Math 481 Lecture 1 — Syntax of Propositional Logic — 9/8/09

1 Introduction

In propositional logic we assume that we have a collection of elementary sentences that we can think of as expressing propositions. For example, the English sentence "Bucky is a badger." expresses the proposition that Bucky is a badger, which so happens to be true. We will not analyze the internal structure of these elementary sentences, that is left for a later time for predicate logic. For now, we will call these *atomic propositions*, since they are left unanalyzed, and use single letters, called *propositional symbols*, to represent them. All that matters for us is that the elementary statements express propositions which are either true or false, but never both.

The *syntax* of propositional logic describes the ways in which these elementary propositions are combined to form more complex propositions, those whose form it is the business of propositional logic to analyze. The construction procedures we consider will be ones basic to mathematical texts, and the operations that combine propositions to form new ones are called *propositional connectives*. The connectives one finds most frequently in a mathematical text are "and", "or", "implies", "if and only if" ("iff") and "not". The meaning given to these by the working mathematician does not precisely reflect the range of meanings they take in everyday discourse; they have been fixed so as to become unambiguous.

The formal symbols for these connectives:

(a) $\neg$ for "not" (*negation*) ,

(b) $\wedge$ for "and" (*conjunction*) ,

(c) $\vee$ for "or" (*disjunction*) ,

(d) $\rightarrow$ for "implies" (*conditional*),

(e) $\leftrightarrow$ for "if and only if" (*biconditional*) .

We will make these meanings precise when we describe the *semantics* of propositional logic. For now, we will describe how to use them to construct complex propositions from atomic propositions. There are also be two atomic propositions whose meaning will be fixed:

(f) $\top$ for "the true"] (*true*),

(g) $\leftrightarrow$ for "the false"] (*false*) .

2 Formal Syntax of Propositional Logic

Definition 1.2.1 The *language of propositional logic* consists of the following symbols:

(i) *Connectives*: $\top, \bot$(constants), $\neg$ (unary), $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$ (binary);

(ii) *Propositional Symbols*: $P, Q, R, P_0, Q_0, R_0, \ldots,$, a countably infinite collection;

(iii) *Punctuation*: (,) (left and right parentheses).

Definition 1.2.2 A (propositional) atom is a propositional symbol, $\top$ or $\bot$.

Now that the symbols of the language are specified, we can describe the how to construct propositions by providing an *inductive definition*.

Definition 1.2.3 The set of propositions is the smallest set **PROP** such that

1. Every propositional atom is in **PROP**.
2. If α is in **PROP**, then so is $(\neg\alpha)$.
3. If α and β are in **PROP**, then so are each of $(\alpha \wedge \beta)$, $(\alpha \vee \beta)$, $(\alpha \rightarrow \beta)$, $(\alpha \leftrightarrow \beta)$.

The inductive definition can also be described using production rules, which might be easier to apply then the inductive definition. The variable *prp* ranges over propositions.

$$\begin{array}{llll} (1) & prp & \Rightarrow & \top \mid \bot \mid P \qquad \text{for each propositional symbol } P \\ (2) & prp & \Rightarrow & (\neg\, prp) \\ (3) & prp & \Rightarrow & (prp \wedge prp) \mid (prp \vee prp) \mid (prp \rightarrow prp) \mid (prp \leftrightarrow prp) \end{array}$$

A grammar rule such as

$$prp \Rightarrow (\neg\, prp)$$

means that one instance of the variable *prp* may be replaced by the right-side. Rules of the form (1) and (2), separated by $\mid$, means that we may replace by one instance of *prp* by any one of the $\mid$ separated sequences of symbols. For example, (3) is just an abbreviation of four production rules:

$$prp \Rightarrow (prp \wedge prp) \qquad prp \Rightarrow (prp \vee prp) \qquad \text{etc.}$$

Example 1.2.4 The following are correctly formed expressions

$$\begin{array}{c} P, \qquad (Q \vee R), \qquad (\neg(Q \vee R)) \\ ((P \rightarrow Q) \leftrightarrow ((\neg P) \vee Q), \qquad ((Q \wedge R) \rightarrow Q) \\ (((Q \rightarrow R) \wedge (\neg R)) \rightarrow (\neg Q)), \qquad (P \vee (\neg P)) \end{array}$$

One the other hand, the following are not correctly formed

$$(()), \qquad (P \wedge Q)\vee, \qquad \neg P \wedge Q$$

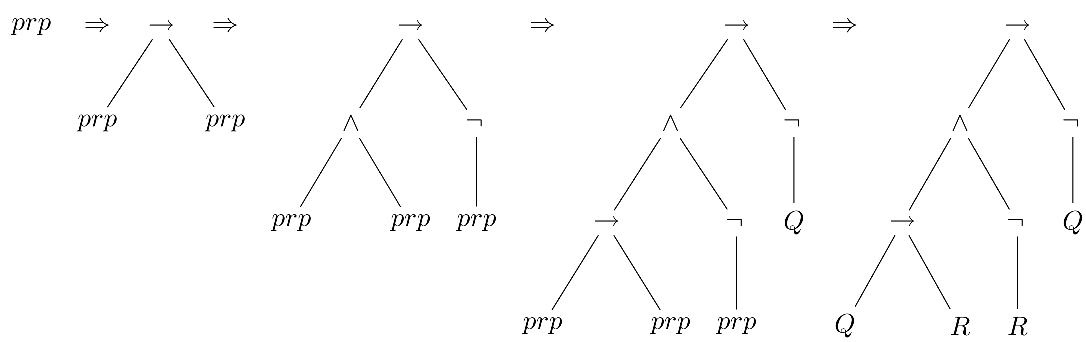
Lets show that $(((Q \rightarrow R) \wedge (\neg R)) \rightarrow (\neg Q))$ is correctly formed by deriving it using the production rules.

