Derivations in the Mist: Trees and Natural Deduction

1. Now that we know that Tableau is a complete system of proof, it would be useful to connect the tree system of proof with natural deduction systems more commonly taught in introductory logic courses. For if we could show that from any closed tree, we could construct a natural deduction proof of the same conclusion from the same premises, then our knowledge that the tree system is complete could be extended to natural deduction systems as well. That is what we shall proceed to do.

2. Our recipe for generating the derivation we want will have us proceeding through the tree in what is called a “pre-order traversal.” We begin at the root and then systematically descend (or “push”) through the left-most path until that path closes. We then “pop” back up the path to the most proximate branching point and push through the right-branch, once again pushing through left-hand branches as we descend. Whenever we reach a closed point, we simply “pop” back up and push through the right branch of the most proximate branching point for which the right-hand branch hasn’t been traversed. As we traverse the tree, we will construct a derivation that includes (and thereby makes available) all of the formulas on the left-hand side of the tree, as well as the negations of all the formulas on the right-hand side of the tree. It will also include several more formulas to boot.

3. We begin our derivation by listing all of the premises. They of course also appear on the left-hand side of the root of our tree proof. We then assume the negation of the conclusion for indirect proof (IP), which appears on the right-hand side of the root of our tree. We then show how all of the formulas that ever appear on the left hand side of a path in the tree and the negations of all of the formulas that ever appear along the right hand side of a path may either be derived or assumed.

4. Every branch in the tree proof will correspond to the introduction of an indirect subproof inside our derivation, in which we assume the formula (or its negation) that goes along with the left branch. The eventual closure of that node’s rightmost subordinate branch will then correspond to the discovery of a contradiction corresponding to that assumption. Our procedure for traversing the tree guarantees that any path subordinate to a formula in the tree will also have that formula (or its negation) available in the derivation. Since branch closure is a product of one and the same formula appearing both on the left and the right sides of a path, it will correspond to the availability of both a formula and its negation in our derivation.

5. There will be one more path through the tree than there are total branching points. The eventual closure of the right-most branch of the tree will thus correspond to the discovery of a contradiction corresponding to the initial assumption of the negation of the conclusion. The final line of our derivation will thus be the desired conclusion justified by indirect proof (negation elimination).
6. We now go rule-by-rule, showing how we may derive all the formulas (or their negations, if on the right) corresponding to the development of formulas along subordinate paths in the tree.

Non-branching rules:

(1) Negation:

If \( \phi \) is the product of developing \( \neg \phi \) on the left, then we needn’t do anything. By hypothesis, \( \neg \phi \) is already available to us at this point in the derivation. If on the other hand, \( \phi \) is the product of developing \( \neg \phi \) on the right, then by hypothesis, \( \neg \neg \phi \) will be available to us at this point in the derivation. Simple derive \( \phi \) through DN.

(2) Conjunction on the right:

If \( \phi \) is the product of developing a conjunction on the left, then it will be on the left, and by hypothesis, it will be a conjunct of an available conjunction. Simply derive it through conjunction elimination.

(3) Disjunction on the right:

If \( \phi \) is the product of developing a disjunction on the right, then it will also be on the right. By hypothesis, it will be a disjunct of an available disjunction that is negated. Simply derive the negation of that disjunction through DeMorgan’s and conjunction elimination.

(4) Conditional on the right:

When a conditional on the right is developed, we place the antecedent on the left and the consequent on the right of a subordinate path. By hypothesis, we will have the negation of that conditional available to us at that point in our derivation. We may thus derive the antecedent and the negation of the consequent through the definition of the conditional, DeMorgan’s, conjunction elimination, and DN (to the antecedent).
Branching Rules:

(5) Disjunction on the left:

At the point of the branching, a disjunction will be available in our derivation. Each of the branches will be headed by one of the disjuncts on the left. Assume the left-most disjunct for IP. After that branch closes completely, derive that disjunct’s negation through negation introduction. At this point, both the disjunction and the left-most disjunct’s negation will be available in our derivation. Derive the right-most disjunct through disjunctive syllogism.

(6) Conjunction on the right:

At the point of the branching, the negation of a conjunction will be available in our derivation. Each of the branches will be headed by one of the conjuncts on the right. Assume the negation of the left-most conjunct for IP. After that branch closes completely, derive that conjunct’s double negation through negation introduction. At this point, both the negation of the conjunction and the left-most conjunct’s double negation will be available in our derivation. Derive the negation of the right-most conjunct through DeMorgan’s and disjunctive syllogism.

(7) Conditional on the Left:

At the point of the branching, a conditional will be available in our derivation. One branch will be headed by the antecedent on the right, the other by the consequent on the left. Assume the negation of the antecedent for IP. After that branch closes completely, derive the antecedent through negation elimination. At this point, both the conditional and its antecedent will be available in our derivation. Derive the right-most disjunct through modus ponens.

7. That’s about all there is to it! Now take a look at a few closed tree proofs, and try to follow this recipe to construct corresponding natural deduction derivations.

Exercises: Construct tree-proofs for the following entailments, and then use those trees to construct corresponding natural deduction proofs. You may use any of the “derived rules” mentioned in the
procedure outlined above.

1. \((P \rightarrow T), (\neg T \rightarrow \neg R), (\neg P \rightarrow R) \models T\)
2. \(P \models (Q \lor \neg Q)\)
3. \(((M \rightarrow O) \rightarrow M) \models M\)
4. \((P \lor Q), (\neg(P \land R), (\neg(Q \land S)) \models (\neg R \land S)\)
5. \((P \lor Q), (\neg P \lor R), (Q \rightarrow S) \models (\neg R \land S)\)
6. \((L \rightarrow (M \lor N)), \neg M, \neg N \models \neg L\)
7. \(A, (D \lor E), (D \rightarrow (B \land C)), (E \rightarrow (B \land \neg C)) \models (A \land B)\)
8. \((P \rightarrow (Q \lor R)), (R \rightarrow (P \rightarrow S)), (\neg S \land P) \models (P \rightarrow Q)\)