Chapter I

Preliminaries

§ 0. Foreword on Trees

Trees will play an important role throughout this work, so we shall commence with some pertinent definitions:

By an *unordered tree*, $\mathcal{T}$, we shall mean a collection of the following items:

(1) A set S of elements called *points*.

(2) A function, ℓ, which assigns to each point x a positive integer $\ell(x)$ called the *level* of x.

(3) A relation xRy defined in S, which we read "x is a *predecessor* of y" or "y is a *successor* of x". This relation must obey the following conditions:

C_1: There is a unique point a_1 of level 1. This point we call the *origin* of the tree.

C_2: Every point other than the origin has a unique predecessor.

C_3: For any points x, y, if y is a successor of x, then $\ell(y)=\ell(x)+1$.

We shall call a point x an *end point* if it has no successors; a *simple point* if it has exactly one successor, and a *junction point* if it has more than one successor. By a *path* we mean any finite or denumerable sequence of points, beginning with the origin, which is such that each term of the sequence (except the last, if there is one) is the predecessor of the next. By a *maximal path* or *branch* we shall mean a path whose last term is an end point of the tree, or a path which is infinite.

It follows at once from C_1, C_2, C_3 that for any point x, there exists a unique path P_x whose last term is x. If y lies on P_x, then we shall say that y *dominates* x, or that x is *dominated* by y. If x dominates y and $x \neq y$, then we shall say that x is (or lies) *above* y, or that y lies *below* x. We shall say that x is *comparable* with y if x dominates y or y dominates x. We shall say that y is *between* x and z if y is above one of the pair $\{x,z\}$ and below the other.

By an *ordered tree*, $\mathcal{T}$, we shall mean an unordered tree together with a function θ which assigns to each junction point z a sequence $\theta(z)$ which contains no repetitions, and whose set of terms consists of all the successors of z. Thus, if z is a junction point of an ordered tree, we can speak of the $1^{st}, 2^{nd}, \ldots, n^{th}, \ldots$ successors of z (for any n up to the number of successors of z) meaning, of course, the $1^{st}, 2^{nd}, \ldots, n^{th}, \ldots$ terms of $\theta(z)$.

For a simple point x, we shall also speak of the successor of x as the *sole* successor of x.

We shall usually display ordered trees by placing the origin at the top and the successor(s) of each point x below x, and in the order, from left to right, in which they are ordered in the tree. And we draw a line segment from x to y to signify that y is a successor of x.

We shall have occasion to speak of adding "new" points as successors of an end point x of a given tree $\mathcal{T}$. By this we mean more precisely the following: For any element y outside $\mathcal{T}$, by the adjunction of y as the sole successor of x, we mean the tree obtained by adding y to the set S, and adding the ordered pair $\langle x,y\rangle$ to the relation R (looked at as a set of ordered pairs), and extending the function ℓ by defining $\ell(y)=\ell(x)+1$. For any distinct elements $y_1,\ldots,y_n$, each outside S, by the adjunction of $y_1,\ldots,y_n$ as respective $1^{\mathrm{st}},2^{\mathrm{nd}},\ldots,n^{\mathrm{th}}$ successors of x, we mean the tree obtained by adding the y_i to S, adding the pairs $\langle x,y_i\rangle$ to R and extending ℓ by setting $\ell(y_1)=\ldots=\ell(y_n)=\ell(x)+1$, and extending the function θ by defining $\theta(x)$ to be the sequence $(y_1,\ldots,y_n)$. [It is obvious that the extended structure obtained is really a tree].

A tree is called *finitely generated* if each point has only finitely many successors. A tree, $\mathcal{T}$, is called *finite* if $\mathcal{T}$ has only finitely many points, otherwise the tree is called *infinite*. Obviously, a finitely generated tree may be infinite.

We shall be mainly concerned with ordered trees in which each junction point has exactly 2 successors. Such trees are called *dyadic* trees. For such trees we refer to the first successor of a junction point as the *left successor*, and the second successor as the *right successor*.

