PHIL012. SYMBOLIC LOGIC – PROPOSITIONAL LOGIC – DERIVATIONS

When we argue, what we want are (i) clearly specifiable rules, (ii) that apply to any particular subject matter, and (iii) that legitimate transitions from one proposition (or set of propositions) to the next. A **proof** is a finite sequence of well-formed formulas (or propositions), each of which is either a premise, assumption, or is the result of the preceding formulas and a derivation rule. What this means is:

(1) no proof will be infinitely long, the conclusion of a proof is its *conclusion*
(2) every line in a proof will be one of the following: a premise, assumption, or derived proposition.
(3) various rules (called 'derivation rules') allow for moving forward in a proof.

To indicate the presence of an argument, we use the *single turnstile:* '⊢'. This symbol indicates the presence of an argument: the propositions to the *left* of the turnstile are the **premises** while the proposition to the *right* of the turnstile is the **conclusion**. For example, the turnstile in the following

P∧R, Z ⊢ R

indicates the presence of an argument where '**P∧R**' and '**Z**' are the premises and '**R**' is the conclusion.

Proofs: The Setup

A proof begins with an initial setup involving *three* columns:

(1) for numbering the premises,
(2) writing (stacking) the propositions,
(3) justification of propositions and indicating the goal proposition (or conclusion)

To illustrate, consider the following **P∧R, Y→R ⊢ Z**

1	P∧R	P
2	Y→R	P / Z

In the setup of the above proof, 'P∧R' and 'Y→R' are premises (and we use 'P') to indicate this. The conclusion (goal proposition) is 'Z'.

Proofs: Intelim Derivation Rules

In what follows, we develop a system of derivation rules (PD). A derivation rule is a syntactically-articulated rule that states conditions under which you can move forward in a proof. In some sense, it gives you *permission* to move down a line in a proof. There are two main types of derivation rules: **introduction rules** (these introduce a proposition of a certain type into a proof) and **elimination rules** (these work from propositions of a certain type in a proof).

1	**Conjunction Introduction (∧I)** From **P** and **Q**, we can derive **P**∧**Q**. Also, from **P** and **Q**, we can derive **Q**∧**P**.		**P** **Q** **P ∧ Q** **Q ∧ P**	 ∧I ∧I

Conjunction introduction states that from two different propositions, we can derive the conjunction of these propositions. For example, prove: P, R, Z ⊢ P∧Z

1	P	P
2	R	P
3	Z	P / P∧Z
4	P∧Z	1,3∧I

2	**Conjunction Elimination (∧E)** From **P∧Q**, we can derive **P**. Also, from **P∧Q**, we can derive **Q**.		**P ∧ Q** **P** **Q**	 ∧E ∧E

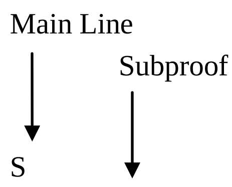
Conjunction elimination states that from a conjunction, we can derive either of the conjuncts. For example, prove: P∧Z ⊢ Z

1	P∧Z	P
2	Z	1∧E

Classroom Exercises
1. P∧(R∧M) ⊢ R
2. P∧(R∧M) ⊢ M
3. P, R, M ⊢ (M∧P)∧R
4. P, R, M ⊢ (M∧R)∧R
5. P∧R, ¬Z∧¬W ⊢ P∧¬W
6.* P∧(¬R∧¬W), L, R∧Z ⊢ ¬R∧(L∧Z)

Assumptions & Subproofs

An assumption (abbreviated as ‘A’) is a proposition taken to be, or assumed, true for the purpose of proof. Each time you make an assumption, you indent, draw a line indicating that you are moving into a **subproof**, and justify that proposition you assumed with an ‘A’.



1	S		Premise
2		│ B	A

After you’ve made an assumption, you can reason within the subproof that has been created by the assumption:

1	S	Premise
2	│ B	A
3	│ S∧B	1,2∧I

The above is similar to saying ‘Let’s agree that ‘S’ is true. Now, let’s assume ‘B’ is true. Well, if ‘S’ is true, then given our assumption ‘B’, it follows that ‘S∧B’.’

