

# Module 5

## Knowledge Representation and Logic – (Propositional Logic)

# Lesson 12

## Propositional Logic inference rules

## 5.5 Rules of Inference

Here are some examples of sound rules of inference. Each can be shown to be sound once and for all using a truth table. The left column contains the premise sentence(s), and the right column contains the derived sentence. We write each of these derivations as  $A \vdash B$ , where A is the premise and B is the derived sentence.

Name	Premise(s)	Derived Sentence
Modus Ponens	$A, A \Rightarrow B$	B
And Introduction	$A, B$	$A \wedge B$
And Elimination	$A \wedge B$	A
Double Negation	$\sim\sim A$	A
Unit Resolution	$A \vee B, \sim B$	A
Resolution	$A \vee B, \sim B \vee C$	$A \vee C$

In addition to the above rules of inference one also requires a set of equivalences of propositional logic like " $A \wedge B$ " is equivalent to " $B \wedge A$ ". A number of such equivalences were presented in the discussion on propositional logic.

## 5.6 Using Inference Rules to Prove a Query/Goal/Theorem

A proof is a sequence of sentences, where each sentence is either a premise or a sentence derived from earlier sentences in the proof by one of the rules of inference. The last sentence is the query (also called goal or theorem) that we want to prove.

Example for the "weather problem" given above.

1.	Q	Premise
2.	$Q \Rightarrow P$	Premise
3.	P	Modus Ponens(1,2)
4.	$(P \wedge Q) \Rightarrow R$	Premise
5.	$P \wedge Q$	And Introduction(1,3)
6.	R	Modus Ponens(4,5)

### 5.6.1 Inference vs Entailment

There is a subtle difference between entailment and inference.

**Inference ( $\vdash$ ):** Given 2 sentences  $p$  and  $q$  we say  $q$  is inferred from  $p$ , written  $p \vdash q$ , if there is a sequence of rules of inference that apply to  $p$  and allow  $q$  to be added.

Notice that inference is not directly related to truth; i.e. we can infer a sentence provided we have rules of inference that produce the sentence from the original sentences.

However, if rules of inference are to be useful we wish them to be related to entailment. Ideally we would like:

$$p \vdash q \quad p \models q$$

iff

but this equivalence may fail in two ways:

- $p \vdash q \quad p \not\models q$   
but

We have inferred  $q$  by applying rules of inference to  $p$ , but there is some model in which  $p$  holds but  $q$  does not hold. In this case the rules of inference have inferred "too much".

- $p \models q \quad p \not\vdash q$   
but

$q$  is a sentence which holds in all models in which  $p$  holds, but we cannot find rules of inference that will infer  $q$  from  $p$ . In this case the rules of inference are insufficient to infer the things we want to be able to infer.

## 5.7 Soundness and Completeness

These notions are so important that there are 2 properties of logics associated with them.

**Soundness:** An inference procedure  $\vdash$  is sound if whenever  $p \vdash q$  then it is also the case that  $p \models q$ .

"A sound inference procedure infers things that are valid consequences"

**Completeness:** An inference procedure  $\vdash$  is complete if whenever  $p \models q$  then it is also the case that  $p \vdash q$ .

“A complete inference procedure is able to infer anything that is that is a valid consequence”

The “best” inference procedures are both sound and complete, but gaining completeness is often computationally expensive. Notice that even if inference is not complete it is desirable that it is sound.

*Propositional Logic and Predicate Logic each with Modus Ponens as their inference produce are sound but not complete.* We shall see that we need further (sound) rules of inference to achieve completeness. In fact we shall see that we shall even restrict the language in order to achieve an effective inference procedure that is sound and complete for a subset of First Order Predicate Logic.

The notion of soundness and completeness is more generally applied than in logic. Whenever we create a knowledge based program we use the syntax of the knowledge representation language, we assign a semantics in some way and the reasoning mechanism defines the inference procedures. The semantics will define what entailment means in this representation and we will be interested in how well the reasoning mechanism achieves entailment.

