Logical Agents

September 14, 2004

The aim of AI is to develop intelligent agents that can reason about actions and their effects and about the environment, create plans to achieve a goal, execute the plans, observe the changes, and change the course of actions if necessary to attain or maintain a goal. To achieve this goal machines need knowledge and the capability to reason with it. Until machines can acquire knowledge by themselves, we need to feed them with knowledge and teach them how to reason with the knowledge. Thus, we need to be able to represent knowledge of interest to the agent such as

- information about the environment
- information about its actions (capability)

and provide it a means to reason with this knowledge. One could argue that natural language (for instance, English) could be used for representing knowledge and our commonsense reasoning could do the job helping the agents in reasoning. As machines are still far from understanding natural language, this will be the dream of mankind for years to come. Furthermore, no formal theory has been proved to be able to capture commonsense reasoning in full.

This forces us to resort to a *formal language* that computers can understand in full and the reasoning process can be carried out in such a language. This constitutes a *logic*. Propositional logic is perhaps the simplest logic ever developed.

1 Knowledge-Based Agents

The components of a knowledge-based agent are its *knowledge base* (KB) and an *inference engine* that can be used to derive answer(s) to questions posed to the KB (ASK) or to add new knowledge into the KB (TELL).

The agent, with its KB and inference engine, will decide upon *what to do*, given its perception about the world. The main loop for this agent is

```
function KB-Agent(percept) return an action
  static KB - the knowledge base
         t  - the time moment
  TELL(KB,Make-Percept-Sentence(percept, t)
  action <- ASK(KB, Make-Action-Query(t))
```

```
TELL(KB,Make-Action-Sentence(action, t)
t <- t+1
return action
```

2 Propositional Logic

2.1 Syntax

Describe how to construct sentences.

Definition 1 (Alphabet) *An* alphabet *of a propositional logic theory (or propositional theory, for short) is a set of symbols consisting of:*

- proposition symbols*: like P, Q etc. each represent a proposition,*
- constant symbols*: $True$ and $False$*
- logical connectives*: $\wedge, \vee, \Leftrightarrow, \Rightarrow, \neg$, and parentheses $($, $)$.*

While the first item will change from theory to theory, the last two items will be the same for every propositional theory. Instead of proposition symbols and constant symbols we will shorten to propositions and constants, respectively, if no confusion is possible.

Definition 2 (Sentence) *Given an alphabet, a* sentence *is defined as follows.*

- *Each constant symbol ($True$ or $False$) is an atomic sentence.*
- *Each proposition symbol is an atomic sentence.*
- *If P, Q are sentence then so does $P \wedge Q$, $P \vee Q$, $P \Leftrightarrow Q$, $P \Rightarrow Q$, $\neg P$. This sentences are called non-atomic sentences.*
- *If P is an atomic sentence then (P) is an atomic sentence.*
- *If P is a non-atomic sentence then (P) is also a non-atomic sentence.*

Name	Symbol	Example	Name of part
$\wedge$	Conjunction/And	$P \wedge Q$	P, Q - conjuncts
$\vee$	Disjunction/Or	$P \vee Q$	P, Q - disjuncts
$\Rightarrow$	Implication	$P \Rightarrow Q$	P - premises , Q - conclusion P - antecedent , Q - consequent
$\Leftrightarrow$	Equivalence	$(P \vee Q) \Leftrightarrow (Q \vee P)$	-
$\neg$	Negation	$\neg P$	-

Example 1 *Some atomic propositions:*

- P *– (This can represents the fact “Every student works hard”)*

- Q – *(This can represents the fact "Tom is a student")*
- R – *(This can represents the fact "Tom works hard")*

Some non-atomic sentences:

- $(P \wedge Q) \Rightarrow S$ – *(This could be understood as the implication: "Every student works hard and Tom is a student" implies "Tom works hard")*
- $(P \wedge Q) \Rightarrow (\neg S)$ – *("Every student works hard and Tom is a student" implies "Tom does not work hard")*

Definition 3 *A* theory *is a set of sentences.*

2.2 Semantics

Specify how sentences relate to states of affair. (The meaning of sentences in the world that is described by the propositions). This is done by specifying the interpretation of the symbols belonging to the alphabet.

Definition 4 (Interpretation) *An* interpretation I *of an alphabet is a mapping from its set of proposition symbols to the set of truth values* $\{True, False\}$*, i.e., for each symbol* P*,* $I(P) = True$ *or* $I(P) = False$*.*

In every interpretation I of T, the logical constant $True$ and $False$ is assigned the truth value $True$ and $False$, respectively.