$$\begin{array}{l} prp \overset{(3)}{\Rightarrow} (prp \rightarrow prp) \overset{(3)}{\Rightarrow} ((prp \wedge prp) \rightarrow prp) \\ \overset{(3)}{\Rightarrow} (((prp \rightarrow prp) \wedge prp) \rightarrow prp) \overset{(1,1)}{\Rightarrow} (((Q \rightarrow R) \wedge prp) \rightarrow prp) \\ \overset{(2,1)}{\Rightarrow} (((Q \rightarrow R) \wedge (\neg R)) \rightarrow prp) \overset{(2,1)}{\Rightarrow} (((Q \rightarrow R) \wedge (\neg R)) \rightarrow (\neg Q)) \end{array}$$

It useful to represent propositions as *formation trees.* These are each generated from a derivations as follows:

1. Each application of rule (1) labels a node with a propositional atom with no immediate successors. (So, the leaves are the propositional atoms.)
2. Each application of rule (2) labels a node with $\neg$ and has one immediate successor.
3. Each application of rule (3) labels a node with a binary symbol $\diamond$ and has immediate successors.

The formation tree grows downward as each rule is applied.



Now that we have given precise conditions for the propositional language, we relax those conditions for the ease of the reader.

Convention 1.2.5 (omitting parentheses) In order to ease the reader's burdens by minimizing the number of parentheses when naming propositions we adopt several conventions. These conventions are still unambiguous as any linguist would attest.

1. We will drop the outermost parentheses. For example, $P \wedge Q$ refers to $(P \wedge Q)$.
2. We adopt an order of *precedence* to the logical connectives. Starting from highest precedence to lowest"

$$\neg \quad \wedge \quad \vee \quad \rightarrow \quad \leftrightarrow .$$

   For example,

$$\begin{array}{rcl} \neg P \wedge Q & \text{is} & ((\neg P) \wedge Q) \\ P \vee Q \wedge R & \text{is} & ((P \vee Q) \wedge R) \\ P \wedge Q \rightarrow R \vee Q & \text{is} & ((P \wedge Q) \rightarrow (R \vee Q)) \end{array}$$

3. When one connective symbol is used repeatedly, grouping is right associative. For example, for any binary connective $\diamond$,

$$P \diamond B \diamond R \quad \text{is} \quad (P \diamond (B \diamond R))$$

3 Justifying Inductive Definition

Remark 1.3.1 The definition of **PROP** is an example of an *inductive definition.* There are two equivalent ways of justifying the existence of an inductively defined set like **PROP**.

- (Top-Down) **PROP** is the unique smallest set meeting conditions (1) through (3), containing the propositional atoms (1) and closed under the constructions in (2) and (3). The existence of this set is easily established. First, there are sets which meet the conditions (1) through (3), the set of all strings formed from the symbols does. Second, the intersection of any family of sets which meet the conditions (1) through (3) also meets these conditions. Now, let **PROP** be the intersection of all sets meeting conditions (1) through (3). This will be the smallest such set.
- (Bottom-Up I) We can provide a more explicit definition of **PROP**: it is the set of all expressions which can be derived in *finitely* many steps from the production rules starting with *prp*. More formally, we first define a *derivation sequence of length k* by induction on the natural numbers:
  1. The only derivation sequence of length one is $\langle prp \rangle$.

2. If σ is a derivation sequence of length k, α is the last term of the sequence and $\alpha \Rightarrow \beta$ by one application of a production rule (1), (2) or (3), then $\sigma\widehat{\ }\beta$ is a derivation sequence of length $k+1$.

Now, α is a proposition if for some k, there is a derivation sequence σ of length k, α is the last term in the sequence and α contains no instances of the variable *prp*. Notice that the shortest derivation sequence for a proposition α the number of nodes in the formation tree for α (which is also the number of non-parenthesis symbols in α).

- (Bottom-Up II) Derivations are fine, but even a shortest derivation is not uniquely determined (since there are variations in order of steps). A third approach, which is more convenient, is to define propositions by their *depth*:

  1. The propositions of *depth no more than 0* are the propositional atoms.
  2. If α is a proposition whose depth is no more than n, then $(\neg\alpha)$ is a proposition whose *depth is no more than* $n+1$.
  3. If α and β are propositions whose depth is no more than n, then $(\alpha \diamond \beta)$ is a proposition whose *depth is no more than* $n+1$, for each $\diamond \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$.

  Then α is a proposition if α is a proposition whose depth is no more than n for some n. The *depth* of α is the smallest such n. The depth of a proposition (as defined here) is exactly the same as the depth of the formation tree, where this is the longest path from root to leaf. Notice the depth of a proposition α is always greater than the depth of its immediate constituents: the depth of $(\neg\alpha)$ is greater than the depth of α, the depth of $(\alpha \diamond \beta)$ is greater than the depth of α and the depth of β.

It is not apparent that these three ways of defining **PROP** are equivalent. This is discussed in Appendix 2.

What is important about inductively defined sets is that they give us access to a method of proof (the Principle of Structural Induction) and a method of definition (the Principle of Structural Recursion). More on these next time