You are not limited to one assumption. You can make assumptions within assumptions. For example, consider the proof just below. This proof reads something like the following:

> Let’s say that ‘Q’ is true. Given that ‘Q’ is true, let’s assume ‘S’. Now that we’ve assumed ‘S’, let’s assume ‘W’.

1	Q	P
2	│ S	A
3	│ │ W	A

You can think of subproofs like containers or nests. That is the subproof begun by S *contains* the subproof begun by W. Likewise, the mainline of the proof, beginning with ‘Q’ *contains* the subproofs begun by ‘S’ and ‘W’. In the language of nests, ‘W’ is in the *nest* begun by ‘S’ and ‘S’ is in the nest of the main line of the proof. ‘W’ is in the most deeply nested part of the proof while ‘Q’ is in the least deeply nested part.

In addition, sometimes in proofs you will make an assumption and then make another assumption that is not related to the first assumption:

1	A	P
2	│ B	A
3	│ A∧B	1,2 ∧I
4	│ C	A
5	│ A∧C	1,4 ∧I

You can use derivation rules to reason within subproofs, but there are certain restrictions. The basic rule is the following:

> If **P** is in a section of the proof S_1 that contains another subproof S_2, then **P** can be used in S_2. If **R** is in a section of the proof S_3 that does not contain a subproof S_4, then **R** cannot be used in S_4.

Assumptions & Subproofs: Violations! There are two ways that you can violate the above rule.

Violation #1: You pull information from inside a subproof outside the subproof.

1	R		P
2		$\mid$ Z	A
3		$\mid$ $R \wedge Z$	1,2$\wedge$I
4	$Z \wedge R$		1,2$\wedge$I — **NO!**

Violation #2: You pull information across subproofs.

For example:

1	A		P
2		$\mid$ B	A
3		$\mid$ $A \wedge B$	1,2 $\wedge$I
4		$\mid$ C	A
5		$\mid$ $B \wedge C$	2,4$\wedge$I — **NO!**

3	**Conditional Introduction ($\rightarrow$I)** From a derivation of **Q** within a subproof involving an assumption **P**, we can derive **P$\rightarrow$Q** out of the subproof.		$\mid$ **P** $\mid$. $\mid$. $\mid$. $\mid$ **Q** **P $\rightarrow$ Q**	A $\rightarrow$I

Conditional introduction allows for introducing a conditional **P$\rightarrow$Q** outside of a subproof given a subproof containing **P** as the assumption and **Q** as a derived proposition within that subproof.

Here is an example: $R \vdash Z \rightarrow R$

1	R		P / $Z \rightarrow R$
2		$\mid$ Z	A
3		$\mid$ $Z \wedge R$	1,2$\wedge$I
4		$\mid$ R	3$\wedge$E
5	$Z \rightarrow R$		2-4$\rightarrow$I

In-Class Practice Problem: $R \wedge Z \vdash W \rightarrow Z$

4	**Conditional Elimination (→E)** From **P→Q** and **P**, we can derive **Q**.		**P → Q** **P** **Q**	 →E

Conditional eliminations allows for deriving a proposition **Q** provided we have a conditional **P→Q** and the antecedent **P** of that conditional.

Here is an example: Z→R, Z∧P ⊢ R

1	Z→R	P
2	Z∧P	P / R
3	Z	2∧E
4	R	1,3→E

In-Class Practice Problem R→Z, Z→W, R∧M ⊢ W∧M

Reiteration

5	**Reiteration (R)** From **P** we can derive **P**.		**P** **.** **.** **.** **P**	 R

Reiteration allows for deriving a proposition **P** provided **P** already occurs in the proof. Here are two examples.

