Reduction Trees

While truth tables allow us to determine the validity or invalidity of arguments by searching for counter-examples, they are limited because as they grow in size, the chance of error increases and the time required to construct and read them becomes increasingly prohibitive. Reduction trees give us a more efficient way of exploring the possible substitution instances of an argument form. They also provide a different and more comprehensible picture of the structure of the argument.

Reduction tree construction begins by listing the premises of the argument. The denial of the conclusion is then added as the last item in the list. If the argument is valid, then it is impossible to assert the premises along with the denial of the conclusion without contradiction. The process of developing the rest of the tree involves uncovering the contradictions that must occur if the argument is valid.

Let's first consider a simple argument, one whose premises and conclusion consist only of truth functionally simple statements.

It is cloudy.
<u>It is raining.</u>
Therefore it is cloudy.

While this argument is ridiculously simple, it will illustrate the principle behind the reduction tree method. Using "C" and "R" as statement constants, we list the premises and the denial of the conclusion.

1.	C	premise
2.	R	premise
3.	~C	denial of conclusion

Examining the list, we see that it contains both a statement (C) and the denial of that statement (~C). The list is contradictory. One cannot assert the premises and the denial of the conclusion without contradiction. Thus the argument is valid.

When an argument contains, either as a premise or as the conclusion, a compound statement, the tree gets more complicated. The compound statements must be analyzed into component parts until contradictions are identified or all statements are either truth functionally simple or are the negations of truth functionally simple statements. Let's begin with an argument containing a conjunction.

<u>1. A & B</u>
∴ A

The argument is obviously valid, but we can't successfully construct a tree without some way of handling the premise.

RULE OF CONJUNCTION: List the conjuncts at each open branch in the path containing the conjunction, and check the line number of the conjunction.

Setting up the tree, and applying the rule of conjunction, gives us the following.

√	1.	A & B	premise
	2.	~A	denial of conclusion
	3.	A	from 1, conjunction
	4.	B	
		x	

The "x" below line 4 indicates that the lines above it contain a statement and its denial. The check beside line 1 reminds us that that line has been analyzed in lines 3 and 4.

Next we will look at an argument that contains a disjunction.

1. A ∨ B
<u>2. ~A</u>
∴ B

Conjunctions only require listing the conjuncts. If the conjunction is true, both components must be true. A disjunction is true if either or both components are true. A different rule is required for disjunction.

RULE OF DISJUNCTION: At the end of each open branch in the path containing the disjunction, create a new branch with the disjuncts. Then check the line number of the disjunction.

An open path is one with no "x" at the end of it. Here is the tree for the above argument.

√	1.		A ∨ B		premise
	2.		~A		premise
	3.		~B		denial of conclusion
	4.	A		B	from 1, disjunction
		x		x	

Since each path of the tree is closed, the argument is valid.

We also need rules for the denials of conjunctions and disjunctions. The argument below contains a denied conjunction.

1. ~(A & B)
2. A

∴ ~B

A statement of the form "~(p & q)" is logically equivalent to a statement of the form "~p V ~q." Thus a denied conjunction requires a branch.

RULE OF DENIED CONJUNCTION: At the end of each open branch in the path containing the denied conjunction, create a new branch with the denial of the conjuncts at the end of each new path. Check the line number of the denied conjunction.

For this argument, we will also need the rule of double denial.

RULE OF DOUBLE DENIAL: List, at each open branch in the path, any statement in the path that previously had a double denial, and check the line number that contained the double denial.

Using the above rules, we can now construct the following tree.

√	1.		~(A & B)		premise
	2.		A		premise
√	3.		~~B		denial of conclusion
	4.		B		from 3, double denial
	5.	~A		~B	From 1, denied conjunction
		x		x	

The check by line 1 indicates that it has been analyzed in line 5. The check by line 3 indicates it has been analyzed in line 4. There are x's at the ends of both paths. ~A contradicts line 2. ~B contradicts line 4. The argument is valid.

Next we look at an argument containing a denied disjunction.

1. ~(A ∨ B
2. B ∨ C

∴ C

A statement of the form "$\sim(p \vee q)$" is logically equivalent to a statement of the form "$\sim p$ & $\sim q$." Thus a denied disjunction requires a list.