### 5.7.1 Decidability

Determining whether  $p \models q$  is computationally hard. If  $q$  is a consequence then if  $\vdash$  is complete then we know that  $p \vdash q$  and we can apply the rules of inference exhaustively knowing that we will eventually find the sequence of rules of inference. It may take a long time but it is finite.

However if  $q$  is not a consequence (remember the task is *whether or not*  $p \models q$ ) then we can happily apply rules of inference generating more and more irrelevant consequences. So the procedure is guaranteed to eventually stop if  $q$  is derivable, but may not terminate otherwise.

**Decidability:** A problem is decidable if there is a procedure that is guaranteed to terminate having determined whether the answer is “yes” or “no”.

**Semi-Decidability:** A *problem* is only *semi-decidable* if there is a procedure that is guaranteed to terminate in one of these cases but not both.

Entailment in Propositional Logic is decidable since truth tables can be applied in a finite number of steps to determine whether or not  $p \models q$ .

Entailment in Predicate Logic is only semi-decidable; it is only guaranteed to terminate when  $q$  is a consequence. One result of this semi-decidability is that many problems are not decidable; if they rely on failing to prove some sentence. Planning is typically not decidable. A common reaction to a non-decidable problem is to *assume* the answer after some reasoning time threshold has been reached. Another reaction to the semi-decidability of Predicate Logic is to restrict attention to subsets of the logic; however even if its entailment is decidable the procedure may be computationally expensive.

## Questions

1. Consider a knowledge base  $KB$  that contains the following propositional logic sentences:

$$\begin{aligned} Q &\Rightarrow P \\ P &\Rightarrow \neg Q \\ Q \vee R \end{aligned}$$

- a) Construct a truth table that shows the truth value of each sentence in  $KB$  and indicate the models in which the  $KB$  is true.
- b) Does  $KB$  entail  $R$ ? Use the definition of entailment to justify your answer.
- c) Does  $KB$  entail  $R \Rightarrow P$ ? Extend the truth table and use the definition of entailment to justify your answer.
- d) Does  $KB$  entail  $Q \Rightarrow R$ ? Extend the truth table and use the definition of entailment to justify your answer.

2. Using propositional logic prove (D) from (A,B,C):

- A.  $P \Rightarrow (Q \Leftrightarrow R)$
- B.  $\neg(Q \Leftrightarrow R)$
- C.  $(S \wedge Q) \Rightarrow P$
- D.  $\neg P \wedge (S \Rightarrow \neg Q)$

## Solution

1.a. Truth table:

P	Q	R	$Q \Rightarrow P$	$P \Rightarrow \neg Q$	$Q \vee R$	$R \Rightarrow P$	$Q \Rightarrow R$
T	T	T		F			
T	T	F		F			F
T	F	T	Model in which KB is true				T
T	F	F			F		
F	T	T	F			F	
F	T	F	F				F
F	F	T	Model in which KB is true				T
F	F	F			F		

1.b.  $KB \models R$ . In every model in which the KB is true (indicated in the table), R is also true; thus the KB entails R.

1.c. The KB does not entail P. When (P,Q,R) is (F,F,T),  $R \Rightarrow P$  is false.

1.d.  $KB \models (Q \Rightarrow R)$ . In every model in which the KB is true,  $Q \Rightarrow R$  is also true.

2. The proof steps are described below.

Transforming A,B,C and the negation of D to CNF gives the following clauses:

A.1  $\neg P \vee \neg Q \vee R$ .

A.2  $\neg P \vee Q \vee \neg R$ .

B.1  $Q \vee R$

B.2  $\neg Q \vee \neg R$

C.  $\neg S \vee \neg Q \vee P$ .

D.1  $P \vee S$ .

D.2  $P \vee Q$ .

Resolution proof may then proceed as follows:

E.  $Q \vee \neg R$  (D.2 + A.2, factored)

F.  $Q$ . (E + B.1)

G.  $\neg R$  (F+B.2).

H.  $\neg Q \vee P$ . (C+D.1, factored)

I.  $P$  (H+D.2, factored)

J.  $\neg Q \vee R$  (I+A.1).

K.  $R$  (J+F)

L.  $\emptyset$ . (K+G).