Appendix I: Rules of Inference

What follows are graphic representations for each of the 15 Rules of Inference, followed by a brief discussion, that will be used in this course. This supplement to the material in the Manual Part II, is intended to provide an alternative explanation of what each rule requires and what results from its application, in a more condensed, visual way that may be more easily accessible for some learners.

Each rule is presented as an “input-output” machine. The “input(s)” represent the type of statement to which you can apply a given rule, or the type of statement or statements required to be “in play” in order to apply the rule. These statements could be premises that are given at the outset of the argument, derived statements or even assumptions. In the diagram the input is presented as a type of statement. Any substitution instance of this type can be a suitable input.

The “output” is the result, the inference, or the conclusion you can draw forward when the rule is validly applied to the input(s). The output statement would be the statement you add to your left-hand derivation column, and justify with the rule you are applying.

Within each box is an explanation of what the rule does to any given input to achieve the output.

Each rule works on a main logical operator, or with a quantifier. Each rule is unique, reflecting the specific meaning of the logical sign (quantifier or operator) addressed by the rule.
Quantifier rules

**Universal Elimination**  \( \forall \varepsilon \)

**Input**

\((\forall x) Bx\)

1) ELIMINATE the universal quantifier

2) Change the variable \((x, y, \text{ or } z)\) to an individual

**Output**

\(Ba\)

Discussion: You can use this rule on ANY universally quantified statement, at any time in a proof. You can replace the variable with ANY individual constant, at any time in the proof. What individual constant you choose will depend on the premises and the conclusion statements in the argument you are proving.

* There are 4 possible situations that will impact which individual constant to use in a proof.

1) If there is no individual in the argument, and you do not need to use Universal Introduction to quantify the individual, you may use *any* individual constant, a-w.

2) If there is an individual constant already in the argument – in a premise or in the conclusion – you replace the variable with *that* individual constant.

3) If you have introduced “t” into the proof using Existential Elimination, you replace the variable with “t” when you use this rule.

4) If all premises are universally quantified and the conclusion is universally quantified, you must use “r” to replace the variable.

Where you have a compound expression, such as \((\forall x)(Ax \supset \neg Bx)\) or other like statements where the quantifier has scope over more than one variable, you must replace all occurrences of the variable with the same individual constant. So, the output for \((\forall x)(Ax \supset \neg Bx)\) would be \(Aa \supset \neg Ba\).
Discussion: This rule can only be applied to a “randomly selected individual.” By convention we are using the symbol of lower case “r” to stand for a “randomly selected individual.” A “randomly selected individual” represents all members of the class but is not a specific or individually designated or identified individual. I call it the “John Doe factor.” John Doe is somehow represents all of us yet none of us specifically.

The difference between “r” and “x” is subtle, yet significant. “r” is not a variable but a representative individual. We acknowledge this difference by using “r,” to show that we understand the conditions under which it is valid to derive a universal conclusion or generalize across an entire class of individuals.
**Existential Elimination**  
\[ \exists x \, B(x) \]

**Input:** \((\exists x) \, B(x)\)

**1) ELIMINATE** the existential quantifier

**2) Change the variable** \((x)\) to a “temporarily named” individual \((t)\)

**RESTRICTIONS:**
1) “\(t\)” cannot be an individual constant within the argument context
2) you must requantify before the proof is complete

**Output:** \(B(t)\)

**Discussion:** This rule can be applied to any existentially quantified expression.

The existential quantifier implies that some “\(x\)” exists. However, we do not know who or what this “\(x\)” is. For example, the statement above could mean, Someone is brave, or there exists someone who is brave. Or it could mean, Something is blue. But it doesn’t tell us who it is that is brave, or what it is that is blue. Because we want our reasoning to be specific and focused, we give this someone or something a temporary name. The letter “\(t\)” is used for this purpose.
**Existential Introduction** \( \exists \)

**Input**

\( Ba \)

1) **INTRODUCE** the existential quantifier

2) **Change the individual constant** (a-w) to a variable (x)

**Output**

\( (\exists x) \, Bx \)

**Discussion:** “a” could just as easily be **any** individual constant. You may apply this rule whenever you have any individual constant in your argument, any time.