Example #1: Z ⊢ Z

1	Z	P / Z
2	**Z**	**1R**

Example #2: R ⊢ Z→R

1	R		P / Z→R
2		| Z	A
3		| **R**	**1R**
4	Z→R		2-3→I

Classroom Exercises
1. P→Z, P ⊢ P∧Z
2. (P∧Z)→W, P, Z ⊢ W
3. P ⊢ R→P
4. P, M ⊢ (R∨F)→(P∧M)

5. $(R\lor F)\to Z,\ M\land(R\lor F) \vdash M\land Z$
6. $R\to P,\ R\land L,\ \vdash P\to(L\land R)$
7. $\vdash R\to R$
8. $\vdash R\to(R\land R)$
9. $\vdash R\to[Z\to(M\to Z)]$

A Word of Encouragement

For students taking a first course in symbolic logic, proofs tend to be one of the most difficult topics to grasp. Unlike decision procedures (truth tables and truth trees), the process of solving a proof requires the use of strategies and a little trial-and-error. As you work through various proofs, don't get discouraged if you are unable to get the answer immediately or if you have to start a proof over. Just keep at it and practice, practice, practice!

Negation Introduction & Elimination

6	**Negation Introduction ($\neg$I)** From a derivation of a proposition **Q** and its literal negation $\neg$**Q** within a subproof involving an assumption **P**, we can derive $\neg$**P** out of the subproof.		| **P** | . | . | . | **$\neg$Q** | **Q** **$\neg$P**	A $\neg$I
7	**Negation Elimination ($\neg$E)** From a derivation of a proposition **Q** and its literal negation $\neg$**Q** within a subproof involving an assumption $\neg$**P**, we can derive **P** out of the subproof.		| **$\neg$P** | . | . | . | **$\neg$Q** | **Q** **P**	A $\neg$E

Classroom Illustrations

1. $P\land\neg P \vdash R$
2. $(P\lor Q)\to R,\ P\lor Q,\ \neg R \vdash \neg W$

Disjunction Introduction & Elimination

8	**Disjunction Introduction (∨I)** From **P**, we can validly infer **P**∨**Q** or **Q**∨**P**.		**P** **P ∨ Q** **Q ∨ P**	 ∨I ∨I
9	**Disjunction Elimination (∨E)** From **P**∨**Q** and two derivations of **R**—one involving **P** as an assumption in a subproof, the other involving **Q** as an assumption in a subproof—we can derive **R** out of the subproof.		**P ∨ Q** **P** . . . **R** **Q** . . . **R** **R**	 A A ∨E

In-Class Illustrations

1. $(P\vee Z)\rightarrow R,\ Z \vdash R\vee\neg L$
2. $P\vee Q,\ P\rightarrow R,\ Q\rightarrow R \vdash R$
3.* $T\vee(Z\wedge M),\ (Z\wedge M)\rightarrow(\neg R\wedge S),\ T\rightarrow(\neg R\wedge S) \vdash \neg R$

Biconditional Elimination & Introduction

10	**Biconditional Introduction (↔I)** From a derivation of **Q** within a subproof involving an assumption **P** and from a derivation of **P** within a separate subproof involving an assumption **Q**, we can derive **P**↔**Q** out of the subproof.		**P** . . . **Q** **Q** . . . **P** **P ↔ Q**	A A ↔I
11	**Biconditional Elimination (↔E)** From **P**↔**Q** and **P**, we can derive **Q**. And, from **P**↔**Q** and **Q**, we can derive **P**.		**P ↔ Q** **P** **Q** **P ↔ Q** **Q** **P**	 ↔E ↔E

Classroom Exercises   1. $(F\lor Z)\to(T\land P), (P\lor M)\to(R\land Z) \vdash P\leftrightarrow Z$ 2. $(P\lor\neg M)\leftrightarrow R, P\leftrightarrow(W\lor L), L \vdash R$

Proofs: Strategies

There are two main types of strategies: *proof strategies* and *assumption strategies*.