RULE OF DENIED DISJUNCTION: List the denial of the disjuncts at the end of each open branch in the path containing the denied disjunction. Check the line number of the denied disjunct.

√	1.		$\sim(A \vee B)$	premise
√	2.		$B \vee C$	premise
	3.		$\sim C$	denial of conclusion
	4.		$\sim A$	from 1, denied disjunction
	5.		$\sim B$	
	6.	B	C	from 2, disjunction
		x	x	

The denied disjunction is in line 1. Lines 4 and 5 are listings of the denials of the disjuncts. Line 6 is a branch analyzing the disjunction in line 2. Since both paths contain contradictions, the argument is valid.

A statement of material implication, "$p \supset q$", is equivalent to a statement of the form, "$\sim p \vee q$". Since material implication is equivalent to a disjunctive form, the rule is similar to the rule for disjunction.

RULE OF MATERIAL IMPLICATION: At the end of each open branch in the path containing the material implication, create a new branch with the denial of the antecedent at one path and the consequent at the other. Then check the line number of the material implication.

The use of the rule is illustrated in the following tree.

√	1.	$A \supset B$	premise
	2.	$\sim B$	premise
	3.	$\sim\sim A$	denial of conclusion
	4.	$\sim A$ B	from 1, material implication
		x x	

Line 1 is the material implication. It is analyzed in line 4. Note carefully that the antecedent is denied, but not the consequent.

A statement of the form, "$\sim(p \supset q)$," is equivalent to a statement of the form, "p & $\sim q$." Denied material implication uses a list rather than a branch.

RULE OF DENIED IMPLICATION: List, at each open branch in the path containing the denied implication, the antecedent and the denial of the consequent. Then check the line number of the denied implication.

The rule is used in the following tree.

√	1.	$\sim(A \supset B)$	premise
	2.	$\sim A$	denial of conclusion
	3.	A	from 1, denied implication
	4.	$\sim B$	
		x	

The denied implication is in line 1. The analysis of this line is in lines 3 and 4. Note that the consequent is denied, but not the antecedent.

A statement of material equivalence, "$p \equiv q$," is logically equivalent to a statement of the form, "$(p \& q) \vee (\sim p \& \sim q)$." The rule of material equivalence involves both a branch and a list.

RULE OF MATERIAL EQUIVALENCE: Create a new branch at the end of each open path containing the material equivalence. At the first branch, list each term of the equivalence. At the second branch, list the denials of each term. Check the line number of the equivalence.

Here the rule is illustrated.

√	1.	$A \equiv B$		premise
	2.	A		premise
	3.	$\sim B$		denial of conclusion
	4.	A	$\sim A$	from 1, material implication
	5.	B	$\sim B$	
		x	x	

Line 1 contains the material equivalence. It is analyzed in lines 4 and 5. The first path is closed because it contains B and ~B. The second is closed because it contains A and ~A.

The denial of material equivalence, "$\sim(p \equiv q)$," is logically equivalent to "$(p \& \sim q) \vee (\sim p \& q)$." Thus the denial of material equivalence also involves both a branch and a list.

RULE OF DENIAL OF EQUIVALENCE: At the end of each open branch in the path containing the denied equivalence, create a new branch. At the end of the first path of the branch, list the first term of the equivalence and the denial of the second. At the end of the second path of the branch, list the denial of the first term and then the second term of the equivalence. Check the line number of the denial of the equivalence.

The rule is illustrated in the following tree.

√	1.	$\sim(A \equiv B)$		premise
	2.	$\sim A$		premise
	3.	$\sim B$		denial of conclusion
	4.	A	$\sim A$	from 1, denied equivalence
	5.	$\sim B$	B	
		x	x	

The denied equivalence is in line 1. It is analyzed in lines 4 and 5. The first path contains A and ~A. The second path contains B and ~B. Thus the argument is valid.