There’s not much more to say about this rule. The logical inference is pretty obvious. If something is true of any given individual (a, b, c, etc.), then it follows immediately that it is true of someone/thing.
**Discussion:** This rule lets you change quantifiers, but you must change the position of the negation sign. It also works in reverse. So, $(\exists x) \sim Bx$ can be changed to $\sim(\forall x) Bx$.

**Note:** You will NOT use this rule where negations are not present.

**Note:** It is not valid to simply change quantifiers. You can, however, *always* derive an existential statement from a universal statement. You just need to take the step of going through an individual constant.

You can NEVER validly derive a universal statement from an existential statement. There are many variations of this kind of fallacy, most commonly called hasty generalization, sweeping, fallacious or unwarranted generalization.
**LOGICAL EQUIVALENCE**

**SPECIAL CASE: Standard Form Categorical Statements**

**INPUT**

\[ \sim (\forall x) (Bx \supset Cx) \]

**1) CHANGE the quantifier**

**2) MOVE the negation sign to the second term**

**3) CHANGE the main logical operator to coordinate with the new**

**Output**

\[ (\exists x) (Bx \cdot \sim Cx) \]

**Discussion:** This special case is a short cut for changing “not all” expressions, into their logically equivalent “some are not” expressions and vice versa, or “it is not true that some” expressions, into “no” or “none.”

Note the extra change you must make to coordinate the main logical operator with the quantifier.

Note that it is not a logically valid inference to move from “not all” to “some are,” or from a negative universal statement to an existential negative.

To use this rule easily and confidently, you need first to have learned the standard form predicate formulae for each of the 4 generalized categorical statements (All S is P, Some S is P, No S is P and Some S is not P.)
**LOGICAL OPERATOR RULES**

**Conjunction Elimination** \( \cdot E \)

**Input**

\[ A \land B \]

**Discussion:** This rule can be applied to any statement that has a conjunction as its main logical operator.

Either side of the conjunction may be derived, in any order. In this case, you can derive \( B \) as easily as \( A \). If the conjunction is assumed to be true, then it follows that both conjuncts are true. This is a very simple rule to use.

**Output**

\[ A / B \]
**Conjunction Introduction**

**Discussion:** This rule allows you to construct conjunctions. You need two inputs. Each input must be presented on its own line, either as a premise, a derived statement or a temporarily introduced assumption.

You may conjoin the two conjuncts in any order. This means that you can derive $B \cdot A$ as easily as $A \cdot B$. Which one of these you derive depends on the context of the proof and what is required to get to the given conclusion.
Discussion: This rule only allows you to derive the consequent (right side) of a conditional statement if, and only if you first have available its antecedent (left side) statement.

This is perhaps the most important rule you will use. It will be used in the majority of proofs that you will do. This rule only allows you to derive the right side, or the consequent of a conditional statement. You must first either have “in play” or you must derive the left side (the antecedent).
**Discussion:** This rule allows you to derive one side of a bi-conditional statement if, and only if you have the other side. In the diagram above the blue statement symbols represent an alternative pattern.

The difference between Conditional Elimination and Bi-conditional Elimination is that you may attack the Bi-conditional from either the right or the left side. One side will get you the other side. This is NOT true for Conditional Elimination, which cannot be used to derive the left side. You can only derive the right side of a conditional statement, and only if you first have the left side “in play.”
**Discussion:** This rule plays off the fact that the bi-conditional statement is an abbreviation of the statement \((A \supset B) \cdot (B \supset A)\). So, to prove a bi-conditional, you simply need to find the two conditional statements that are implied by the bi-conditional relationship.

While, in this form, Bi-conditional Elimination is not itself an assumption rule, you may need to use assumptions to prove one or both of the conditional statements.
Disjunction Introduction  \( v I \)

**Input**

This rule allows you to construct disjunctions.

You need one input. The input becomes one of the disjuncts. The other disjunct is anything you need it to be.

**Output**

\( A \lor B \)

Discussion: This rule allows you to conclude a disjunction from any statement.

This is a very simple rule to execute. You simply “create” a disjunction from any statement (the “root” statement) by adding a disjunction symbol and *any statement you want or need* onto the “root.” You may add to the left or the right depending on what you need.

