems with Propositional Logic

- good vehicle for introducing what logic / inference is
- too weak to even handle the Wumpus world
  'Don't go forward when Wumpus is in front of you' requires 64 rules (16 squares x 4 orientations for agent)
- Problem: generating rules, handling truth tables (2^n rows for n symbols)!
- Important problem: How to represent change when agent moves from [1, 1] to [1, 2]?
  Solution: Time-stamp symbols :-(
  But (1) length of game is not known in advance, (2) rewrite a time-dependent version of each rule..

Problem: PL only allows propositions, no relations, no objects

Normal forms: Other approaches to inference use syntactic operations on sentences, often expressed in standardized forms

Conjunctive Normal Form (CNF)

conjunction of $\underbrace{\textit{disjunctions}\text{ of }\textit{literals}}_{\textit{clauses}}$

E.g., $(A \vee \neg B) \wedge (B \vee \neg C \vee \neg D)$

Disjunctive Normal Form (DNF)

disjunction of $\underbrace{\textit{conjunctions}\text{ of }\textit{literals}}_{\textit{terms}}$

E.g., $(A \wedge B) \vee (A \wedge \neg C) \vee (A \wedge \neg D) \vee (\neg B \wedge \neg C) \vee (\neg B \wedge \neg D)$

Horn Form

conjunction of *Horn clauses* (clauses with ≤ 1 positive literal)

E.g., $(A \vee \neg B) \wedge (B \vee \neg C \vee \neg D)$

Often written as set of implications: $B \Rightarrow A$ and $(C \wedge D) \Rightarrow B$

Summary

Logical agents apply inference to a knowledge base to derive new information and make decisions

Basic concepts of logic:

— **Syntax**: formal structure of **sentences**

— **Semantics**: **truth** of sentences WRT **models**

— **Entailment**: necessary truth of one sentence given another

— **Inference**: deriving sentences from other sentences

— **Soundness**: derivations produce only entailed sentences

— **Completeness**: derivations can produce all entailed sentences

Truth-table method is sound and complete for propositional logic

Wumpus world requires the ability to represent partial and negated information, reason by cases, etc.

Propositional logic suffices for some of these tasks