Proof Strategies

SP#1 (E)	First, eliminate any conjunctions with $\land$E, disjunctions with $\lor$E, conditionals with $\to$E, and biconditionals with $\leftrightarrow$E. Then, if necessary, use any necessary introduction rules to reach the desired conclusion.
SP#2 (B)	First, work backward from the conclusion using introduction rules (e.g. $\land$I, $\lor$I, $\to$I, $\leftrightarrow$I). Then, use **SP#1** (**E**).

1. $P\to(R\land M), (P\land S)\land Z \vdash R$
2. $P\to R, Z\to W, P \vdash R\lor W$

Assumption Strategies

SA#1 (P,¬Q)	If the conclusion is an ***atomic proposition (or a negated proposition*)**, assume the negation of the proposition (or the non-negated form of the negated proposition), derive a contradiction and then use $\neg$I or $\neg$E.
SA#2 (→)	If the conclusion is a ***conditional***, assume the antecedent, derive the consequent, and use $\to$I.
SA#3 (∧)	If the conclusion is a ***conjunction***, you will need two steps. *First*, assume the negation of one of the conjuncts, derive a contradiction, and then use $\neg$I or $\neg$E. *Second*, in a separate subproof, assume the negation of the other conjunct, derive a contradiction, and then use $\neg$I or $\neg$E. From this point, a use of $\land$I will solve the proof.
SA#4 (∨)	If the conclusion is a ***disjunction***, assume the negation of the whole disjunction, derive a contradiction, and then use $\neg$I or $\neg$E

1. $P\to Q, \neg Q \vdash \neg P$
2. $R \vdash \neg(D\lor L)\to R$
3. $\neg(P\lor R) \vdash \neg P\land\neg R$
4. * $\neg(\neg P\land\neg Q) \vdash P\lor Q$

Consider $\neg(\neg P\land\neg Q) \vdash P\lor Q$. The strategy associated with assumptions is **SA#4 (∨)**

1	$\neg(\neg P \land \neg Q)$	P / $P\lor Q$

2	│ ¬(P ∨ Q)	A / P, ¬P

The subgoal at this point is to generate a proposition **P** and its literal negation **¬P** in the subproof, but it is not clear how to do this. You cannot generate **P** and **¬P** out of nothing so consider what propositions you do have and try to derive a proposition that is a literal negation of these.

1	¬(¬P ∧ ¬Q)	P / P∨Q
2	│ ¬(P ∨ Q)	A / P, ¬P
	│ .	
	│ .	
	│ .	
#	│ ¬P∧¬Q P∨Q	**?**

We thus have two options:

Option #1: derive ¬P∧¬Q since ¬(¬P∧¬Q) is its literal negation
Option #2: derive P∨Q since ¬(P∨Q) is its literal negation

Consider **option #2**. If we were to try to derive P∨Q, we need to make an assumption, and the strategic rule associated with deriving *disjunctions* **SA#4(∨)** says to assume the negation, derive **P** and **¬P**, and then use ¬E or ¬I. In the case of the above proof, the next step would be as follows:

1	¬(¬P ∧ ¬Q)	P / P∨Q
2	│ ¬(P ∨ Q)	A / P, ¬P
3	│ │ ¬(P∨Q)	A / P, ¬P

But this does not help since we still have no way to get P, ¬P in the proof.

So, consider **option #1**. If we were to try and derive ¬P∧¬Q, we would need to make an assumption, and the strategic rule associated with *conjunctions* **SA#3(∧)** says to assume the literal negation of each of the conjuncts in separate subproofs, derive **P** and **¬P** in each, and then use ¬I or ¬E.

1	¬(¬P ∧ ¬Q)	P / P∨Q
2	│ ¬(P ∨ Q)	A / P, ¬P
3	│ │ P	A / P, ¬P
	│ │	
	│	
n	│ │ Q	A / P, ¬P

Now the proof can be more easily solved.

