Formula trees and other notations for propositional logic

Dirk Schlimm

September 30, 2017 (updated version)

1 Terminology

1.1 Formula trees

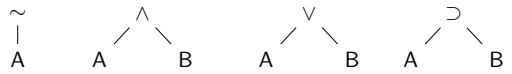
Definition: A *language* $\mathcal{L}$ of propositional (or sentential) logic consists of

- a set $\mathrm{PVar}(\mathcal{L})$ of *propositional variables*: P_0, P_1, $\mathsf{P}_2, \ldots$,
- a set $\mathrm{Conn}_1(\mathcal{L})$ of *unary logical connectives*, containing: $\sim$ (negation),
- a set $\mathrm{Conn}_2(\mathcal{L})$ of *binary logical connectives*, containing: $\wedge$ (conjunction), $\vee$ (disjunction), and $\supset$ (implication).[1]

If we do not need to distinguish between the *arity* of the connectives, we can just speak of the set $\mathrm{Conn}(\mathcal{L})$ of *logical connectives*, such that $\mathrm{Conn}(\mathcal{L}) = \mathrm{Conn}_1(\mathcal{L}) \cup \mathrm{Conn}_2(\mathcal{L})$ (assuming that the sets $\mathrm{Conn}_1(\mathcal{L})$ and $\mathrm{Conn}_2(\mathcal{L})$ are disjoint).

Definition: A *formula tree* is defined by the following recursive definition:

1. *Base clause:* Any propositional variable P_i is a formula tree. These are called *atomic* formula trees.
2. *Inductive clauses:* If A and B are formula trees, so are:



3. *Final clause:* Only trees that are obtained by the above rules are formula trees.

Definition: If a formula tree is not a single propositional variable, then the top connective of a formula tree is called the *main connective*. This is also called the *root* or the *top node* of the tree.

Definition: If the main connective is *unary*, we also say that the tree is unary; if the main connective of a formula tree is binary, then we say that the tree is *binary*.

Definition: For the unary tree in the above inductive clauses, A is the *subtree*. For the binary trees in the above inductive clauses, A is the *left subtree* and B is the *right subtree*.

[1] Alternative symbols used in the literature: & (conjunction), $\rightarrow$ (implication), and $\neg$ or $\overline{\ }$ (negation).

1.2 Truth values

Definition: A *truth value assignment* is a mapping $f : \mathrm{PVar}(\mathcal{L}) \rightarrow \{\mathsf{T}, \mathsf{F}\}$.

In other words, f maps each propositional variable P_i to one of the truth values T or F.

Definition: The *truth value* of a formula tree A under a truth value assignment f is defined as follows:

- *Case 1 (Atomic formula tree):*

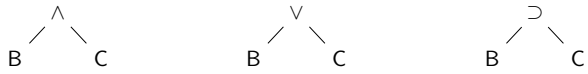
  If A is a propositional variable, then the truth value of A is $f(\mathsf{A})$.

- *Case 2 (Unary formula tree):* If A is of the form

  then the truth value of A is determined by the following truth table as depending on the truth value of B:

B	A
T	F
F	T

- *Case 3 (Binary formula tree):* If A is of the form



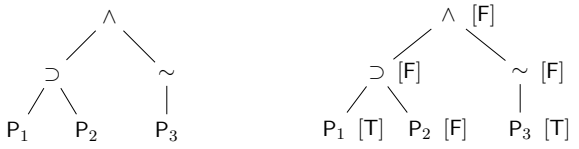
  then the truth value of A is determined depending on the truth values of B and C by these truth tables, respectively:

B	C	A
T	T	T
T	F	F
F	T	F
F	F	F

B	C	A
T	T	T
T	F	T
F	T	T
F	F	F

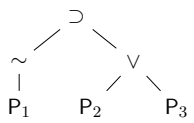
B	C	A
T	T	T
T	F	F
F	T	T
F	F	T

Example: We can annotate the nodes in a formula tree with truth values to determine the truth value of a tree, beginning with the truth value assignment of the leaf nodes, i. e., of the propositional variables. Say we have a truth value assignment f that maps P_1 to T, P_2 to F, and P_3 to T, i. e., $f(\mathsf{P}_1) = \mathsf{T}$, $f(\mathsf{P}_2) = \mathsf{F}$, and $f(\mathsf{P}_3) = \mathsf{T}$.
Under this assignment, the truth value of the following formula tree is F, as the annotated formula tree on the right shows.