[*Exercise:* In a dyadic tree, define x to be to the left of y if there is a junction point whose left successor dominates x and whose right successor dominates y. Prove that if x is to the left of y and y is to the left of z, then x is to the left of z].

§ 1. Formulas of Propositional Logic

We shall use for our undefined logical connectives the following 4 symbols:

(1) $\sim$ [read "not"], (2) $\wedge$ [read "and"],
(3) $\vee$ [read "or"], (4) $\supset$ [read "implies"].

These symbols are respectively called the *negation, conjunction, disjunction*, and *implication* symbols. The last 3 are collectively called *binary* connectives, the first ($\sim$) the *unary* connective.

Other symbols shall be:

(i) A denumerable set $p_1,p_2,\ldots,p_n,\ldots$ of symbols called *propositional variables*.

(ii) The two symbols (,), respectively called the *left parenthesis* and the *right parenthesis* (they are used for purposes of punctuation). Until we come to *First-Order Logic*, we shall use the word "variable" to mean *propositional variable.*

We shall use the letters "p", "q", "r", "s" to stand for any of the variables $p_1,p_2,\ldots,p_n,\ldots$. The notion of *formula* is given by the following recursive rules, which enable us to obtain new formulas from those already constructed:

F_0: Every propositional variable is a formula.
F_1: If A is a formula so is $\sim A$.
F_2,F_3,F_4: If A, B are formulas so are $(A\wedge B)$, $(A\vee B)$, $(A\supset B)$.

This recursive definition of "formula" can be made explicit as follows. By a *formation sequence* we shall mean any finite sequence such that each term of the sequence is either a propositional variable or is of the form $\sim A$, where A is an earlier term of the sequence, or is of one of the forms $(A\wedge B)$, $(A\vee B)$, $(A\supset B)$, where A, B are earlier terms of the sequence. Now we can define A to be a *formula* if there exists a formation sequence whose last term is A. And such a sequence is also called a *formation sequence* for A.

For any formula A, by the *negation* of A we mean $\sim A$. It will sometimes prove notationally convenient to write A' in place of $\sim A$. For any 2 formulas A, B, we refer to $(A\wedge B)$, $(A\vee B)$, $(A\supset B)$ as the *conjunction, disjunction, conditional* of A, B respectively. In a conditional formula $(A\supset B)$, we refer to A as the *antecedent* and B as the *consequent.*

We shall use the letters "A", "B", "C", "X", "Y", "Z" to denote formulas. We shall use the symbol "b" to denote any of the binary connectives $\wedge$, $\vee$, $\supset$; and when "b" respectively denotes $\wedge$, $\vee$, $\supset$ then $(X\,b\,Y)$ shall respectively mean $(X\wedge Y)$, $(X\vee Y)$, $(X\supset Y)$. We can thus state the formation rules more succinctly as follows:

F_0: Every propositional variable is a formula.
F_1: If X is a formula so is $\sim X$.
F_2: If X, Y are formulas, then for each of the binary connectives b, the expression $(X\,b\,Y)$ is a formula.

In displaying formulas by themselves (i.e. not as parts of other formulas), we shall omit outermost parentheses (since no ambiguity can result). Also, for visual perspicuity, we use square brackets [ ] interchangeably with parentheses, and likewise braces { }. Usually we shall use square brackets as exterior to parentheses, and braces as exterior to square brackets.

Example. Consider the following formula:

$$((((p \supset q) \wedge (q \vee r)) \supset (p \vee r)) \supset \sim(q \vee s)).$$

It is easier to read if displayed as follows:

$$\{[(p \supset q) \wedge (q \vee r)] \supset (p \vee r)\} \supset \sim(q \vee s).$$

Biconditional – we use "$X \leftrightarrow Y$" as an abbreviation for $(X \supset Y) \wedge (Y \supset X)$. The formula $X \leftrightarrow Y$ is called the *biconditional* of X, Y. It is read "X if and only if Y" or "X is equivalent to Y".