Definition 5 (Truth Value of Sentences) *Given an alphabet and* I *be an interpretation of it. The truth value of a non-atomic sentence* S *is defined using the following rules:*

- $S = P \wedge Q$, $I(S)$ is $True$ if $I(P) = I(Q) = True$; otherwise, $I(S) = False$.
- $S = P \vee Q$, $I(S)$ is $True$ if $I(P) = True$ or $I(Q) = True$; otherwise, $I(S) = False$.
- $S = P \Leftrightarrow Q$, $I(S)$ is $True$ if $I(P) = I(Q)$; otherwise, $I(S) = False$.
- $S = P \Rightarrow Q$, $I(S)$ is $True$ if $I(P) = False$ or $I(Q) = True$; otherwise, $I(S) = False$.
- $S = \neg P$, $I(S)$ is $True$ if $I(P) = False$; otherwise, $I(S) = False$.

We say that S *is true with respect to* I *if* $I(S) = True$*.*

The truth value of a sentence can also be determined using the following table.

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

2.3 Model, Validity, and Entailment

Given a theory T and an interpretation I.

Definition 6 (Model - Satisfiability) *An interpretation I is a* model *of T if $I(s) = True$ for every sentence $s \in T$.*

T is satisfiable if T has a model.

Definition 7 (Validity) *A sentence S over an alphabet A is* valid *if S is true with respect to every possible interpretation I of A.*

Definition 8 (Entailment) *A sentence S is* entailed *by a theory T, denoted by $T \models S$ if every model M of T is also a model of S.*

Naive algorithm for checking satisfiability of T. Enumerate all possible interpretations, check one-by-one if any interpretation is a model of T. If one is, return 'Yes'; if none is, return 'No'.

Modify the above algorithm for checking entailment?

NOTE. A theory could be viewed as a sentence (a conjunction of all the sentences belonging to it).

2.4 Inference rules for propositional logic

An *inference rule* is of the form

$$\frac{\alpha}{\beta}$$

says that the sentence β can be derived from a sentence α denoted by $\alpha \vdash \beta$. It is established to make the inference process easier.

Following are inference rules for propositional logic.

- **Modus Ponens**

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

- **And-Elimination**

$$\frac{\alpha \wedge \beta}{\beta}$$

- **And-Introduction**

$$\frac{\alpha, \beta}{\alpha \wedge \beta}$$

- **Or-Introduction**

$$\frac{\alpha}{\alpha \vee \beta}$$

- **Double-Negation Elimination**

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

- **Unit Resolution**

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

- **Resolution**

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

2.5 Basic Definitions and Problems in Propositional Logic

Given a theory T and a sentence S.

Satisfiability Problem. Does T has a model?

Entailment Problem. Determining whether T entails S (or $T \models S$)? This allows us to generate new 'True' sentences (e.g. S) from the old 'True' sentences (e.g. sentences of T)

Inference Procedure. A procedure that, given a theory T, generates new sentences that are entailed by T; or answers the query whether $T \models S$ for every sentence S.

An inference procedure is *sound* if it generates only entailed sentences.

A sequence of operations (or inference rules) that is used by a sound inference procedure of T in showing that $T \models S$ is called a *proof* of S (wrt. T).

An inference procedure is *complete* if it can find a proof for any sentence that is entailed by the theory.

Complexity. Satisfiability is NP-complete. Entailment is co-NP complete.

Example 2 Checking the validity of a sentence:

$$P \wedge (Q \wedge R) \Leftrightarrow (P \wedge Q) \wedge R$$

Using truth table:

P	Q	R	$P \wedge Q$	$Q \wedge R$	$P \wedge (Q \wedge R)$	$(P \wedge Q) \wedge R$
T	T	T	T	T	T	T
T	T	F	T	F	F	F
T	F	T	F	F	F	F
T	F	F	F	F	F	F
F	T	T	F	T	F	F
F	T	F	F	F	F	F
F	F	T	F	F	F	F
F	F	F	F	F	F	F

Observe that whenever left-hand-side (LHS) is true, so is right-hand-side (RHS) and whenever LHS is false, so is RHS (and vice versa). So, LHS $\Rightarrow$ RHS.

Using inference rules:

$$\frac{P \wedge (Q \wedge R)}{P} \quad \text{and} \quad \frac{P \wedge (Q \wedge R)}{Q \wedge R} \quad \text{(And-Elimination) (1)}$$

$$\frac{Q \wedge R}{Q} \quad \text{and} \quad \frac{Q \wedge R}{R} \quad \text{(And-Elimination) (2)}$$

(1) and (2) imply that P, Q, and R can be derived from $P \wedge (Q \wedge R)$. (3)

$$\frac{P, Q}{(P \wedge Q)} \quad \text{(And-Introduction) (4)}$$

$$\frac{(P \wedge Q), R}{(P \wedge Q) \wedge R} \quad \text{(And-Introduction) (5)}$$

(1)-(5) show that

$$P \wedge (Q \wedge R) \Rightarrow (P \wedge Q) \wedge R$$

Similarly, we can show that

$$(P \wedge Q) \wedge R \Rightarrow P \wedge (Q \wedge R)$$

The last two implications show the equivalence.