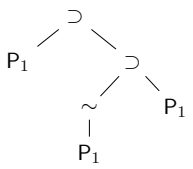


Exercise 1: If a truth value assignment f maps P_1 to T, P_2 to F, and P_3 to T, then what are the truth values of the following three formula trees?

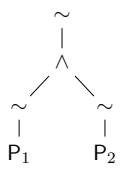
a)



b)



c)



1.3 Translating formula trees into other notations

There are three main ways of traversing a tree in order to write it in a linear notation:[2]

Prefix, preorder, Polish	Infix, inorder, algebraic	Postfix, postorder, endorder
Visit the root	Traverse the left subtree	Traverse the left subtree
Traverse the left subtree	Visit the root	Traverse the right subtree
Traverse the right subtree	Traverse the right subtree	Visit the root

A unary node, with only one subtree, will be considered to have an empty left subtree (and its subtree as 'right' subtree).

1.3.1 Infix, inorder, or algebraic

The three formula trees from Exercise 1 are written as *infix*, *inorder*, or *algebraic* formulas as follows:

$$\sim \mathsf{P}_1 \supset (\mathsf{P}_2 \vee \mathsf{P}_3) \qquad \mathsf{P}_1 \supset (\sim \mathsf{P}_1 \supset \mathsf{P}_1) \qquad \sim(\sim \mathsf{P}_1 \wedge \sim \mathsf{P}_2)$$

Note, that for each left subtree that is traversed an open parenthesis '(' has to be added and each time a right subtree has been traversed a close parenthesis ')' has to be added.

1.3.2 Prefix, preorder, or Polish

The three formula trees from Exercise 1 are written as *prefix*, *preorder*, or *Polish* formulas as follows (notice that no parentheses are necessary in prefix and postfix representations):

$$\supset \sim \mathsf{P}_1 \vee \mathsf{P}_2 \, \mathsf{P}_3 \qquad \supset \mathsf{P}_1 \supset \sim \mathsf{P}_1 \, \mathsf{P}_1 \qquad \sim \wedge \sim \mathsf{P}_1 \sim \mathsf{P}_2$$

1.3.3 Postfix, postorder, or endorder

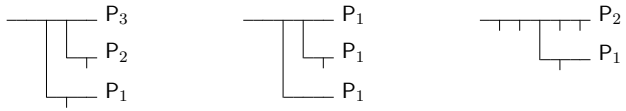
The three formula trees from Exercise 1 are written as *postfix*, *postorder*, or *endorder* formulas as follows:

$$\mathsf{P}_1 \sim \mathsf{P}_2 \, \mathsf{P}_3 \vee \supset \qquad \mathsf{P}_1 \mathsf{P}_1 \sim \mathsf{P}_1 \supset\supset \qquad \mathsf{P}_1 \sim \mathsf{P}_2 \sim \wedge \sim$$

[2] More information can be found in Section 2.3 'Trees' of Donald Knuth, *The Art of Computer Programming*, Vol. 1, 3rd ed. Addison-Wesley, Reading MA (pp. 308–423).

1.3.4 Frege's Begriffsschrift notation

A variant of the prefix representaion is Frege's two-dimensional *Begriffsschrift* notation, in which the three formula trees from Exercise 1 are written as:



(Compare this with the formula tree representation.)

1.3.5 Dot notation by Peano/Whitehead and Russell

A variant of the infix notation is the *dot notation* used by Peano, and Whitehead and Russell, where the three formula trees from Exercise 1 are written as follows:

$$\sim \mathsf{P}_1 . \supset . \mathsf{P}_2 \vee \mathsf{P}_3 \qquad \mathsf{P}_1 \supset . \sim \mathsf{P}_1 \supset \mathsf{P}_1 \qquad \sim : \sim \mathsf{P}_1 . \wedge . \sim \mathsf{P}_2$$

Exercise 2: Write the formula tree from the Example on the bottom of p. 2 in infix, prefix, and postfix notation. (Try also to write them in Frege's Begriffsschrift notation and in Peano's dot notation.)