Uniqueness of Decomposition. It can be proved that every formula can be formed in only one way—i. e. for every formula X, one *and only one* of the following conditions holds:

(1) X is a propositional variable.

(2) There is a *unique* formula Y such that $X = Y'$.

(3) There is a *unique* pair X_1, X_2 and a *unique* binary connective b such that $X = (X_1 b X_2)$.

Thus no conjunction can also be a disjunction, or a conditional; no disjunction can also be a conditional. Also none of these can also be a negation. And, e. g., $(X_1 \wedge X_2)$ can be identical with $(Y_1 \wedge Y_2)$ only if $X_1 = Y_1$ and $X_2 = Y_2$ (and similarly with the other binary connectives). We shall not prove this here; perfectly good proofs can be found, e. g. in Church [1] or Kleene [1].

In our discussion below, we shall consider a more abstract approach in which this combinatorial lemma can be circumvented.

***Discussion.** First we wish to mention that some authors prefer the following formation rules for formulas:

F'_0: Same as F_0.

F'_1: If X is a formula, so is $\sim(X)$.

F'_2: If X, Y are formulas, so is $(X)b(Y)$.

This second set of rules has the advantage of eliminating, at the outset, outermost parentheses, but has the disadvantage of needlessly putting parentheses around variables.

It seems to us that the following set of formation rules, though a bit more complicated to state, combines the advantages of the two preceding formulations, and involves using neither more nor less parentheses than is necessary to prevent ambiguity:

F''_0: Same as before.

F''_1: If X is a formula but not a propositional variable and p is a propositional variable, $\sim(X)$ and $\sim p$ are formulas.

F''_2: If X, Y are both formulas, but neither X nor Y is a propositional variable, and if p, q are propositional variables, then the following expressions are all formulas:

(a) $(X)b(Y)$,
(b) $(X)bq$,
(c) $pb(Y)$,
(d) pbq.

In all the above 3 approaches, one needs to prove the unique decomposition lemma for many subsequent results. Now let us consider yet another scheme (of a radically different sort) which avoids this.

First of all, we delete the parentheses from our basic symbols. We now define the *negation* of X, not as the symbol $\sim$ followed by the first symbol of X, followed by the second symbol of X, etc. but simply as the *ordered pair* whose first term is "$\sim$" and whose second term is X. And we define the *conjunction* of X, Y as the *ordered triple* whose first term is X, whose second term is "$\wedge$" and whose third term is Y. [In contrast, the conjunction of X and Y, as previously defined, is a sequence of $n+m+3$ terms, where n, m are the respective number of terms of X, Y. The "3" additional terms are due to the left parenthesis, right parenthesis and "$\wedge$"]. Similarly we define the disjunction (conditional) of X, Y as the ordered triple $\langle X, b, Y \rangle$ where b is the binary connective in question.

Under this plan, a formula is either a (propositional) variable, an ordered pair (if it is a negation) or an ordered triple. Now, no ordered pair can also be an ordered triple, and neither one can be a single symbol. Furthermore, an ordered pair uniquely determines its first and second elements, and an ordered triple uniquely determines its first, second and third elements. Thus the fact that a formula can be formed in "only one way" is now immediate.

We remark that with this plan, we can (and will) still *use* parentheses to describe formulas, but the parentheses are *not* parts of the formula. For example, we write $X \wedge (Y \vee Z)$ to denote the ordered triple whose first term is X, whose second term is "$\wedge$", and whose third term is itself the ordered triple whose first, second and third terms are respectively, Y, $\vee$, Z. But (under this plan) the parentheses themselves do not belong to the object language[1]) but only to our metalanguage[1]).

The reader can choose for himself his preferred notion of "formula", since subsequent developments will not depend upon the choice.

[1]) The term *object language* is used to denote the language talked *about* (in this case the set of formal expressions of propositional logic), and the term *metalanguage* is used to denote the language in which we are talking about the object language (in the present case English augmented by various common mathematical symbols).