The validity of this move is based on the definition of the inclusive disjunction as defined by a truth table. An inclusive disjunction is true in any case when one disjunct is true. Therefore, the second disjunct could be false and yet the disjunction as a whole will be true. For this reason, you do not need to derive the statement you are adding on.

This move may seem – like you are doing something invalid. But you are not—think about it. All we are doing in a proof is moving from statements we are assuming are true, to derive further statements as valid deductions. We are not proving truth. We are proving validity.

You use this rule strategically, to move the proof along, often to create statements you need for conditional elimination or bi-conditional elimination, or to add a statement into the conclusion that is not present in the original premise set.
Assumption Rules

In this course, we will use three assumptions rules. Here are some general things that you should know about using assumption rules.

- All assumption rules start with an assumption.
- An assumption is a statement that YOU introduce temporarily so that you can move the proof along.
- Once introduced, an assumption works just like a premise, or any statement that has been derived. You can use it immediately to derive more statements.

- All assumptions must proceed to a specific subgoal.
- The subgoal changes depending on the rule you are using.
- All assumptions must be closed after they reach the subgoal.

- A final conclusion is drawn outside the scope of the closed assumption.
- Once an assumption is closed, you cannot validly use any steps derived within the scope of the assumptions. (This would include the assumption and all statements including the subgoal.)

- You may introduce one assumption inside of another assumption. This is called “nesting.” When you nest assumptions you must close the assumptions in reverse order.

- There is no limit on the number of assumptions you can introduce. More sophisticated proofs involve multiple assumptions.

In the diagrams that follow, broken line arrows will indicate that the input is an assumption. Within the “box” will be the subgoal, also indicated by a broken line arrow. The “outcome” is the statement that will be justified by the rule being illustrated in the diagram. This is what the assumption rule or strategy lets you derive and it is derived outside the scope of all assumptions.
Conditional Introduction

Input

A

The assumption is always the antecedent of the conditional you want to prove.

SUBGOAL

B

The subgoal is always the consequent of the conditional you want to prove.

Output

A ⊃ B

Discussion: This rule allows you to derive a new conditional statement, one that is not present in the premise set or cannot be directly derived.

You will get your assumption from the conditional you want to prove. In this course it will be the antecedent of the conclusion for most problems. Remember that the antecedent, or left side of the conditional operator is an “if” statement, and an assumption is a “what if” kind of statement.

The subgoal is always the consequent of the conditional you are trying to prove. In most proofs, it is a statement that can be derived only after the assumption is introduced.

The conditional statement you derive is stated outside the scope of the assumption.

In this proof strategy, you are proving a relationship between the assumption and the subgoal.
**Negation Introduction / Negation Elimination**  
~I / ~E

AKA: Reductio ad absurdum, indirect proof, proof from contradiction

**Discussion:** The assumption is always the opposite of what you want to prove. The subgoal is always a contradiction. The form of contradiction we will use in this class is \( x \cdot \neg x \). You may replace “x” with any statement symbol – simple or compound, as you are able to derive them within the proof. For example, you might derive \( D \cdot \neg D \), or you might derive \( Q \cdot \neg Q \), or you might derive \( (A \lor B) \cdot \neg (A \lor B) \). All these are examples of a contradiction.

Negation Introduction and Negation Elimination are the same rule. The distinction depends on whether you add or eliminate the negation sign after the subgoal has been reached. If you want to prove \( \neg A \) you assume \( A \). If you want to prove \( A \), you assume \( \neg A \). If you want to prove \( \neg (B \lor E) \), you assume \( B \lor E \).

You will use this rule ONLY when there are negations within the proof and when a negation sign “moves.”
Disjunction Elimination  v E  
aka Disjunction Exhaustion, Proof by Cases

A v B  
The disjunction MUST be given as a premise or derived before you can set up for this rule.

Discussion: You will only use this rule when you have a disjunction within the argument-proof context that can function as the source of the assumptions. The disjunction asserts that either A or B is true. So, while you may not know which is true, you can be certain that one of them is true. It is this certainty that provides the valid basis of this rule.

A and B are two disjuncts. Each disjunct is assumed separately. Under each assumption you must derive a common statement, in this case “C.” C is the statement you want to prove. It is stated three times, once under each assumption, and a final time outside the scope of any assumption. It is this last statement of “C” that is justified with vE rule.