Classroom Exercises

1. (¬P∧L)→¬Q, (M∧T)∧(¬R∧L), (M∧¬R)→(Z∧¬P)⊢ ¬Q∨(A↔B)

2. ¬R ⊢ P∨¬W→(Q∨¬R)
3. ⊢ ¬(W∧¬W)
4. P, (P∨W)→(R∧T), (T∨¬V)↔(¬R∧T) ⊢ S
5.* ¬P∨R ⊢ P→R
6.* P→R ⊢ ¬P∨R

Proofs: Additional Derivation Rules (PD+)

The set of 10 intelim rules along with reiteration forms PD, a derivation system capable of proving any valid argument in PL. In other words, PD consists of all of the *essential* derivation rules we need. However, you may have noticed that the proofs for many straightforwardly valid arguments are overly difficult or time-consuming. For example, the proof of P∨Q, ¬Q ⊢ P is overly complicated given that the argument is straightforwardly valid. In what follows, a number of additional derivation rules are added to PD to form PD+. These additional derivation rules serve to expedite the proof solving process.

12	**Disjunctive Syllogism (DS)** From **P∨Q** and ¬**Q**, we can derive **P**. From **P∨Q** and ¬**P**, we can derive **Q**.		**P ∨ Q** **¬Q** **P** **P ∨ Q** **¬P** **Q**	 DS DS

The general idea is that given a disjunction **P∨Q** and the literal negation of one of the disjuncts (either **¬P** or **¬Q**), we can derive the other disjunct.

1	P∨(R∧S)	P / P
2	¬(R∧S)	P
3	P	1,2DS

13	**Modus Tollens (MT)** From **P→Q** and ¬**Q**, we can derive ¬**P**.		**P → Q** **¬Q** **¬P**	 MT

The general idea is that given a conditional **P→Q** and the literal negation of the consequent **¬Q**, the negation of the antecedent **¬P** can be derived.

1	P→(S∨R)	P / P
2	¬(S∨R)	P
3	¬P	1,2MT

14	**Hypothetical Syllogism (HS)** From **P→Q** and **Q→R**, we can derive **P→R**.		**P → Q** **Q → R** **P → R**	 HS

The idea is that if you have two conditionals **P→Q** and **Q→R** where the consequent of one conditional **P→Q** is the antecedent of the other conditional **Q→R**, then you can derive a third conditional **P→R**.

Classroom Exercises

1. (R∧T)∨¬W, S∧¬¬W ⊢ R∧T
2. (P∧S)→W, ¬W∧T ⊢ ¬(P∧S)
3. (R∧T)→¬W, M→(R∧T), ¬W→(S∧R) ⊢ M→(S∧R)
4.* P∨¬(R∨S), R, L→¬P ⊢ ¬L

Proofs: Additional Derivation Rules (PD+), The Replacement Rules

All of the previous derivation rules have been *inference rules*, these are derivation rules that allow for deriving a proposition of one form from a proposition of another form. In addition to adding DS, MT, and HS to PD, we will also add a new kind of derivation rule, known as *replacement rules*. Replacement rules are derivation rules that allow for interchanging certain formulas or sub-formulas.

15	**Double Negation (DN)** From **P**, we can derive ¬¬**P**. From ¬¬**P**, we can derive **P**.		**P** ⊣ ⊢¬¬**P**	DN

DN allows for replacing a single formula or single subformula with its doubly negated form or taking a doubly negated formula and replacing it with its unnegated form. For example,

1	P→R	P / ¬¬(P→R)
2	¬¬(P→R)	1DN

It is important to note that replacement rules can be applied to a single *subformula*. For example,

1	P∨¬¬(R∧S)	P / P∨ (R∧S)
2	P∨(R∧S)	1DN

But, be careful! DN must be applied to the whole of a formula or subformula and not to part of one subformula and part of another subformula:

1	P∨(R∧S)	P
2	P∨¬(¬R∧S)	1DN – **NO!**

16	**De Morgan's Laws (DeM)** From **¬(P∨Q)**, we can derive **¬P∧¬Q.** From **¬P∧¬Q**, we can derive **¬(P∨Q).** From **¬(P∧Q)**, we can derive **¬P∨¬Q.** From **¬P∨¬Q**, we can derive **¬(P∧Q).**		**¬(P∨Q)** ⊣ ⊢**¬P∧¬Q** **¬(P∧Q)** ⊣ ⊢**¬P∨¬Q**	DeM DeM

In the case of DeM, you can interchange a negated disjunction **¬(P∨Q)** with a conjunction whose conjuncts are negated **¬P∧¬Q** (and vice versa) and you can interchange a negated conjunction **¬(P∨Q)** with a disjunction **¬P∨¬Q** whose disjuncts are negated (and vice versa).