These are all the rules necessary for evaluating arguments by means of reduction trees. All the arguments examined so far have been valid. We now need to look at some invalid arguments. In the case of an invalid argument, there will be at least one path of the tree which will not close. Any unclosed path represents a counter-example to the argument. The counter-example can be read off the open path by determining the truth values of the constants or variables along that path. A simple example is the argument:

$$\begin{array}{l} A \vee B \\ \underline{A \quad\;\;} \\ \therefore \sim B \end{array}$$

The tree for the argument is:

√	1.	$A \vee B$		premise
	2.	A		premise
√	3.	$\sim\sim B$		denial of conclusion
	4.	B		from 3, double denial
	5.	A	B	from 1, disjunction
		o	o	

This argument is invalid. Neither path has closed, and every line is either checked, or consists only of a truth functionally simple statement or the denial of a truth functionally simple statement. An "o" is placed at the end of the path to indicate that in the path every line is either checked, or consists only of a truth functionally simple statement or the denial of a truth functionally simple statement.

It is helpful to compare the tree to a truth table for the argument.

		premise	premise	conclusion
A	B	$A \vee B$	A	$\sim B$
T	T	T	T	F
T	F	T	T	T
F	T	T	F	F
F	F	F	F	T

The first line of the truth table contains true premises and a false conclusion. This constitutes the one counter-example to the argument. Looking at the two open paths of the reduction tree, we can see that A and B are true in both paths. This is how we can read the counter example from the tree.

Now a more complex example.

√	1.		$A \equiv B$			premise
√	2.		$\sim B \vee C$			premise
	3.		C			premise
√	4.		$\sim\sim A$			denial of conclusion
	5.		A			from 4, double denial
	6.	$\sim B$		C		from 2, disjunction
	7.	A $\sim A$		A $\sim A$		from 1, material
	8.	B $\sim B$		B $\sim B$		equivalence
		x x		o x		

The argument is invalid, since only three of the four paths are closed. Examining the third path which is open, we see that A, B, and C are all true. This suggests that we will find the first row of the truth table to be a counter-example.

A B C	$A \supset B$	$\sim B \vee C$	C	$\sim A$
T T T	T	F T	T	F
T T F	T	F F	F	F
T F T	F	T T	T	F
T F F	F	T T	F	F
F T T	T	F T	T	T
F T F	T	F F	F	T
F F T	T	T T	T	T
F F F	T	T T	F	T

Exercise

Construct reduction trees for the following argument forms. If the argument form is invalid, give a counter-example

1. $p \supset q$
   $r \supset s$
   $\underline{q \supset r}$
   $\therefore\ p \supset s$

2. $p \supset q$
   $r \supset s$
   $\underline{\sim s}$
   $\therefore\ \sim p$

3. $(p \,\&\, q) \vee r$
   $\underline{\sim p}$
   $\therefore\ r$

4. $(p \vee q) \vee r$
   $\underline{\sim p}$
   $\therefore\ r$

5. $(p \& q) \supset r$
$r \vee s$
$\underline{s}$
$\therefore p$

6. $p \equiv q$
$\underline{p \vee q}$
$\therefore p \& q$

7. $\sim(p \& q)$
$q \vee r$
$\underline{p}$
$\therefore r$

8. p
$(p \vee q) \supset r$
$\underline{\sim(r \& s)}$
$\therefore \sim s$

9. $\sim p \supset \sim q$
$p \supset r$
$\underline{q}$
$\therefore r$

10. $\sim(p \vee q)$
$\underline{(r \& s) \vee p}$
$\therefore s$

11. $(p \& q) \vee (r \& s)$
$\underline{p \vee q}$
$\therefore r \vee s$

12. $(p \& q) \vee (r \& s)$
$\underline{p \vee r}$
$\therefore q \vee s$

13. $p \supset (q \supset r)$
$\underline{(\sim p \vee r) \supset s}$
$\therefore s$

14. p
$(p \vee q) \supset r$
$r \vee s$
$\underline{\sim s}$
$\therefore \sim q$

15. $\sim(p \& q)$
$\underline{\sim q \supset r}$
$\therefore r \vee s$

16. $\sim(p \supset q)$
$\underline{q \supset r}$
$\therefore r \supset \sim p$

17. $\sim(p \equiv q)$
$\underline{p \vee \sim q}$
$\therefore \sim p$

18. $\sim[(p \& q) \vee (r \supset s)]$
$\underline{p \vee q}$
$\therefore s$

19. $p \supset q$

$q \supset r$

$\underline{\sim(p \supset s)}$

$\therefore \sim(r \supset s)$

20. $\sim[(p \,\&\, q) \vee r]$

$\underline{p \supset r}$

$\therefore q$