For example, in the following proof, De Morgan's laws are applied to ¬R∧¬Q to derive ¬(R∨Q), i.e. turning a conjunction with negated conjuncts into a negated disjunction.

1	P→(R∨Q)	P
2	¬R∧¬Q	P / ¬P
3	**¬(R∨Q)**	**2DeM**
4	¬P	1,3MT

In the example below, DeM is applied to the negated disjunction ¬(R∨S) to derive a conjunction with two negated disjuncts.

1	¬(R∨S)	P
2	**¬R∧¬S**	**1DeM**
3	¬R	2∧E

17	**Implication (IMP)** From **P→Q**, we can derive ¬**P∨Q** From **¬P∨Q**, we can derive **P→Q**		**P→ Q ⊣ ⊢¬P ∨ Q**	IMP

In the case of IMP, you can interchange a negated conditional **P→Q** with a disjunction **¬P∨Q**.

Remembering that replacement rules can be applied to single sub-formula, notice how IMP is applied to the subformula 'P→R' in '¬(P→R)' in the following example:

1	¬(P→R)	P
2	**¬(¬P∨R)**	**1IMP**
3	¬¬P∧¬R	2DeM
4	¬¬P	3∧E
5	P	4DN

Classroom Exercises
1. ¬¬P→R, P, ¬¬R→(W∧Z) ⊢ ¬¬(W∧¬¬Z)
2. ¬(P∨R)→(¬Z∨¬W), ¬P∧¬R ⊢ ¬(Z∧W)
3. (P→R), (¬P∨R)→(Z→¬R) ⊢ ¬Z∨¬R
4.* ¬P∨R, ¬(P→R) ⊢ S
5.* P→¬(Z∨S), ¬(P→R) ⊢ ¬Z∨W
6.* P, ¬(¬P∧¬R)→¬(S→T) ⊢ S

Proofs: Revised Strategic Rules

In enhancing our proof system from PD to PD+, we also want to enhance the strategies with which we solve proofs.

SP#1 (**E+**)	First, eliminate any conjunctions with ∧E, disjunctions with DS or ∨E, conditionals with →E or MT, and biconditionals with ↔E. Then, if necessary, use any necessary introduction rules to reach the desired conclusion.
SP#2 (B)	First, work backward from the conclusion using introduction rules (e.g. ∧I, ∨I, →I, ↔I). Then, use **SP#1**(**E**).
SP#3 (**EQ+**)	Use DeM on any negated disjunctions or negated conjunctions, and then use **SP#1**(**E**). Use IMP on negated conditionals, then use DeM, and then use **SP#1**(**E**).

Classroom Exercises

1. P↔(R∨S), P∧¬S, Q→¬R ⊢ ¬Q
2. R∨(M∧T), ¬R∧¬W, L→W ⊢ ¬L
3. (R∨M)∨¬(S∨T), (S∨T)∨(Z∧E), ¬(R∨M) ⊢ E
4. ¬(P∨R), ¬P→¬(M∨S), ¬R→¬Q ⊢ ¬M∧¬Q
5. ¬(P→R), P→Z, ¬R→M ⊢ Z∧M
6. ¬(¬P→¬R), Z→P ⊢ ¬Z∧R
7.*⊢ ¬(P→R)→(S→¬R)
8.* ⊢ ¬(P∨R)→[(Z→R)→¬Z]
9.** ⊢ [¬(P→M)∧¬(T→S)]∨(P∨¬P)